Architectures multiprocesseurs pour applications de télécommunication basées sur les technologies d'intégration 3D

Walid Lafi

To cite this version:

Walid Lafi. Architectures multiprocesseurs pour applications de télécommunication basées sur les technologies d'intégration 3D. Autre. Université de Grenoble, 2011. Français. NNT : 2011GRENT037 . tel-00623415

HAL Id: tel-00623415

https://theses.hal.science/tel-00623415

Submitted on 14 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Micro et nano électronique**

Arrêté ministériel : 7 août 2006

Présentée par

« Walid LAFI »

Thèse dirigée par **« Ahmed JERRAYA »** et
codirigée par **« Didier LATTARD »**

préparée au sein du **Laboratoire CEA-LETI**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique et Traitement du Signal**

Architectures multiprocesseurs pour applications de télécommunication basées sur les technologies d'intégration 3D

Thèse soutenue publiquement le **« 11 Juillet 2011 »**,
devant le jury composé de :

Mme Lorena ANGHEL
Professeur à l'IP Grenoble, Présidente du jury
M. Ian O'CONNOR
Professeur à l'Ecole centrale Lyon, Rapporteur
M. Lionel TORRES
Professeur à l'Université de Montpellier 2, Rapporteur
M. Olivier SENTIEYS
Professeur à l'Université de Rennes 1, Membre
M. Ahmed JERRAYA
Directeur de recherche au CEA-LETI, Membre
M. Didier LATTARD
Ingénieur de recherche au CEA-LETI, Membre

Remerciements

Les travaux de recherche exposés dans la présente thèse de doctorat se sont déroulés au sein du Laboratoires d'Intégration sur Silicium des Architectures Numériques LISAN du Département d'Architecture, Conception et Logiciels Embarqués DACLE du CEA-Leti à Grenoble en France. Je tiens à remercier toutes les personnes qui ont contribué de près ou de loin à l'avancement des travaux présentés dans le présent document.

Pour commencer, j'adresse mes sincères remerciements à Monsieur Didier Lattard, ingénieur-chercheur au CEA-Leti, pour son encadrement, sa disponibilité, son expertise et sa patience durant les trois années de travail. Je tiens également à présenter mes vifs remerciements à Monsieur Ahmed Jerraya, directeur de programmes au CEA-Leti, pour son suivi et ses conseils pertinents en tant que directeur de thèse.

Je remercie tous les membres du jury avec en premier lieu les rapporteurs : M. Ian O'connor, professeur à l'Ecole Centrale de Lyon ; et M. Lionel Torres, professeur à l'Université de Montpellier 2. Mes sincères remerciements aussi à Mme Lorena Anghel, professeur à l'Institut Polytechnique de Grenoble, pour m'avoir fait honneur de président de jury, et à M. Olivier Sentieys, professeur à l'université de Rennes 2.

Je voudrais aussi remercier tous les membres du laboratoire LISAN pour les échanges constructifs, les conseils, et les moments de distractions : Alexandre, Christian, Denis, Didier, Fabien, Frédéric, Ivan, Jean, Jérôme, Pascal, Romain, Sébastien et Yvain.

Merci également aux postdocs, doctorants et stagiaires que j'ai rencontrés durant les 3 ans de travail : Bilel, Camille, Fadoua, Florian, Imen, Pallavi, Pierre Emanuel et Riadh, et tous les autres que j'ai oublié de citer.

Je n'oublie pas de remercier Marc Belleville, responsable scientifique du département DACLE, et François Bertrand, chef du laboratoire LISAN, pour m'avoir accueilli dans le laboratoire et soutenu mes idées. Ma sympathie à toutes les personnes du département, spécialement à Armelle et Caroline pour leurs disponibilités au secrétariat.

Enfin, mes sincères remerciements à toute ma grande famille : mon grand-père, mes parents, mon frère et mes deux sœurs.

Sans l'aide de toutes ces personnes, cette thèse n'aurait pas être possible.

Bonne lecture !

Résumé

Les travaux de cette thèse s'intéressent aux problèmes de performance et de coût des architectures MPSoC à base de NoC, en tirant parti des possibilités offertes par les technologies d'intégration 3D. Plusieurs contributions originales sont proposées. Tout d'abord, une étude approfondie à propos des différentes granularités de partitionnement au sein des circuits 3D est réalisée. En se basant sur cette analyse, ce travail de thèse est orienté aux architectures 3D partitionnées au niveau des blocs macroscopiques. Ainsi, la contribution de l'intégration 3D est limitée aux interconnexions verticales inter-blocs. Afin d'améliorer les performances de ces interconnexions, une topologie hiérarchique de NoC est proposée pour diminuer la latence et augmenter le débit des communications au sein des architectures 3D partitionnées au niveau des macro-blocs. D'autre part, un modèle au niveau du système est présenté pour évaluer et comparer les coûts des différentes options technologiques de l'intégration 3D. Partant de cette évaluation, nous proposons une architecture multiprocesseur reconfigurable empilable pour les applications de télécommunication 4G, en tenant compte des problèmes de coût.

Abstract

This PhD research is intended to deal with cost and performance issues of NoC-based MPSoC architectures by taking advantage of the opportunities offered by 3D integration technologies. Several original contributions are proposed. First, a deep investigation of the different partitioning granularities within 3D circuits is performed. Based on this analysis, this PhD work is oriented to focus on core-level partitioned 3D architectures, and then to restrict the contribution of 3D stacking to the global inter-block vertical interconnections. To enhance the performance of global interconnect architectures, a hierarchical NoC topology is proposed to improve communication latency and throughput within core-partitioned 3D architectures. On the other hand, a system-level cost analysis model is presented to assess and compare several 3D integration technology options. Based on this evaluation, we propose a cost-aware stackable reconfigurable multiprocessor NoC-based architecture to address the requirement of 4G telecom applications.

Contents

Remerciements i
Résumé iii
Abstract iv
Contents v
List of Figures vii
List of Tables ix

Résumé étendu **1**

Introduction **23**

1 3D integration technologies **29**
1 From SoCs and SiPs to 3D systems 30
2 Motivations for 3D die stacking 33
2.1 Reducing cost 33
2.2 Enhancing performance and form factor 34
3 3D circuit manufacturing technologies 36
3.1 Wafer thinning 37
3.2 TSV forming 38
3.3 Bonding 38
3.4 3D IC manufacturing yield 39
4 Challenges in 3D integration technologies 40
4.1 Thermal management 40
4.2 TSV area overhead 42
4.3 CAD tools 42
4.4 Test challenges 42
5 Potential applications 43
6 Conclusion 44

2 From 2D to 3D architectures: partitioning and cost challenges **47**
1 Partitioning challenges 48
1.1 Macro-block stacking 48
1.2 Functional unit stacking 51
1.3 Elementary operators stacking 52
1.4 Synthesis 53
2 Cost challenges 54
2.1 3D IC cost analysis: state of the art 54
2.2 A system-level cost model for 3D ICs 55
2.3 Cost analysis of 3D ICs 60

3 Conclusion 63

3 A hierarchical NoC for 3D MPSoCs 65
1 NoC paradigm 66
1.1 Evolution of interconnection structures 66
1.2 Principles of NoC 66
2 3D NoC: state of the art and challenges 71
2.1 3D mesh NoC 71
2.2 3D NoC-Bus 72
2.3 DimDe router 72
2.4 MIRA router 73
2.5 Other 3D NoC architectures 73
3 Architecture of the asynchronous hierarchical 3D NoC 74
3.1 Architecture of the 3D NoC 74
3.2 Design of the asynchronous router 78
4 Area and power evaluation of the hierarchical 3D router 82
5 Performance of the hierarchical 3D NoC 83
5.1 Simulation platform 83
5.2 Throughput 86
5.3 Latency 88
5.4 Result analysis 88
6 Conclusion 91

4 A reconfigurable and stackable circuit for 4G telecom applications 93
1 Implementation of 4G terminals: algorithms and requirements 94
1.1 OFDM demodulation 94
1.2 Channel estimation 95
1.3 MIMO decoding 96
2 A stackable circuit for LTE applications 98
2.1 Processing units 98
2.2 MIPS-based semi-distributed control 101
2.3 The network interface 103
2.4 3D asynchronous mesh NoC 105
2.5 Design results 105
2.6 Adequacy of the basic circuit to meet LTE requirements 105
3 Units' programming and platform control 107
4 Technological considerations 109
5 Case study: downlink part of the 4x2 LTE mode 110
5.1 Performance results 112
5.2 Power consumption results 113
6 Cost analysis 114
7 Conclusion 119

Conclusions and perspectives 121

Bibliography 125

List of Figures

1 General PhD framework 25

1.1 Evolution of microprocessor complexity 30
1.2 Example of a System-on-Chip 31
1.3 Example of a system-in-package 32
1.4 Convergence of the More Moore and the More than Moore trends . . 32
1.5 An example of a 3D system 33
1.6 Increase in the cost of a set of masks according to technological nodes 34
1.7 A 3D SoC designed with heterogeneous integration and same-die stacking approaches 35
1.8 Interconnects' length shortening in 3D ICs 35
1.9 Three ways to assemble the circuits vertically 37
1.10 F2F and F2B approaches in 3D stacking 37
1.11 Different TSV sizes and form factors 38
1.12 Die bonding using the copper pillar technique 39
1.13 Yield according to area 40
1.14 Yield according to area 41
1.15 TSV area vs transistor area 42
1.16 3D applications roadmap 44

2.1 Partitioning granularities: (a) macroscopic blocks (b) functional units (c) basic operators 48
2.2 Memory stacking schemes: (a) original 2D circuit, (b) stacking more SRAM above the initial circuit, (c) stacking L2 DRAM cache on top of the CPU, (d) stacking more DRAM on top of the initial 2D circuit 49
2.3 The 3D stacked processor-cache-memory system 50
2.4 PicoServer: a multi-processor architecture connected to a conventional DRAM using 3D integration technology 50
2.5 The investigated stacking approaches 55
2.6 Unit cost for a 2D circuit and its W2W, D2W and IbS 3D versions for different die areas 61
2.7 Unit cost for different numbers of layers using the W2W scheme . . . 62
2.8 Unit cost for different numbers of layers using the D2W scheme . . . 62
2.9 Unit cost for different numbers of layers using the IbS scheme 62

3.1 Interconnect structures: (a) direct connections, (b) bus, (c) NoC . . 68
3.2 Different NoC topologies 69
3.3 A static deadlock 71

3.4 3D mesh NoC architecture 72
3.5 3D NoC-bus router 73
3.6 DimDe router 73
3.7 MIRA router 74
3.8 Conceptual view of the 3D hierarchical router 75
3.9 Architecture of the 3D hierarchical NoC 76
3.10 Architecture of the 3D hierarchical router 77
3.11 Send/Accept protocol 77
3.12 NoC packet format 77
3.13 Synchronous circuit 79
3.14 A bundled-data channel 79
3.15 The 4-phase and 2-phase bundled-data protocols 80
3.16 The delay-insensitive 4-phase dual-rail protocol 80
3.17 Micro-architecture of the input port 81
3.18 Micro-architecture of the output port 82
3.19 Throughput variation under different synthetic traffic patterns 87
3.20 Latency variation under different synthetic traffic patterns 88

4.1 A functional block diagram of an LTE UE reception chain with 4 receive (Rx) antennas 95
4.2 A MIMO-OFDM system with N_T TX antennas and N_R RX antennas 96
4.3 3D reconfigurable circuit obtained by stacking multiple instances of a same basic circuit 98
4.4 SME buffer functional model 99
4.5 Architecture of the Mephisto DSP 100
4.6 Architecture of the OFDM core 101
4.7 The basic circuit 101
4.8 The MIPS-based global control unit 102
4.9 Diagram of tasks 103
4.10 The network interface blocs 104
4.11 The communication and configuration controller 104
4.12 TSV characteristics 105
4.13 Programming of the processing units and the NI 108
4.14 Example of the control code 108
4.15 Physical model of a TSV as proposed by Cadix and al. [1] 109
4.16 Driver and receiver buffers of a TSV 110
4.17 Mapping of the 3GPP LTE application on our 3D platform 111
4.18 2D reference architecture MAGALI 112
4.19 An abstract view of the simulation platform 112
4.20 Power consumption profile of the up-layer MIPS processor 114
4.21 Total cost of the 3 digital circuits for a total production volume of 1 million 117
4.22 Total cost of the 3 digital circuits for a total production volume of 10 million 117
4.23 Cost-effective approaches in the case of 3D stacking of different dies . 118
4.24 Cost-effective approaches in the case of 3D same-die stacking 118

List of Tables

2.1 Constant values used for test cost model 58
2.2 Costs of different 3D stacking steps 59
2.3 Technological parameters for 32nm-technology die 60
2.4 Technological parameters for 130nm-technology interposer 60

3.1 Comparison between NoC and bus architectures 67
3.2 Area and power results for the 4x4, 5x5 and 7x7 routers in 65nm technology 83
3.3 Delay results for the 4x4, 5x5 and 7x7 routers in 65nm technology . 84

4.1 The theoretical complexity of the LTE terminal with NT TX antennas and NR RX antennas 97
4.2 The theoretical complexity of the inversion of 2x2 and 4x4 matrixs . 97
4.3 The theoretical complexity of channel estimation and MIMO decoding for different LTE modes 97
4.4 Synthesis results in 65nm technology 106
4.5 The theoretical complexity of channel estimation and MIMO decoding for different LTE modes 106
4.6 The numbers of instances (of the basic circuit) required to address the different LTE modes 107
4.7 High density TSV electrical parameters 110
4.8 Time to process a TTI 113
4.9 Power consumption of the processing units 113
4.10 Power consumption of the MIPS control processor 114
4.11 Power consumption due to control 114

Résumé étendu

Introduction

Contexte

Les progrès de l'industrie électronique ont été menés depuis des décennies par la loi de Moore qui prévoit l'augmentation de la densité des transistors au sein des puces électroniques. Aujourd'hui, continuer la miniaturisation des circuits intégrés selon cette loi empirique semble incertain et très coûteux, en raison de plusieurs obstacles tels que l'augmentation exponentielle des courants de fuite et le gain limité en performance pour les technologies agressives. La technologie d'intégration 3D apparaît comme une solution prometteuse qui permet de poursuivre la miniaturisation des circuits intégrés sans suivre la loi de Moore.

Le laboratoire d'Electronique et de Technologie de l'Information LETI mène une activité importante concernant les technologies d'intégration 3D. En effet, plusieurs projets ont été réalisés pour développer ces technologies prometteuses, et des progrès ont été atteints. Par ailleurs, le LETI jouit d'une expertise significative dans le développement des architectures multiprocesseurs dédiées aux applications de télécommunications 4G. Depuis 2009, le LETI a mis au point le circuit MAGALI dédié au traitement en bande de base des applications de téléphonie mobile de la quatrième génération. Cette thèse fait partie du travail effectué par le LETI pour tirer parti des possibilités offertes par les technologies émergentes en vu d'améliorer les performances des architectures multiprocesseurs dédiées aux applications de télécommunications sans fil. Ces applications sont de plus en plus exigeantes en termes de puissance de calcul et de consommation d'énergie, en particulier avec la parution de la norme 4G.

Motivations et objectifs

Les technologies d'intégration 3D consistent à empiler verticalement des circuits intégrés et les interconnecter grâce à des vias qui traversent les couches de silicium (*TSV: Through-Silicon-Vias*). Les avantages potentiels de l'intégration 3D sont multiples et peuvent varier en fonction de l'approche d'empilement choisie : la multifonctionnalité, l'augmentation des performances, la diminution de la puissance consommée, la réduction du facteur de forme, l'amélioration du rendement et de la fiabilité, l'intégration hétérogène et la réduction des coûts. Forte de tous ces avantages, la technologie d'empilement 3D est une solution prometteuse pour surmonter les limitations actuelles des technologies CMOS agressives, telles que les gains limités en performances, les problèmes de fiabilité, le coût de fabrication prohibitif...

Néanmoins, afin de tirer parti du potentiel de l'intégration 3D, il est nécessaire d'analyser les différentes possibilités de partitionnement des circuits sur plusieurs couches, et de comparer les avantages et les inconvénients de chaque possibilité en termes de performance, de consommation d'énergie, d'efforts de conception et de coût de fabrication. Il est possible de partitionner un circuit selon plusieurs granularités, dont chacune a des avantages et des inconvénients. Un premier objectif de cette thèse est d'étudier chacune de ces options de partitionnement afin d'identifier celles qui s'adapte le mieux aux exigences de l'application visée en termes de performance et de viabilité économique. Au cours de ces travaux de thèse, l'analyse de partitionnement nous a conduits à choisir le partitionnement au niveau des blocs macroscopiques (empiler de la mémoire au dessus du processeur par exemple), et donc de limiter la contribution de l'empilement 3D au niveau des interconnexions globales verticales inter-blocs. Le deuxième objectif de la thèse est de proposer une topologie nouvelle de réseaux-sur-puce qui permet d'améliorer les performances des interconnexions au sein des architectures 3D partitionnées au niveau des macro-blocs.

Dans l'industrie électronique, le gain en performance ne peut pas être le facteur déterminant au moment de décider de passer d'un processus de fabrication à un autre tout à fait nouveau. En effet, le coût était toujours la motivation principale derrière le progrès de l'industrie des semi-conducteurs au cours des dernières décennies, et il continuera à l'être dans l'avenir. Ainsi, l'orientation de la communauté industrielle vers les nouvelles technologies d'intégration 3D doit être impérativement justifiée par des gains considérables en termes de coût. Le deuxième objectif de cette thèse est d'effectuer une analyse de coût des technologies d'empilement 3D. Cette analyse nous a permis d'identifier une approche d'empilement 3D rentable économiquement. Notre objectif final est d'étudier plus profondément cette approche d'empilement dans le contexte des applications de télécommunications sans fil.

Contributions

Ces travaux de recherche portent sur les architectures multiprocesseurs pour les applications avancées de télécommunications sans fil basées sur des technologies d'intégration 3D. Afin de répondre aux objectifs présentés précédemment, plusieurs contributions ont été réalisées :

- une étude bibliographique complète sur les technologies d'intégration 3D et les architectures 3D à différentes granularités de partitionnement,

- un modèle d'estimation de coût au niveau système pour étudier les tendances économiques des technologies d'intégration 3D,

- une nouvelle topologie de réseaux-sur-puce 3D permettant d'améliorer la performance des communications inter-bloc au sein des architectures 3D partitionnées au niveau des blocs macroscopiques,

- une architecture multiprocesseur reconfigurable empilable à base de réseau-sur-puce permettant d'implémenter les applications de télécommunications de 4ème génération (4G), tout en réduisant le coût de fabrication.

La figure 1 illustre le cadre général de ces travaux de recherche concernant les objectifs, les contributions et les résultats majeurs.

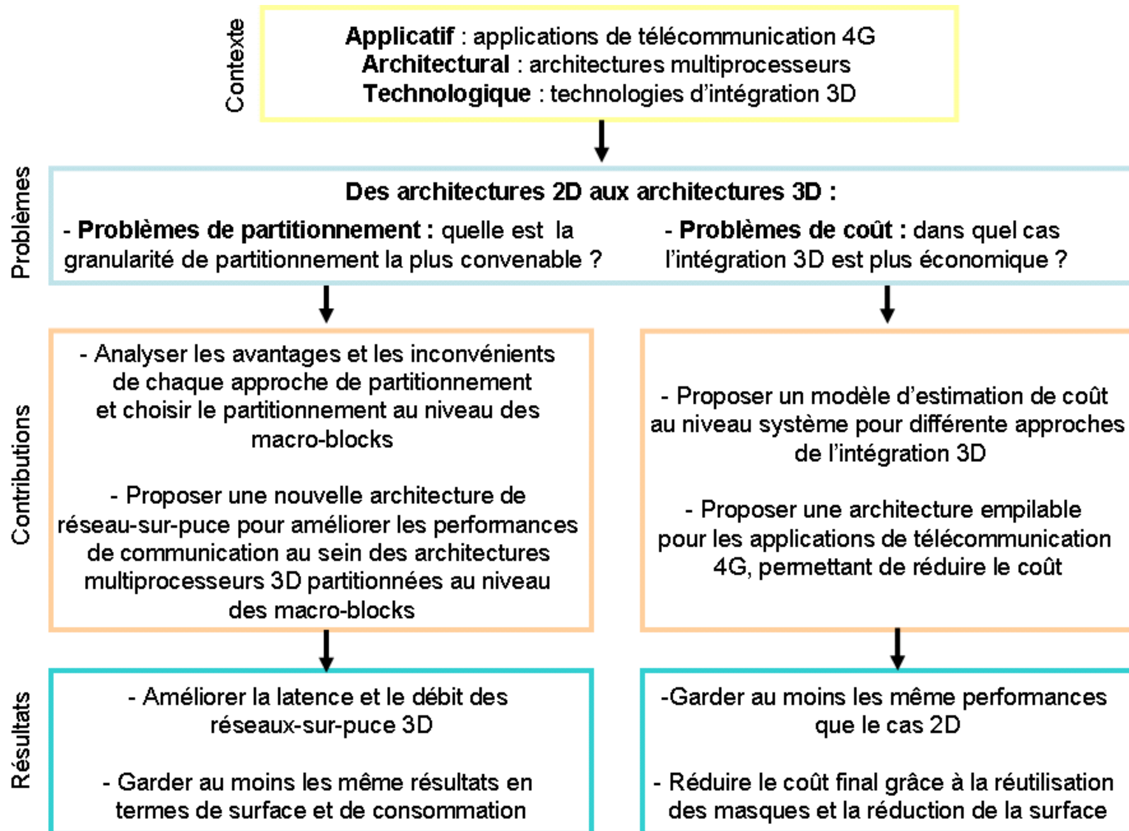


Figure 1 : Flot de thèse

Organisation du rapport

Outre la présente introduction, ce rapport est composé de quatre chapitres. L'objectif du premier est de présenter une vue d'ensemble sur les technologies d'intégration 3D. Il commence par décrire les tendances technologiques contemporaines. Ensuite, les motivations pour les technologies d'intégration 3D sont présentées. Après, les étapes du processus de fabrication des circuits 3D et les problèmes connexes tels que le rendement sont présentés. Enfin, les limitations majeures de l'empilement 3D, ainsi que les domaines d'application potentiels sont introduits.

Le deuxième chapitre traite de deux aspects importants de l'intégration 3D : le partitionnement et le coût. Ainsi, la première partie du deuxième chapitre discute des aspects de partitionnement des circuits intégrés 3D : elle introduit 3 niveaux de partitionnement, et explique leurs avantages et inconvénients. La deuxième partie s'intéresse aux problèmes de coût, et présente un modèle au niveau du système d'estimation de coût que nous utilisons pour analyser la rentabilité économique de l'intégration 3D.

Le troisième chapitre porte sur l'amélioration des performances des interconnexions au sein des architectures 3D partitionnées au niveau des blocs macroscopiques. Un nouveau routeur de réseaux-sur-puce 3D est proposé afin d'améliorer le débit

et la latence par rapport aux réseaux maillés 3D classiques. Le troisième chapitre commence par expliquer les motivations pour les réseaux-sur-puce, et certains de ses grands principes. Ensuite, les avantages et les inconvénients de plusieurs architectures de réseaux-sur-puce 3D de l'état de l'art sont présentés. Après, le réseau-sur-puce 3D hiérarchique ainsi que la conception du routeur asynchrone sont décrits en détails. Enfin, une évaluation de l'architecture proposée en termes de surface, de consommation et de performance est réalisée.

Le quatrième chapitre présente une architecture multiprocesseur empilable reconfigurable à base de réseaux-sur-puce pour les applications de télécommunications 4G, qui permet de réduire le coût de fabrication. Tout d'abord, certains aspects concernant la mise en œuvre de la partie bande de base numérique des terminaux 4G tels que les différents algorithmes utilisés et leurs exigences en termes de puissance de calcul sont décrites. Ensuite, l'architecture reconfigurable et empilable est présentée, et sa capacité à satisfaire les besoins la norme 4G est prouvée. Après, la reconfiguration et la programmation des différents composants du circuit proposé sont expliquées. Ensuite, un cas d'étude du mode de transmission 4x2 de la norme 4G et des évaluations de performance et de puissance sont présentés. Enfin, les avantages économiques du circuit proposé sont analysés.

La dernière partie de ce rapport est consacrée aux conclusions et aux perspectives d'éventuels travaux futurs.

Ci-après un résumé étendu qui reprend brièvement les 4 chapitres de la thèse.

1 Les technologies d'intégration 3D

Les technologies d'intégration 3D est une solution prometteuse permettant de continuer la miniaturisation des puces électroniques tout en intégrant plus de fonctions différentes. Les technologies d'intégration 3D consistent à empiler verticalement des circuits intégrés et les interconnecter grâce à des TSVs (figure 2).

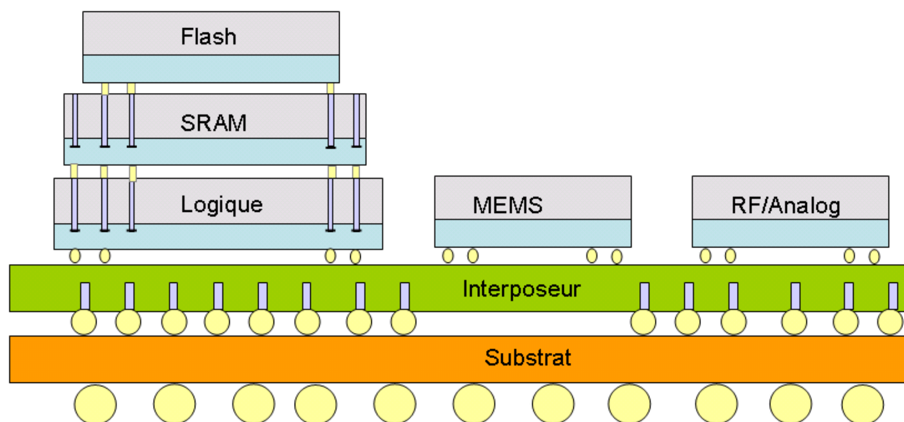


Figure 2 : Exemple de circuit 3D

1.1 Avantages de l'intégration 3D

Un avantage important de l'intégration 3D est la possibilité d'intégrer des technologies hétérogènes fabriquées par des processus différents au sein de la même puce

3D. Ceci veut dire la fabrication des différentes fonctions analogique, numérique ou mémoire de façons indépendantes, et leur intégration ultérieure dans le même système final. Il est alors possible de fabriquer chaque type de circuit en utilisant la technologie la plus adaptée (figure 3).

D'autre part, les technologies d'intégration 3D permettent d'empiler des circuits identiques. Il serait ainsi possible de réutiliser des masques déjà existants pour fabriquer une large gamme de systèmes, au lieu que de concevoir un nouvel ensemble de masque. Pour ce faire, une idée est de concevoir un circuit modulaire qui pourrait être empilé en utilisant les technologies d'intégration 3D afin construire plusieurs systèmes 3D avec des performances adaptées aux besoins de différentes applications. Ainsi, grâce à l'intégration 3D, il serait possible de concevoir plusieurs systèmes en utilisant toujours le même jeu de masque (figure 3).

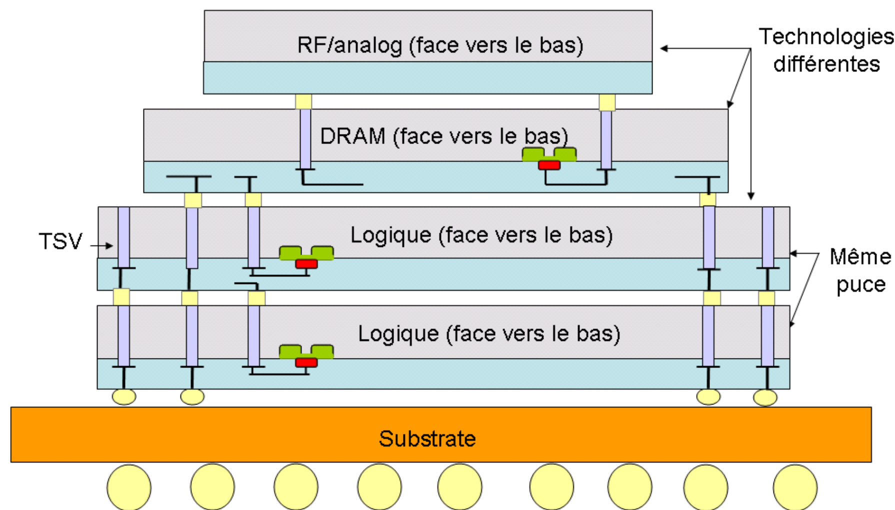


Figure 3 : Un système-sur-puce 3D intégrant à la fois des puces identiques et des puces différentes

Un autre avantage de l'intégration 3D est de remplacer les fils horizontaux longs par des interconnexions verticales courtes (TSVs) (figure 4). Ces fils courts vont diminuer la résistance et la charge capacitive moyenne (La capacité et la résistance d'un fil sont proportionnelles à sa longueur) et réduire le nombre de répéteurs nécessaires à l'amplification du signal. Puisque les fils d'interconnexion avec leurs répéteurs consomment une partie significative de la puissance totale, la réduction de leur longueur vont significativement réduire la consommation électrique globale au sein des circuits 3D.

En outre, les interconnexions courtes au sein des circuits 3D avec la réduction conséquente de leur charge capacitive et du nombre de leurs répéteurs permettra de réduire le bruit résultant de la commutation et du couplage entre les fils. Cela devrait fournir une meilleure intégrité du signal.

Une autre conséquence majeure de la réduction de la résistance et de la capacité du fil est la diminution significative du délai de propagation du signal (proportionnel au produit résistance fois la capacité), entraînant un gain considérable des performances du système.

Enfin, les technologies d'intégration 3D permettent de réduire la surface de la puce et d'améliorer ainsi son facteur de forme. Par conséquent, il serait possible de continuer la miniaturisation des puces sans nécessairement suivre la loi de Moore.

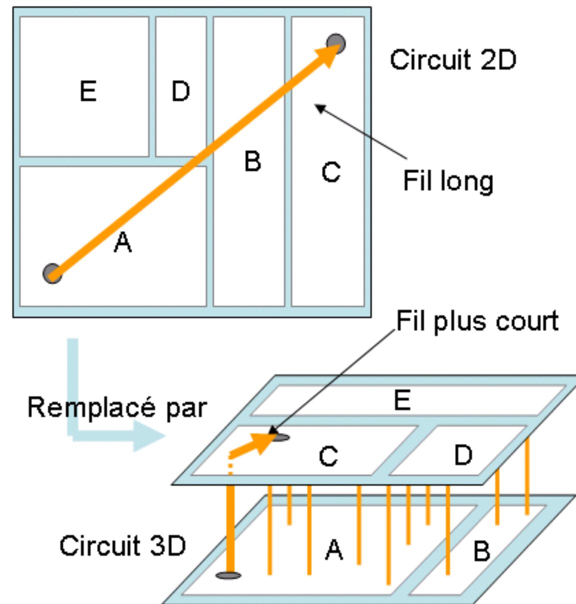


Figure 4 : La réduction des longueurs des interconnexions au sein des systèmes 3D

1.2 Fabrication des circuits 3D

Il existe de nombreuses options technologiques pour fabriquer un circuit 3D. D'abord, il est important de choisir la manière d'assembler les puces (figure 5) :

- puce-à-puce : dans ce cas, l'alignement est délicat, puisqu'il est difficile de manipuler les petites puces. En outre, l'assemblage des puces prend beaucoup de temps.
- puce-à-plaque : dans ce cas, le temps nécessaire pour l'assemblage des puces est beaucoup plus court. En outre, l'alignement est plus facile puisque l'assemblage est effectué sur des objets plus gros.
- plaque-à-plaque : le temps nécessaire pour l'empilement est moins considérable que dans l'option puce-à-puce. Le problème d'alignement est aussi moins critique.

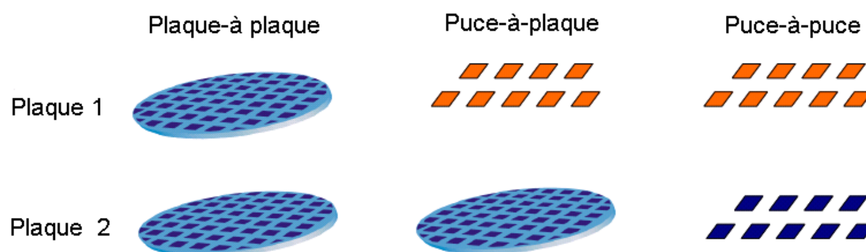


Figure 5 : Les trois façon d'assembler les circuits 3D

D'autre part, il est aussi nécessaire de de décider sur la façon dont les niveaux actifs sont orientés :

- face-à-face : cette option donne une haute densité d'intégration, puisqu'il n'y a pas de TSV dans la zone active. Toutefois, il n'est pas possible d'empiler plus que deux couches de cette façon.

- face-à-arrière : dans ce cas, la densité d'intégration est plus faible, mais l'empilage de plus de deux couches est possible.

Pour fabriquer un circuit 3D, certaines étapes technologiques clés doivent être réalisées : l'amincissement des substrats, le collage des substrats, et la formation des interconnexions verticales (TSVs).

1.3 Limitations des technologies d'intégration 3D

En dépit des avantages précédemment cités, l'intégration 3D est confrontée actuellement à certains problèmes majeurs qui entravent son application industrielle.

Problèmes thermiques

L'un des défis les plus critiques de l'intégration 3D est la dissipation de chaleur. En effet, dans les circuits 3D, plusieurs puces sont emballés dans un petit volume, entraînant une augmentation rapide de la densité de puissance, en particulier si les niveaux empilés sont très actifs (typiquement des circuits logiques). Les hautes températures ont deux inconvénients majeurs : elles peuvent limiter les fréquences de fonctionnement et dégrader la fiabilité des puces empilées. Ces deux facteurs ont fait de la gestion thermique un problème critique pour les circuits 3D.

Surcoût en surface du aux TSVs

Avec les technologies actuelles de l'intégration 3D, le diamètre des TSVs peut varier de $1\mu m$ à $100\mu m$. Cette taille est considérée comme énorme par rapport à la taille des transistors. Par exemple, la surface d'un TSV de diamètre $5\mu m$ est supérieur à la celle de 500 cellules SRAM en technologie 45nm. Ainsi, il est important de réduire la superficie totale des TSVs en minimisant leur taille (rôle du fabricant) et leur nombre (rôle du concepteur) afin de réduire le coût des circuits 3D.

Les problèmes de test

Un obstacle majeur pour les technologies d'intégration 3D est le manque de savoir-faire à propos des problèmes de test des circuits 3D. En effet, les techniques de conception pour la testabilité des circuits intégrés 3D ne sont pas encore suffisamment explorés par la communauté de recherche. En outre, plusieurs autres obstacles sont identifiés tels que notamment l'accès aux modules de test des palques empilés, le coût de test, et les nouveaux défauts résultant des étapes de fabrication spécifiques à l'empilement 3D...

2 Des architectures 2D aux architectures 3D : problèmes de partitionnement et de coût

Lors de la conception d'un nouveau circuit numérique en 3D, il est nécessaire d'analyser les différentes possibilités de partitionner le circuit 2D original. L'approche de partitionnement doit être choisie afin de tirer parti du potentiel de l'intégration 3D. En outre, elle doit tenir compte des limites économiques et techniques décrites

dans la première section. Ainsi, l'analyse de coût au plus tôt au cours du cycle de conception d'un circuit numérique 3D s'avère très importante afin de déterminer les options technologiques les plus économiques.

2.1 Problèmes de partitionnement

Le partitionnement d'un circuit numérique 2D à travers deux ou plusieurs couches peut être réalisé selon 3 granularités différentes : bloc macroscopique, unité fonctionnelle et opérateur de base (figure 6). Chacune de ces granularités de partitionnement a ses avantages et ses limites. Choisir une de ces possibilités de partitionnement dépend du compromis performance-coût choisi.

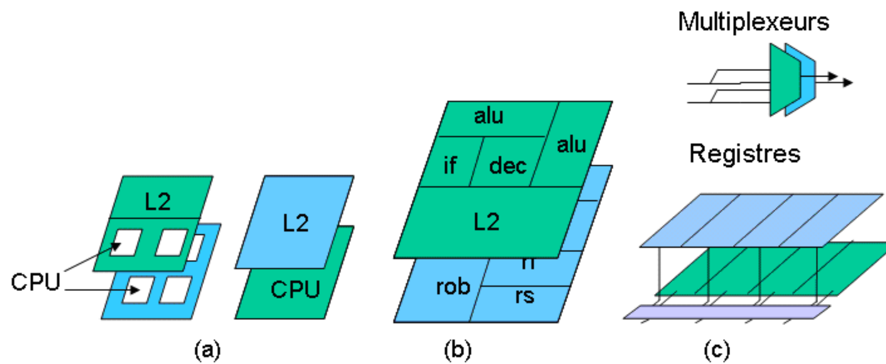


Figure 6 : Les 3 granularités de partitionnement : (a) les macro-blocs, (b) les unités fonctionnelles et (c) les opérateurs élémentaires

La première est au niveau du bloc macroscopique : il s'agit par exemple d'empiler une mémoire cache ou une mémoire principale au dessus d'un processeur, ou d'empiler un processeur au-dessus d'un autre processeur. L'empilement de la mémoire au dessus du processeur permet d'améliorer les performances du système 3D résultant et réduire significativement sa puissance globale, par rapport aux une implémentation 2D classique. Étant donné que la mémoire n'est pas une couche très active, les températures atteintes dans ce type d'architecture sont imperceptiblement plus élevés que celles des circuits 2D.

Il est également possible de partitionner un circuit selon les unités fonctionnelles : il s'agit dans ce cas de partitionner le processeur lui-même sur deux ou plusieurs couches. Cela signifie par exemple l'empilage de l'unité arithmétique et logique directement au-dessus du fichier de registre. A la différence du premier type de partitionnement qui n'affecte pas le processeur lui-même, l'empilement des unités fonctionnelles permet d'améliorer considérablement les performances et diminuer la puissance du processeur 3D. Toutefois, il peut provoquer une augmentation significative de la température du processeur 3D, et affaiblir ainsi sa fiabilité. Il est donc nécessaire d'étudier profondément les problèmes thermiques associés à ce type de partitionnement afin de limiter leurs impacts.

La granularité de partitionnement la plus fine consiste à empiler des opérateurs de base tels que les multiplexeurs et les portes logiques. Cela permet le partitionner les unités fonctionnelles comme le fichier de registre à travers plusieurs couches. Ce type de partitionnement nécessite la re-conception complète des unités fonctionnelles. Il permet de réduire les longueurs des interconnexions à l'intérieur des unités

fonctionnelles, et d'améliorer ainsi leurs performances. En outre, la consommation d'énergie est diminuée proportionnellement à la prépondérance des fils dans l'unité fonctionnelle. Enfin, la densité de puissance augmente considérablement par rapport aux autres types de partitionnement. Ainsi, les contraintes thermiques sont plus importantes.

En guise de conclusion, les 3 types de partitionnement permettent d'améliorer la performance et diminuer la puissance consommée par des taux qui augmentent avec la diminution de la granularité de partitionnement. En même temps, des problèmes tels que les efforts de re-conception, les contraintes thermiques, le surcoût en surface du aux TSVs s'accentuent de plus en plus en miniaturisant la taille des éléments empilés. Compte tenu de ces limitations, l'empilement des macro-blocs semble être l'approche la plus viable économiquement et technologiquement, puisqu'il nécessite le moins d'efforts de conception, induit la plus faible augmentation de température et s'adapte le mieux aux tailles actuelles des TSVs.

Pour cette raison, cette thèse est orientée vers l'empilement des macro-blocs. A ce niveau de partitionnement, l'implication de d'intégration 3D est limitée aux interconnexions verticales inter-blocs. Un premier apport de cette thèse traite des architectures d'interconnexion au sein des systèmes 3D partitionnés au niveau des macro-blocs, et propose une nouvelle topologie de réseaux-sur-puce 3D qui permet l'amélioration des performances. Une présentation détaillée de ce travail est fourni dans le chapitre 3 de la thèse.

2.2 Problèmes de coût

L'intégration 3D est réalisée en plusieurs étapes, dont chacune comprend un large éventail de choix technologiques. Choisir le processus de fabrication optimal dépend principalement du coût. Ainsi, l'estimation du coût des circuits intégrés 3D dès les premiers stades du cycle de conception est importante. Cette section présente une analyse de coût au niveau système, permettant d'évaluer la rentabilité économique des différentes options technologiques de l'intégration 3D.

En plus des approches d'empilage 3D puce-à-puce et puce-à plaque précédemment décrites, nous évaluons une autre approche de l'intégration 3D appelée l'empilement à base d'interposeur. Cette technologie consiste à empiler plusieurs puces (fabriquées avec une technologie agressive telle que la 32nm) sur un interposeur en silicium (fabriqué avec une technologie mature comme la 130nm) pour créer un circuit 3D. Ces puces sont mises côte à côte sur l'interposeur et interconnectées par un très grand nombre de fils afin de fournir une large bande passante inter-puce. L'empilement à base d'inter-poseur permet d'éviter les problèmes de fiabilité qui peuvent résulter de l'empilement de plusieurs puces (fabriquées avec une technologie agressive) l'une dessus de l'autre.

Une description détaillée du modèle de coût proposé et des hypothèses adoptées est fournie dans le chapitre 2 de la thèse. En utilisant le modèle d'estimation de coût proposé, et en faisant varier certaines caractéristiques du circuit étudié telles que la surface et le nombre de couches, des résultats intéressants sont conclus. D'abord, l'intégration 3D est plus efficace que l'approche 2D classique en termes de coûts dans le cas des circuits de grande taille, en utilisant le schéma d'intégration puce-à-plaque ou l'empilement à base d'inter-poseur. Ceci est dû au rendement faible des

circuits de grande taille. En outre, en utilisant toujours le schéma d'intégration puce-à-plaque ou l'empilement à base d'inter-poseur, le nombre optimal de couches 3D (en termes de coût) dépend de la surface du circuit. Plus la surface est large plus le nombre optimal de couches est important.

La deuxième contribution de cette thèse porte sur les architectures 3D rentables économiquement. En utilisant ce même modèle de coût, nous évaluons les gains économiques des architectures 3D résultant de l'empilement de puces identiques, avec une étude de cas réel sur les applications de télécommunications 4G. Plus de détails sur ce travail sont indiqués dans le chapitre 4.

3 Un réseau-sur-puce hiérarchique pour les architectures multiprocesseurs 3D

Pour des raisons de viabilité économique et technique, cette thèse porte sur le partitionnement au niveau des macro-blocs. A cette granularité de partitionnement, l'implication de la technologie d'intégration 3D est limitée aux interconnexions verticales inter-blocs. Compte tenu de la complexité croissante des architectures multiprocesseurs, et de l'importance des besoins de communication dans les circuits numériques 3D, les réseaux-sur-puce 3D sont de plus en plus considérés comme une solution prometteuse permettant de répondre aux exigences des circuits multiprocesseur 3D en termes d'architecture d'interconnexion.

Le réseau-sur-puce a été proposé pour résoudre les problèmes de réutilisabilité et d'extensibilité dont souffre les autres types d'interconnections tels que les bus et les connexions point-à-point (ou directs). Le réseau-sur-puce fournit une architecture de communication qui peuvent être normalisée et donc facilement réutilisable. Par ailleurs, le réseau-sur-puce présente une architecture de communication distribuée sans contrôle global sur l'état du système. Par conséquent, il permet d'intégrer un grand nombre de modules ayant des fonctionnalités différentes, tout en gardant la même bande passante.

Le réseau-sur-puce est composé d'un ensemble de nœuds (ou routeurs) connectés par des liens de communication directes (ou canaux). Chaque nœud comporte un ensemble de ports qui lui permettent de se connecter à ses nœuds voisins mais aussi à des éléments fonctionnels du réseau (les ressources) (figure 7). Les ressources communiquent en échangeant des messages. Au niveau du réseau-sur-puce, le message est appelé un paquet. Un paquet est constitué d'un en-tête qui, souvent, contient des informations sur la nature du paquet et sa destination, d'un champ de données qui comprend les informations utiles menées par le paquet et d'un champ qui indique la fin du paquet.

Récemment, plusieurs études ont été menées afin de développer des architectures d'interconnexions efficaces pour les circuits multiprocesseurs 3D, basées principalement sur l'approche réseau-sur-puce. Une façon simple d'étendre le routeur 2D 5x5 (5 ports d'entrée, 5 ports de sortie) à la dimension verticale consiste à ajouter un port pour le trafic vers la couche d'en haut et un port pour le trafic vers la couche d'en bas (figure 8). Bien sûr, il serait nécessaire d'étendre également les autres composants du routeur tels que les buffers, les arbitres, et le crosssbar. Cette architecture maillée 3D est simple à concevoir et à implémenter, mais elle a un problème

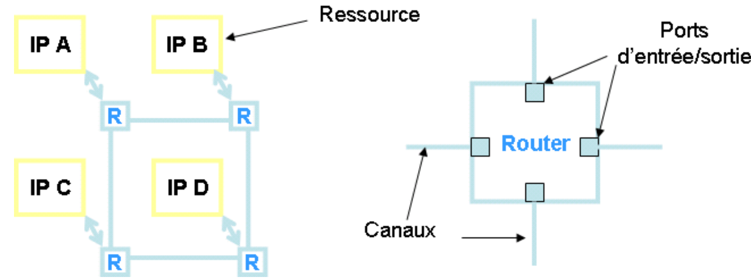


Figure 7 : Un réseau-sur-puce

majeur. En effet, l'ajout de deux ports à chaque routeur demande un crossbar 7x7 plus grand, ce qui implique des augmentations significatives de la surface, la puissance consommée et la latence par rapport au routeur 2D. Beaucoup de recherches ont été menées pour améliorer les performances des réseaux maillés 3D. Ces travaux visent soit l'amélioration du routeur soit l'optimisation de la topologie.

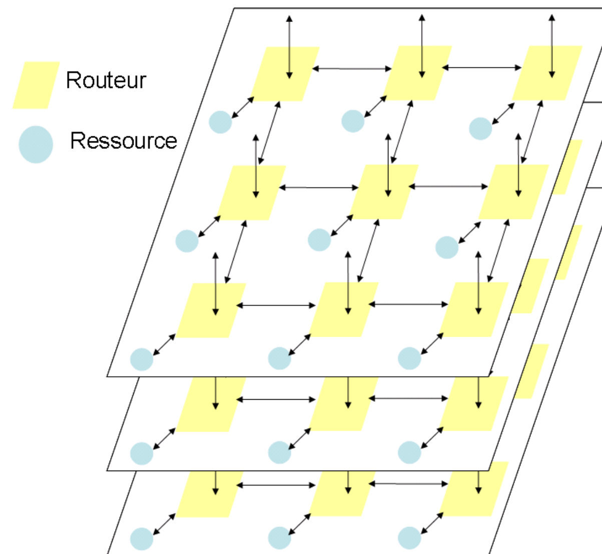


Figure 8 : Un réseau-sur-puce maillé 3D

3.1 Le réseau-sur-puce hiérarchique 3D

Le réseau-sur-puce hiérarchique 3D proposé dans ce travail de thèse consiste à remplacer le routeur 7x7 par 2 routeurs totalement découplés : un routeur 5x5 (appelé le module horizontal) utilisé pour les communications intra-couche, et un routeur 4x4 (appelé le module vertical) utilisé pour les communications inter-couche. Tous les routeurs 7x7, 5x5 et 4x4 ont la même micro-architecture ; la seule différence entre eux est le nombre de ports d'entrée sortie. Les routeurs 5x5 et 4x4 communiquent entre eux. L'élément de traitement (la ressource) est connecté au routeur 4x4. Une vue d'ensemble du routeur hiérarchique proposé et du routeur 7x7 équivalent est illustrée dans la figure 9.

L'architecture du réseau-sur-puce hiérarchique 3D est représentée dans la figure 10. Pour les communications verticales, un flit qui transite entre les couches doit traverser un routeur 4x4 au lieu d'un router 7x7 dans les cas du réseau-sur-puce maillé 3D. Pour les communications horizontales, supposons qu'un flit doit traverser

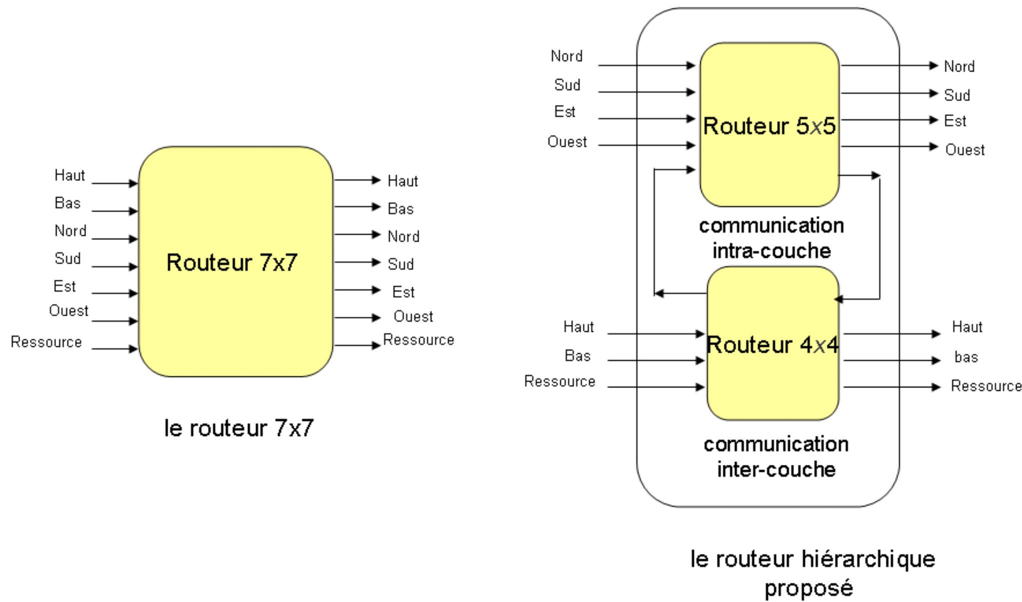


Figure 9 : Vue conceptuelle du routeur hiérarchique

n routeurs. Dans le cas d'un réseau-sur-puce maillé 3D, ce flit doit traverser n routeurs 7x7. Dans le cas du réseau-sur-puce hiérarchique proposé, il doit traverser un routeur 4x4, n routeurs 5x5 et finalement un routeur 4x4. Nos simulations en SystemC-TLM montrent que le réseau-sur-puce hiérarchique proposé est plus efficace que le réseau maillé en raison du fait que la latence intrinsèque (délai de propagation du flit) du routeur 7x7 est supérieur à celles du 4x4 ou du 5x5.

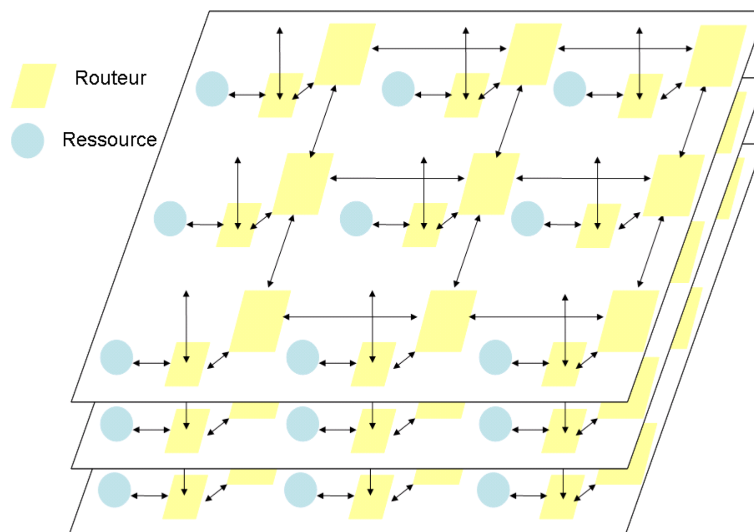


Figure 10 : Réseau-sur-puce hiérarchique 3D

Le réseau-sur-puce hiérarchique 3D est conçu comme un système Globalement Asynchrone Localement Synchrone : les éléments de traitement sont des unités synchrones, tandis que les routeurs sont mis en œuvre en logique asynchrone quasi-insensible au délai. Chaque élément de traitement est connecté au routeur via une interface réseau qui assure la synchronisation entre les domaines synchrone et asynchrone.

3.2 Résultats de surface et de consommation

Le tableau 1 présente les résultats en termes de surface et de puissance consommée des les routeurs 7x7, 5x5 et 4x4. Le routeur 5x5 a été déployé et fabriqué en utilisant la technologie CMOS 65nm de STMicroelectronics dans une puce dédiée au traitement en bande de base des applications de télécommunications sans fil 3G/4G. Les résultats du routeur 5x5 sont extraits de cette puce. Les résultats des routeurs 7x7 et 4x4 les sont extrapolés à partir de ceux du routeur 5x5, en se basant sur notre connaissance de micro-architecture des routeur. En se référant aux résultats du tableau 1, le routeur hiérarchique a presque la même surface et puissance consommée que le routeur maillé 3D.

Tableau 1 : Résultats de surface et de consommation des routeurs 4x4, 5x5 et 7x7 en technologie 65nm

Routeur	Surface (mm^2)	Consommation (mW)
4x4	0.12	8.66
5x5	0.17	11.9
7x7	0.28	19.65

3.3 Résultats de performance

Plateforme de simulation

Afin de modéliser et de simuler le réseau-sur-puce hiérarchique 3D, une classe représentant le routeur asynchrone est développée en utilisant le langage SystemC-TLM. Le but principal de la modélisation TLM en SystemC est d'accélérer la simulation. Pour ce faire, l'idée principale consiste à modéliser les transactions du système à un niveau d'abstraction élevé. Par conséquent, le protocole proposé est modélisé au niveau du flit.

Le modèle TLM de routeur attend les transactions entrantes, arbitre entre ces transactions en fonction de l'état de sortie de chaque nœud (la politique de canal virtuel), attend un certain temps (pour modéliser le délai réel du nœud). Le transfert des flits vers la sortie du nœud est effectué par une simple boucle *for*. Puisque ce modèle est générique en termes de nombre de ports d'entrée sortie, il pourrait facilement être utilisé pour simuler tous les routeurs 4x4, 5x5 et 7x7. Un tel modèle fonctionnel TLM du routeur est extrêmement rapide dans la simulation. En outre, il est suffisamment précis pour obtenir une évaluation correcte des performances du système.

Dans ce travail, nous utilisons 2 types de trafic de données : le trafic aléatoire uniforme et le trafic transposé. Le banc de test utilisé pour la simulation est composé de 256 routeurs (32 routeurs x 8 couches) connecté à 256 générateurs de trafic. Les générateurs de trafic injectent des paquets avec des données aléatoires dans le réseau en activant aléatoirement les canaux virtuels. La longueur d'un paquet (nombre de flits par paquet) est fixé à 4.

Débit

La charge d'injection signifie la vitesse à laquelle les blocs fonctionnels injectent des données dans le réseau. Le débit désigne la capacité des données injectées à transiter à travers le réseau entre le bloc fonctionnel source et le bloc fonctionnel destination.

La figure 11 montre la variation du débit en fonction de la charge d'injection pour les deux types de trafic : uniforme et transposé. Par rapport au réseau-sur-puce maillé 3D, la valeur de saturation du réseau-sur-puce hiérarchique est supérieure de 15% dans le cas d'un trafic uniforme, et de 25% dans le cas d'un trafic uniforme transposé, à celle du réseau-sur-puce maillé 3D. Ainsi, nous pouvons conclure que le réseau-sur-puce hiérarchique est plus performant que le réseau-sur-puce maillé 3D en terme de débit.

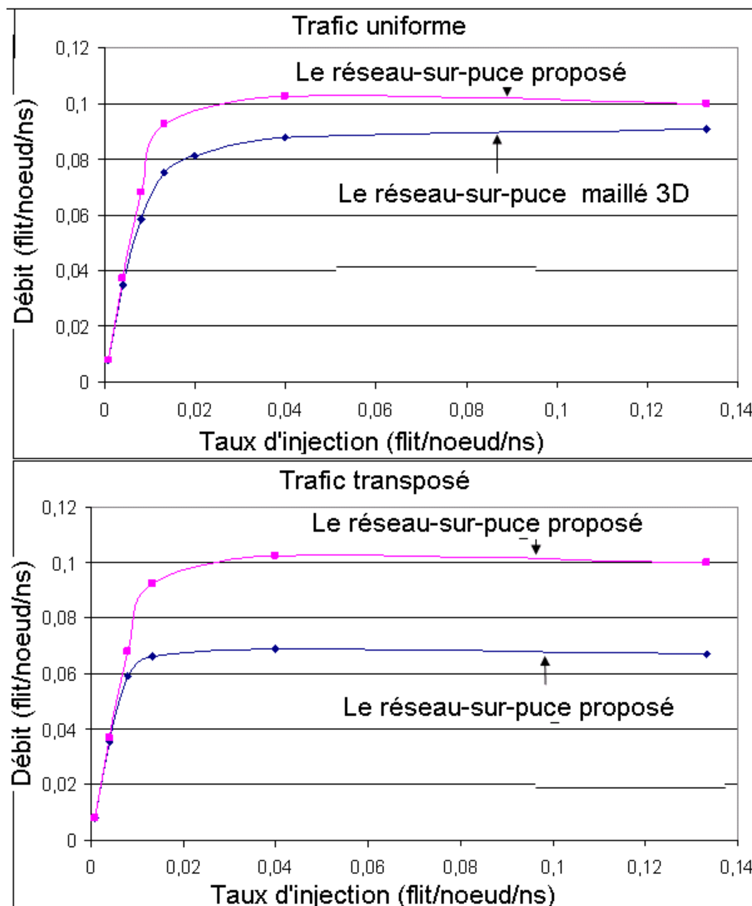


Figure 11 : Variation du débit pour différents types de trafic

Latence

La latence moyenne est définie comme le temps moyen qui s'écoule entre l'injection de l'entête du message dans le réseau par le routeur source, et la réception du dernier flit du message par le routeur destination.

La figure 12 montre la variation de la latence moyenne du réseau en fonction de la variation du débit. Par rapport au réseau-sur-puce maillé, le réseau-sur-puce hiérarchique présente une latence moyenne inférieure de 15% lors de l'utilisation du trafic uniforme, et de 25% lors de l'utilisation du trafic transposé. Ainsi, nous

pouvons conclure que le réseau-sur-puce hiérarchique est plus performant que le réseau-sur-puce maillé classique en en terme de latence.

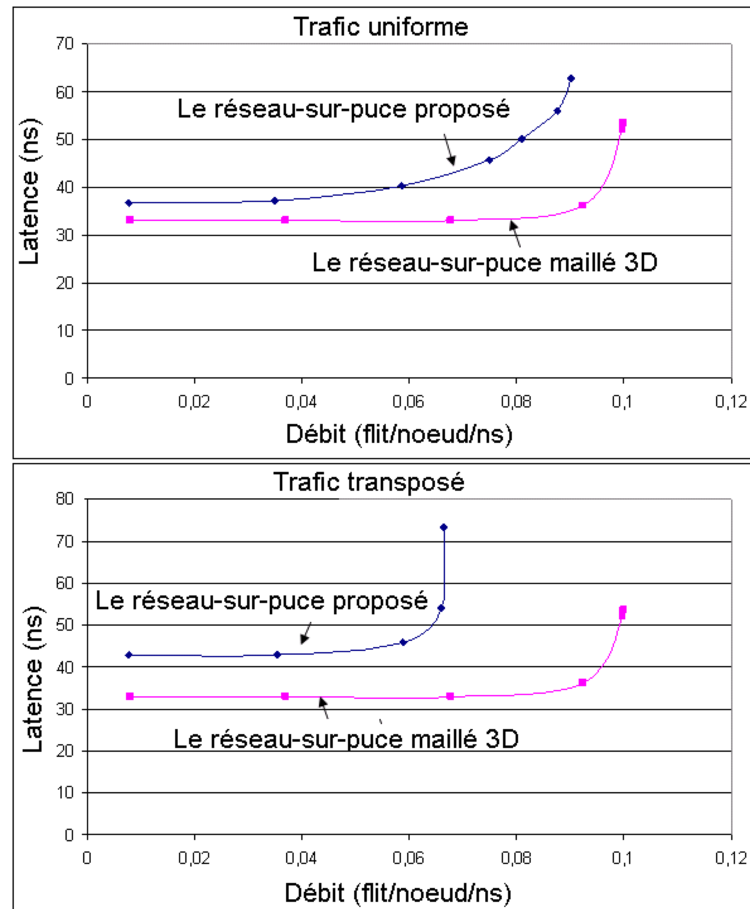


Figure 12 : Variation de la latence pour différents types de trafic

4 Un circuit empilable pour les applications de télécommunications 4G

De nos jours, les concepteurs des systèmes-sur-puce complexes sont confrontés à plusieurs problèmes qui limitent la compétitivité économique des produits fabriqués avec des nœuds technologiques de pointe. En plus du coût prohibitif des masques (qui a déjà dépassé 1 million d'euros), le coût de fabrication est de plus en plus considérable, principalement en raison de la taille énorme des circuits qui réduit le rendement de fabrication. Une solution pour développer des produits économiquement compétitifs consiste à réutiliser le même jeu de masque pour réaliser un large éventail de systèmes, et de fabriquer des circuits de petites tailles afin d'augmenter le rendement.

Pour ce faire, notre proposition est de concevoir un circuit modulaire qui pourrait être empilé grâce aux technologies d'intégration 3D afin de construire des systèmes 3D avec des puissances de calcul adaptées aux besoins de plusieurs applications (figure 13). Dans ce travail, nous nous concentrons sur les architectures modulaires pour les applications de télécommunications 4G, la dernière norme de la téléphonie

mobile. Nous proposons un circuit reconfigurable qui peut adresser les besoins du mode de transmission avec une seule antenne en émission et une seule antenne en réception (*SISO: Single Input Single Output*). En empilant plusieurs instances de ce même circuit, il serait possible d'augmenter les performances du système global et d'adresser plusieurs modes à plusieurs antennes en émission et plusieurs antennes en réception (*MIMO: Multiple Input Multiple Output*).

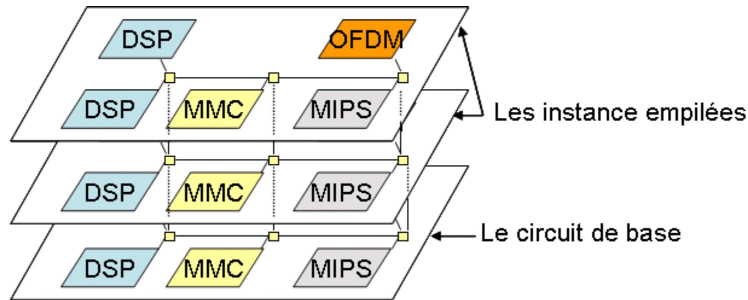


Figure 13 : Circuit 3D obtenu par l'empilement de plusieurs instances d'un même circuit

4.1 L'implémentation des terminaux 4G

Dans cette section, nous nous concentrons uniquement sur le traitement en bande de base de la chaîne de réception, et omettront tous les autres composants du terminal 4G comme les fonctions radiofréquence et analogiques, les protocoles des couches supérieures et les traitements multimédias. La figure 14 représente un schéma fonctionnel des données internes de la chaîne de réception au sein d'un terminal 4G avec quatre antennes de réception. Tout d'abord, le signal radio-fréquence est reçu par les antennes de réception, converti en une grandeur électrique, et numérisé par un convertisseur analogique-numérique. Ensuite, le processeur de bande de base reçoit le signal numérisé à titre d'échantillons complexes et effectue la démodulation OFDM (*Orthogonal Frequency-Division Multiplexing*), l'estimation de canal et le décodage MIMO.

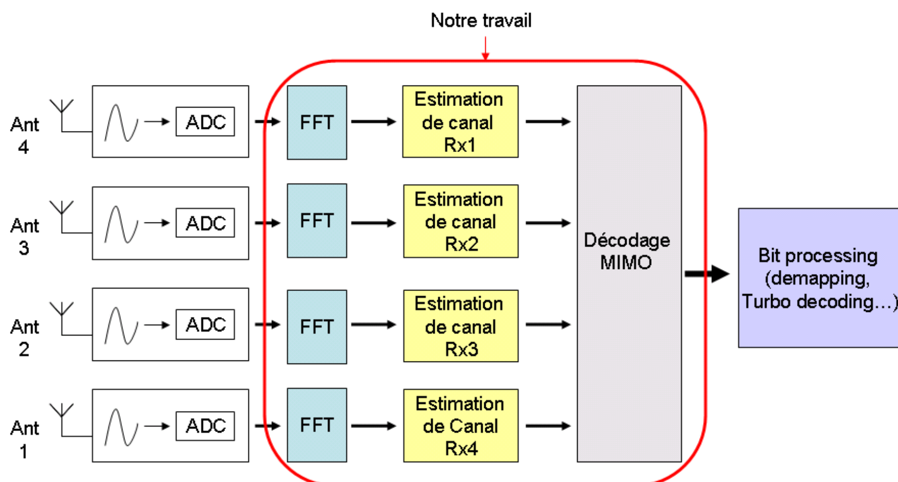


Figure 14 : Un schéma fonctionnel de la chaîne de réception d'un terminal 4G avec quatre antennes

L'analyse des algorithmes nous a permis de déduire le nombre d'additions et de multiplications requis pour effectuer l'estimation du canal et le décodage MIMO pour les différents modes 4G : 4x4 (4 antennes en émission x 4 antennes en réception), 4x2, 2x2, 2x1 et 1x1 (tableau 2).

Tableau 2 : La complexité théorique de l'estimation de canal et du décodage MIMO pour les différents modes 4G

Mode de transmission	Estimation de canal		Décodage MIMO	
	(+)	(×)	(+)	(×)
4x4	572	1,328	179	376
4x2	286	664	21	34
2x2	18	40	11	22
2x1	9	20	3	11
1x1	0	1	0	1

Quand il s'agit de l'implémentation réelle de l'estimation de canal et du décodage MIMO, les algorithmes sont simplifiés afin de réduire les efforts de calcul et la consommation électrique des terminaux 4G. Par conséquent, les complexités théoriques présentées dans les tabelaux 3 et 4 ne correspondent pas aux besoins réels d'un terminal 4G. Néanmoins, dans ce travail nous considérons ces complexités théoriques afin d'avoir une limite supérieure des exigences de performance des terminaux 4G.

4.2 Un circuit empilable pour les applications 4G

Dans cette section, nous présentons un circuit reconfigurable et empilable pour les applications 4G. Le circuit proposé (ci-après appelé circuit de base) peut répondre aux exigences du mode de transmission SISO. En empilant plusieurs instances de ce circuit de base et en réalisant des configurations logiciels, il sera possible d'accroître les performances du système et d'adresser plusieurs modes MIMO.

Le circuit proposé est composé de 5 éléments interconnectés grâce à un réseau-sur-puce. Le premier composant est un contrôleur de mémoire programmable conçus pour effectuer la synchronisation et la distribution des données dans les systèmes de type flot de données. Le deuxième composant est le bloc OFDM utilisé pour effectuer une transformée de Fourier rapide direct et inverse (FFT et IFFT). Le circuit proposé comporte également 2 processeurs de signal numérique dédiés au calcul matriciel, utile pour l'estimation de canal, le décodage MIMO... Le contrôle global des unités précédemment décrites est réalisé par un processeur MIPS 32-bit, par le biais de mécanismes d'interruption et d'adressage direct. Il est en charge de des reconfigurations dynamiques, de l'ordonnancement en temps réel et des synchronisations. Lorsque plusieurs circuits de base sont empilés (pour construire un système 3D), le contrôle global est distribué entre tous les processeurs MIPS du système 3D résultant. Toutes les unités du circuit de base sont interconnectées via un réseau-sur-puce. Ce réseau-sur-puce est également utilisé pour connecter tous les composants d'un système 3D résultant de l'empilement de 2 ou plusieurs circuits de base. Le circuit base est conçu comme un système Globalement Asynchrone Localement Synchrone (*GALS: Globally Asynchronous Locally Synchronous*) : les unités de traitement sont synchrones (chacun a sa propre fréquence de l'horloge), tandis que les routeurs du réseau-sur-puce sont implémentés en logique asynchrone

quasi-insensible au délai. L'approche GALS permet d'éviter les problèmes liés à la distribution du signal d'horloge globale au sein du système 3D.

Les résultats de synthèse en technologie 65nm montrent que le circuit de base a une surface totale de $3.6mm^2$. La surface des TSVs (de diamètre $4\mu m$ et de pitch $10\mu m$) correspond à 1,1% de la taille du circuit de base. Le processeur MIPS chargé du contrôle occupe moins de 5% de la surface du circuit proposé. Le réseau-sur-puce représente 17% de la superficie du circuit.

L'analyse des performance des unités constituant le circuit de base nous a permis de déduire les nombres d'instances du circuit de base nécessaires pour implémenter les différents modes de transmission 4G (tableau 3).

Tableau 3 : Les nombres d'instances du circuit de base nécessaires pour implémenter les différents modes 4G

Mode de transmission	Nombres d'instances
4x4	10
4x2	2
2x2	1
2x1	1
1x1	1

Pour évaluer les performances du circuit proposé, le mode de transmission 4x2 (4 antennes en émission et 2 antennes en réception) a été implémenté sur un système 3D résultant de l'empilement de 2 instances du circuit de base. Afin d'établir une comparaison entre l'architecture 3D proposée et une architecture 2D de référence, ce même mode de transmission a été implémenté sur une architecture 2D composé des mêmes unités de traitement.

L'environnement de simulation comprend 2 générateurs de données implémentés en SystemC-TLM permettant d'imiter les données entrantes des 2 antennes. Pour accélérer la simulation, le réseau-sur-puce asynchrones est modélisé en SystemC-TLM avec des paramètres post-layout (le même modèle décrit dans la section 3). Toutes les unités de traitement des 2 plates-formes sont modélisées au niveau RTL pour fournir des résultats précis au cycle près.

Les résultats de simulation montrent que la performance et la consommation d'énergie du système 3D est comparable à celles du système 2D.

En outre, en se basant sur le même modèle de coût présenté dans la section 2, l'approche consistant à empiler des puces identiques pour les applications 4G est rentable en terme de coût (par rapport à l'approche 2D) dans certains cas, avec le schéma d'intégration plaque-à-plaque.

Les principales limitations de l'approche consistant à empiler des puces identiques sont les contraintes thermiques dues à l'empilement de plusieurs couches de logique très actives. Les températures élevées peuvent limiter les fréquences de fonctionnement des blocs logiques empilés, et dégrader la fiabilité du système 3D.

Conclusion

La technologie 3D d'intégration consiste à empiler des circuits verticalement et les interconnecter par des vias qui traversent les couches de silicium (TSVs). Il en

résulte un circuit de plus petite empreinte ayant des interconnections plus courtes. Les avantages potentiels de l'intégration 3D sont : une performance accrue, une puissance réduite, un petit facteur de forme, un meilleur rendement et un coût global réduit. Tous ces avantages font de l'empilement 3D une solution prometteuse pour surmonter les limitations actuelles des technologies CMOS agressives, tels que les gains limités en performance, les problèmes de fiabilité et les coûts de fabrication prohibitifs.

Plusieurs options technologiques se présentent lors de la fabrication d'un circuit intégré 3D. Il est important de choisir la manière d'assembler les puces (puce-à-puce, puce-à-plaque, ou plaque-à-plaque) et de décider sur la façon dont les niveaux actifs sont orientés (en face-à-face, en face-à-arrière). La fabrication comprend 3 grandes étapes : l'amincissement des plaques, le collage, et la formation des TSVs. En dépit de ses avantages en termes de coût et de performance, l'intégration 3D est confrontée à des défis majeurs. Le premier problème concerne la gestion des contraintes thermique causées par l'augmentation de la densité de puissance. D'autres problèmes importants sont : le surcoût important en surface active du aux TSVs, l'insuffisance des outils de CAO et les problèmes de test.

Le partitionnement des circuits sur plusieurs couches peut être effectué selon 3 granularités différentes : les macro-blocs, les unités fonctionnelles et les opérateurs de base. Le partitionnement selon ces 3 granularités permet d'améliorer les performances et diminuer la puissance consommée par des taux qui augmentent avec la diminution de la granularité de partitionnement. En outre, les problèmes liés à l'intégration 3D tels que les efforts de re-conception et les contraintes thermiques deviennent de plus en plus perceptible en réduisant la taille des éléments empilés. Tenant compte des limitations économique et technique de chacune de ces 3 granularités, l'empilement des macro-blocs semble être l'approche la plus viable. Pour cette raison, cette thèse est orientée vers cette granularité de partitionnement, et plus particulièrement vers les architectures d'interconnexion au sein des systèmes 3D partitionnés selon les macro-blocs. Dans ce contexte, nous proposons une nouvelle topologie de réseau-sur-puce 3D qui permet d'améliorer les performances des communications entre les macro-blocs.

Lors du partitionnement d'un circuit 2D pour obtenir sa version 3D, il est important également de savoir si ce partitionnement est rentable économiquement. Un modèle d'analyse de coût au niveau du système est proposé afin d'étudier la rentabilité économique de l'intégration 3D. En se basant sur ce modèle de coût, l'empilement 3D s'avère rentable (par rapport à l'approche 2D classique) dans les cas des circuits de grande taille, en utilisant les schémas d'intégration puce-à-plaque ou à base d'interposeur. Une autre observation est que le nombre optimal de couches (en termes de coût) dépend de la taille du circuit. La deuxième contribution de cette thèse porte sur les architectures 3D rentables économiquement. En utilisant ce même modèle de coût, nous analysons la rentabilité de l'approche consistant à empiler des puces identiques, avec une étude de cas réel sur les applications de télécommunications 4G.

En ce qui concerne l'architecture d'interconnexion au sein des systèmes 3D partitionnés au niveau des macro-blocs, notre contribution est de proposer et d'évaluer un nouveau routeur hiérarchique pour les réseaux-sur-puce 3D. Ce routeur est entièrement implémenté en logique asynchrone. Il est composé de 2 modules totalement

découplés : un pour les communications intra-couches et un pour les communications inter-couches. Notre étude montre que le routeur hiérarchique a presque la même surface et consommation que le routeur à 7 ports d'entrée/ sortie (utilisé dans les réseaux-sur-puce maillés 3D). Un simulateur rapide développé en SystemC-TLM est utilisé pour évaluer les performances du réseaux-sur-puce proposé en termes de débit et de latence. Les résultats de simulation montrent que le réseau-sur-puce hiérarchique peut dépasser le réseau-sur-puce maillé 3D de plus de 25% en termes de débit et de latence en cas de trafic transposé. L'architecture de réseau-sur-puce hiérarchique sera mise en œuvre au sein d'un véritable circuit démonstrateur 3D. Cela permettra de valider les résultats de simulation.

Concernant les architectures 3D composées de puces identiques, nous présentons un circuit reconfigurable et empilable pour les applications 4G. Le circuit proposé peut répondre aux exigences du mode de transmission SISO. En empilant plusieurs instances de ce circuit et en réalisant des configurations logiciels, il serait possible d'augmenter les performances du système et d'adresser plusieurs modes MIMO. Le circuit proposé est composé de 5 éléments interconnectés grâce à un réseau-sur-puce : un contrôleur de mémoire programmable, un bloc OFDM, et 2 processeurs de signal numérique. Le contrôle global de ces unités est réalisé par un processeur MIPS 32-bit, par le biais de mécanismes d'interruption et d'adressage direct. Il est en charge de des reconfigurations dynamiques, de l'ordonnancement en temps réel et des synchronisations. Lorsque plusieurs circuits de base sont empilés (pour construire un système 3D), le contrôle global est distribué entre tous les processeurs MIPS du système 3D résultant. Toutes les unités du circuit de base sont interconnectées via un réseau-sur-puce. Ce réseau-sur-puce est également utilisé pour connecter tous les composants d'un système 3D résultant de l'empilement de 2 ou plusieurs circuits de base. Pour évaluer les performances du circuit proposé, le mode de transmission 4x2 (4 antennes en émission et 2 antennes en réception) a été implémenté sur un système 3D résultant de l'empilement de 2 instances du circuit proposé. Afin d'établir une comparaison entre cette architecture 3D et une architecture 2D de référence, ce même mode de transmission a été implémenté sur une architecture 2D composée des mêmes unités de traitement. Les résultats de simulation montrent que la performance et la consommation d'énergie du système 3D est comparable à celles du système 2D. En outre, en se basant sur le même modèle de coût présenté dans le chapitre 2, l'approche consistant à empiler des puces identiques pour les applications 4G donne de bons résultats en termes de coût (par rapport à l'approche 2D) dans certains cas, avec le schéma d'intégration plaque-à-plaque.

Une première perspective de ce travail est de traiter des contraintes thermiques dans les circuits 3D composés de puces logiques identiques. En effet, les températures élevées peuvent limiter la viabilité de l'approche consistant à empiler des puces logiques identiques en limitant les fréquences de fonctionnement des blocs empilés et en dégradant la fiabilité du système 3D. Les solutions possibles à ce problème peuvent être d'ordre technologique (par l'insertion des vias thermiques) ou architectural (par l'amélioration de la gestion thermique grâce au mapping dynamique des tâches par exemple).

Une autre perspective est d'investiguer d'autres domaines d'application de l'approche consistant à empiler des puces identiques, qui soient plus susceptibles d'offrir de meilleurs résultats en termes de coût. Ces domaines d'application doivent exiger

des architectures multiprocesseurs de plus grandes tailles et de meilleures modularités (les processeurs généralistes par exemple).

Introduction

Context

The history of integrated circuits has begun since 1954, when the first silicon transistor was produced by Gordon Teal from Texas Instruments. A few years later, the first integrated circuit was invented by Jack Kilby in 1958 and consisted of only a transistor and other passive components fabricated in a single piece of semiconductor material. Since then, major progresses in the computing hardware industry has been achieved following the Moore's law. Around 1971, the first microprocessor the Intel 4004 was built of approximately 2,300 transistors. Nowadays, more than 60 years after the emergence of the first integrated circuit, several million transistors may be integrated on a single silicon chip, to make a whole electronic system.

Advances in the electronic industry have been led for decades by Moore's law that predicts the increase of transistor density. Nowadays, further scaling according to this empirical law seems to be uncertain and very expensive, due to several issues such as the exponential increase of leakage and the not-so-exciting gain in performance for aggressive technologies. 3D integration technology is emerging as a promising solution that allows continuing the miniaturization of integrated circuits without following Moore's law.

The laboratory of electronics and information technologies LETI has an important activity regarding 3D integration technologies. Indeed, several projects have been done to develop these promising technologies, and major progresses have been achieved. Besides, LETI has a significant expertise in the development of multiprocessor architectures devoted for 4G telecom applications. Since 2009, LETI has developed the MAGALI platform, which is a digital baseband circuit for software-defined-radio and cognitive-radio applications that aims the fourth-generation mobile phones.

This thesis is a part of the work carried out by LETI lab to investigate the opportunities provided by emerging technologies and to take advantage of them in order to improve the performance of future multiprocessor architectures dedicated for wireless telecom applications. These applications are becoming more and more demanding in terms of computational efforts and power consumption, especially with the coming out of the 4G standard.

Motivations and objectives

3D integration technologies are to stack many circuits vertically and connecting them using Through- Silicon-Vias (TSVs). This results in a smaller circuit footprint

and shorter vertical interconnections. The potential benefits of 3D integration may include multifunctionality, increased performance, reduced power, small form factor, reduced packaging, increased yield and reliability, flexible heterogeneous integration and reduced overall costs. All these benefits make 3D stacking a promising solution to overcome present limitations of aggressive CMOS technologies, such as limited performance gains, reliability problems, prohibitive fabrication cost...

Nevertheless, in order to take full advantage of the potential of 3D integration technologies in the case of digital architectures, it is mandatory to figure out the different possibilities to partition the original 2D circuit across two or more layers, and to deeply analyze the pros and cons of each possibility in terms of performance gain, power consumption, redesign effort, and fabrication cost. It is possible to partition a 2D digital circuit according to several granularities, each of which has benefits and drawbacks. The first objective of this thesis is to investigate each one of these partitioning options in order to identify the one that best suits our requirements in terms of performance and economical viability.

The partitioning analysis leads us to choose the core-level partitioning, and then to restrict the contribution of 3D stacking to the global inter-block vertical interconnections inside digital 3D architectures. Our second objective is to propose an enhanced NoC infrastructure that allows improving communication performance within core-partitioned 3D multiprocessor architectures.

In electronic industry, performance gain by itself could not be the determining factor when deciding to move from a fabrication process to another totally new. For instance, cost has been driving progresses in semi-conductor industry for the past decades, and will continue to do in the future. 3D integration technologies in turn obey to this rule: the orientation of the industrial community to these new technologies has to be imperatively justified by a considerable cost-effectiveness. The second objective of this thesis is to perform a cost analysis of 3D stacking technologies in order to be aware of their economical trends.

The cost analysis allows us identifying a cost-effective 3D stacking approach. Our final objective is to further investigate this approach within the context of wireless telecom applications. This exploration aims to apply the proposed approach on a real application case study in order to determine its consequences in terms of performance and power consumption.

Contributions

This work focuses on multiprocessor architectures for advanced wireless telecom applications based on 3D integration technologies. In order to respond to the objectives previously presented, several contributions have been achieved:

- a deep literature review about 3D integration technologies and architectures at the different partitioning granularities,
- a system-level cost analysis model to investigate the economical trends of 3D integration technologies,
- a novel 3D NoC topology to enhance the performance of inter-core communications in the case of core-level partitioned 3D architectures,

- a cost-aware stackable reconfigurable multiprocessor NoC-based architecture to address the requirement of 4G telecom applications.

Figure 1 illustrates the framework of this work concerning context, objectives, contributions and main results.

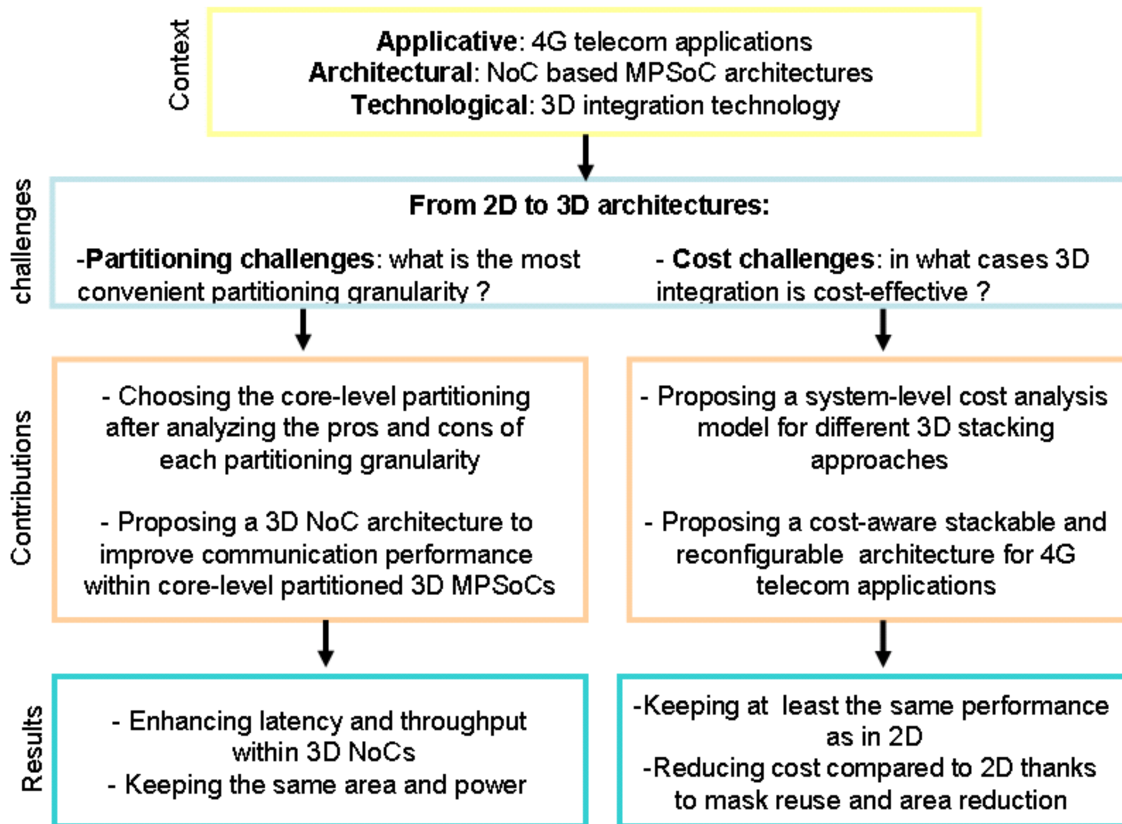


Figure 1: General PhD framework

Report organization

Besides this introduction, this report is composed of four chapters. The aim of the first one is to present an overview regarding 3D integration technologies. First, some details about recent technological trends are provided. Then, main motivations for 3D integration technologies are presented. After that, 3D manufacturing process steps and related issues (such as yield) and main limitations are described. Finally, some potential applications are introduced.

The second chapter is intended to deal with two main issues of 3D die stacking: partitioning granularity and cost-effectiveness. Thus, the first part of this chapter deals with partitioning aspects of 3D multiprocessor ICs: it introduces 3 levels of partitioning, and explains their pros and cons. The second part focuses on cost issues, and presents a system-level cost estimation model that we use to prove the cost-effectiveness of 3D architectures.

The third chapter focuses on improving the performance of inter-core communications in the case of core-level partitioned 3D multiprocessor architectures. A new 3D NoC router is proposed in order to enhance throughput and latency compared to classic 3D mesh NoC. The third chapter begins by explaining the motivations for NoC paradigm, and some of its main principles. Then, the pros and cons of several state-of-the-art 3D NoC architectures are presented. After that, the proposed 3D hierarchical NoC and the design of the asynchronous router are described in details. Finally, an evaluation of the proposed NoC in terms of area, power and performance is performed.

The fourth chapter presents a cost-aware stackable reconfigurable multiprocessor NoC-based architecture to address the requirement of 4G telecom applications. First, some aspect regarding the implementation of the digital baseband part of 4G terminals such as the different algorithms of the 4G standard and their computational requirements are described. Then, the proposed reconfigurable and stackable circuit is presented, and its adequacy to meet the performance needs of the 4G standard is proved. After that, reconfiguration and programming of the different hardware components of the stackable circuit are explained. Next, a case study of the 4x2 4G mode of transmission with performance and power evaluation is presented. Finally, the economical benefits of the proposed circuit are analyzed.

The last part of this report is dedicated to discuss conclusions and some perspectives of potential future works.

Publications

Book chapter

W. Lafi, D. Lattard, "3D integration and heterogeneous architectures", Heterogeneous Embedded Systems Design Theory and Practise, to be edited by I. O'Connor, G. Nicolescu, C. Piguet and to be published by Springer in 2011.

Journal paper

W. Lafi, D. Lattard, A. Jerraya, "An asynchronous hierarchical router for NoC-based 3D MPSoCs", minor rewriting, publication expected to the Software, Practice and Experience journal.

International Conference papers

- W. Lafi, D. Lattard, A. Jerraya, "A 3D reconfigurable platform for 4G telecom applications", International conference on Design, Automation and Test in Europe (DATE 2011), March 14-18, Grenoble, France.

- F. Clermidy, F. Darve, D. Dutoit, W. Lafi and P. Vivet, "3D embedded multicore: some perspectives", International conference on Design, Automation and Test in Europe (DATE 2011), March 14-18, Grenoble, France.

- W. Lafi, D. Lattard, A. Jerraya,"An efficient hierarchical router for 3D NoC architecture", 21st IEEE International Symposium on Rapid System Prototyping (RSP 2010), June 8-11, 2010, Fairfax, Virginia, USA.

- W. Lafi, D. Lattard, A. Jerraya, "High level modelling and performance evaluation of address mapping in NAND flash memory", The 16th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2009), December 13-16, 2009, Hammamet, Tunisia.

Patent

A patent was submitted to the French patent office under the reference 10 51517. It involves the hierarchical router for the NoC-based 3D architectures.

- Title: An integrated circuit die, and an integrated circuit including such an integrated circuit die.
- Inventors: Walid Lafi, Didier Lattard.
- Date: March, 2, 2010.

Chapter 1

3D integration technologies

The electronics industry has achieved an impressive development over the last decades, mainly due to rapid progresses in integration technologies. As more and more complex functions are required in present electronic devices dedicated for video processing and telecommunications for example, the need to integrate these functions in a small system/package and to reduce their power consumption is also increasing.

To address this challenge, current technological solutions are to continue system scaling according to Moore's Law (the number of transistors on a chip doubles every 18 to 24 months). An important consequence of this trend is the emergence of the System-on-Chip (SoC), which refers to integrating many functions into a single integrated circuit (chip).

Another direction of today's integrated circuit is the *More than Moore*, which is to use existing CMOS processes to develop new micro-devices referred to as System-in-Package (SiP). SiPs are composed of several circuits (with extended functionality such as sensors, MEMS, RF or analog) integrated in a single package.

Each of these two trends is facing several physical and technological problems, which limit their economical viability and the performance of their electronic products. To overcome these challenges, a potential solution is to merge the *More Moore* and the *More than Moore* approaches. The 3D die stacking seems to be a good technological support to achieve this.

The aim of this chapter is to introduce 3D integration technologies in terms of advantages, manufacturing process, limitations and potential applications. It is organized as follows. In section 1, details about current technological trends are given. Section 2 deals with the motivations for 3D integration technologies. An overview of 3D manufacturing process and related challenges such as yield are presented in section 3. Section 4 focuses on the well-known limitations of 3D stacking. Finally, some potential applications are introduced in section 5.

1 From SoCs and SiPs to 3D systems

Moore's law predicts that, thanks to advances in fabrication technology, chip's complexity (number of transistors) doubles every two years. Gordon Moore, a founder of Intel, described this trend in 1965, and slightly modified its formulation in 1975, to reach the previous statement. This empirical law has been surprisingly accurate and has been almost followed since 1973. It has allowed the semiconductor industry to define common goals of progress. An outlook written by an international consortium is provided at regular intervals by the ITRS association (International Technology Roadmap for Semiconductors). Thanks to this outlook, engineers have been able to anticipate technological opportunities in order to design new integrated electronic systems. Moore's law has contributed to the accelerated pace of innovation, particularly in the field of computer engineering. Figure 1.1 illustrates the evolution of microprocessor complexity (usually represented by the total number of transistors per chip) over the last thirty years [2].

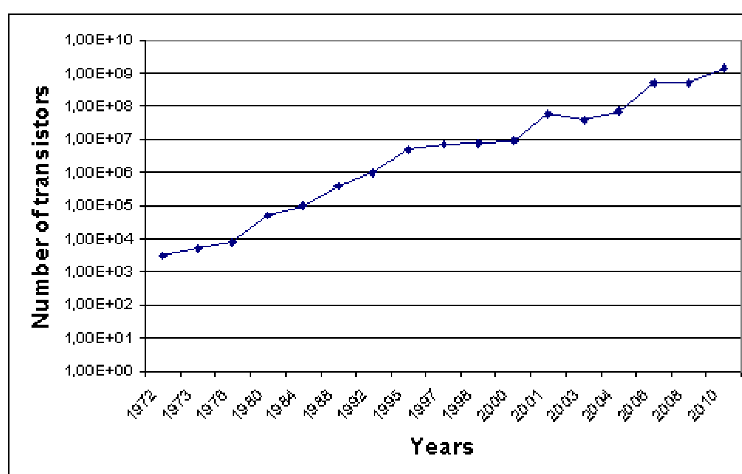


Figure 1.1: Evolution of microprocessor complexity

System on Chip (SoC)

A system-on-chip (SoC) refers to an electronic integrated system, which is usually realized in CMOS technology. Modern technological advances makes possible the integration of a complete electronic system on a single silicon chip [2]. This integrated circuit contains a large number of different electronic functions and is generally highly complex. Recent SoCs may include multiple microprocessor cores, digital signal processors (DSP), associated hardware operators, memory interfaces and inputs/outputs (figure 1.2). Complexities of such systems can reach over a billion transistors. Another feature of these systems on a chip is the very important part occupied by embedded software. There are two main types of SoCs: purely digital SoCs, and mixed SoCs, which co-integrate digital, analogue and sometimes RF functions. Embedded systems are a privileged application domain of SoCs.

Currently, advanced integration technologies are facing several problems such as the exponential increase of leakage and the not-so-exciting gain in performance.

CMOS technology seems to reach its physical limits, and further scaling of SoC becomes uncertain and very expensive.

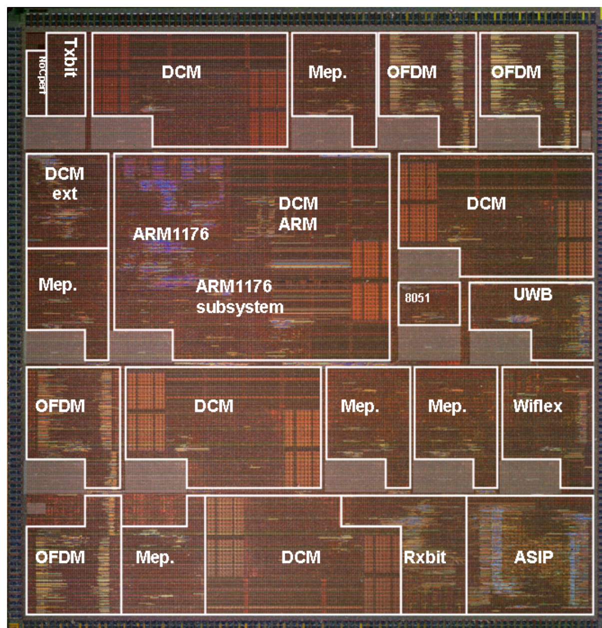


Figure 1.2: Example of a System-on-Chip

System in Package (SiP)

A system-in-package (SiP) refers to several integrated circuits (radio, analogue, digital) and/or discrete components (active or passive such as resistors and capacitors, or sensors and actuators) that are interconnected and enclosed in a single package [2]. This package involves the same standard packaging and/or assembly platforms used for simple integrated circuits (figure 1.3). The order of magnitude of these packages ranges from a few square millimetres to a few square centimetres. The dies containing integrated circuits may be stacked vertically on a substrate. They are internally connected by fine wires that are bonded to the package. The SiP performs most of the functions of a complete electronic system. It is primarily valuable in space constrained environments like MP3 players and mobile phones.

Despite its benefits, standard packaging technologies used for SiP production are still limited in terms of form factor and wire length. Another challenge with SiP is the decrease in manufacturing yield since any defective module in the package will result in a non-functional final circuit, even if all other chips in that same package are functional.

3D integration technology

3D die stacking is emerging as a promising solution that allows converging SiPs and SoCs (figure 1.4), by combining higher performance (More Moore) with complex and extended functionalities (More than Moore) in a single 3D circuit. 3D integration technology is to stack many circuits vertically and interconnecting them using Through-Silicon-Vias (TSVs). This results in smaller circuit footprint and

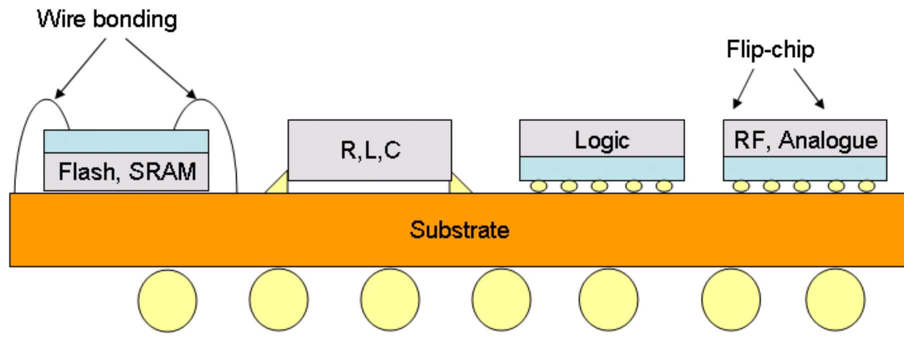


Figure 1.3: Example of a system-in-package

shorter vertical interconnections, which improves system performance and power. Besides, heterogeneous systems can be built easily, since each layer can support diverse technology. Figure 1.5 depicts an example of a 3D system. It is composed of

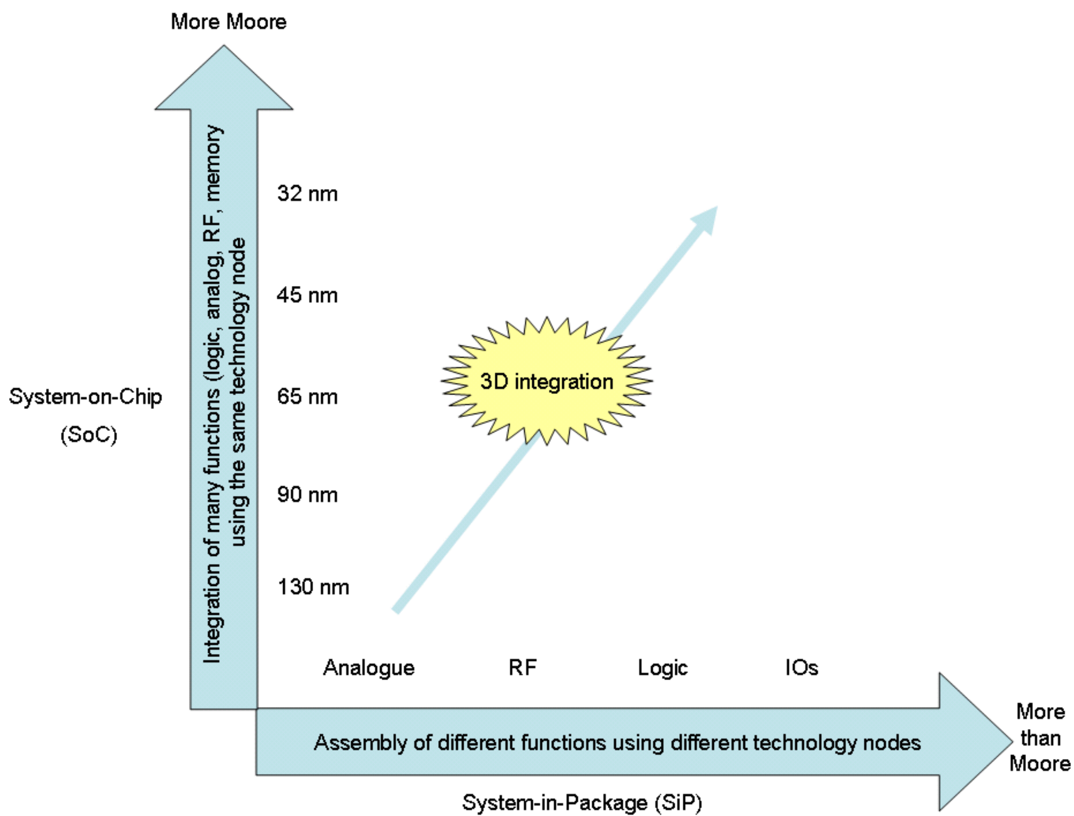


Figure 1.4: Convergence of the More Moore and the More than Moore trends

several dies and a 3D chip, stacked on top of an interposer. The silicon interposer is fabricated in a mature technology such as 130nm. The interposer may be active (including active components such as network-on-chip routers) or passive (containing only wires to ensure interconnection between the stacked dies). The stacked dies have different sizes and different functionalities. They are fabricated in aggressive technology such as 32nm. In the case of passive interposer, these dies are set side by side and interconnected by a very large number of connections in order to provide high inter-die interconnect bandwidth [3]. By using both TSVs and solder bumps,

it is possible to mount the interposer-based stack on a package substrate using classic flip-chip assembly techniques (Figure 1.5). The coarse-pitch TSVs provide the connections between the package and the interposer for the parallel and serial I/O, power/ground, clocking, data signals... Interposer-based stacking allows avoiding the reliability issues that can result from stacking multiple dies (fabricated on an aggressive technology) on top of each other.

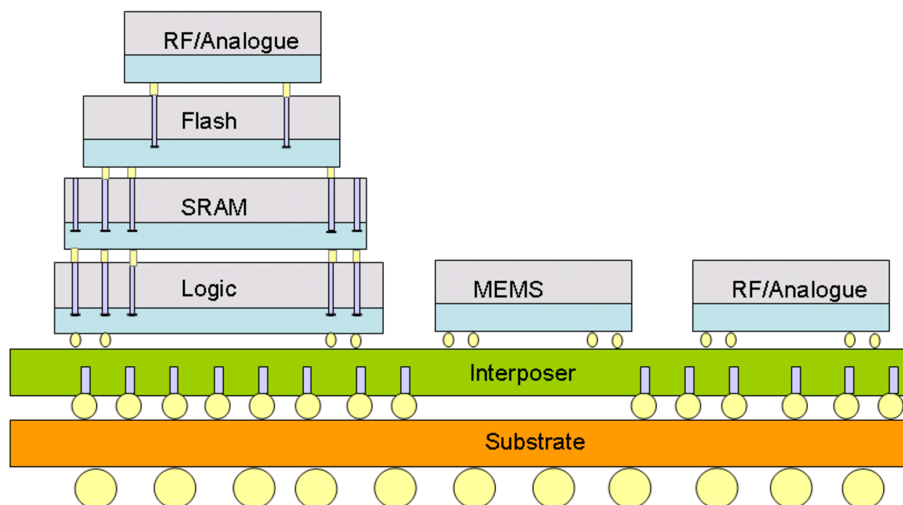


Figure 1.5: An example of a 3D system

2 Motivations for 3D die stacking

2.1 Reducing cost

Heterogeneous 3D integration

Present SoCs usually integrate heterogeneous functions (digital, memories, DSPs, analog and RF). These functions are initially designed for different manufacturing technologies. Although it is possible to fabricate all these devices on a single die using the same technology, this would be suboptimal in terms of performance, area, and power. Besides, this further complicates the fabrication process and increases manufacturing cost. Indeed, advanced digital technologies are not well adapted to realize functions such as analog or RF circuits. Past attempts to converge these different functions onto a single monolithic circuit resulted in many issues related mainly to cost and performance. For example, in a RF-CMOS process, the total price of a final wafer exceeds that of pure CMOS by more than 15% [4]. It is preferable then to make each function in its own mature technology node in order to get higher performance and lower cost.

A significant advantage of 3D integration is the possibility to integrate heterogeneous technology dies built with different processes on the same 3D circuit. This means manufacturing independently different functions such as analog, digital, or memory and integrating them in the same final system. It is then possible to manufacture each type of circuit using the most adapted technology [5, 6].

Same-die 3D stacking

Currently, a general VLSI application without regular system architecture requires multiple sets of masks. This can be extremely expensive since mask prices for cutting-edge processes have been increasing steadily (figure 1.6) [2]. According to the ITRS, the cost of only one mask set has already exceeded one million euro. For this reason, reusing mask and reducing the number of mask layers are becoming highly recommended.

In order to develop cost-competitive products, a potentiel solution is to reuse masks to address a wide range of systems. To do so, it is possible to design a modular circuit that could be stacked using 3D integration technologies to build 3D systems with processing performances adapted to several application requirements. Therefore, it would be possible to design several different systems using always the same mask set, thanks to 3D integration technology. Stacking many instances of the same circuit is referred to as same-die 3D stacking in this thesis.

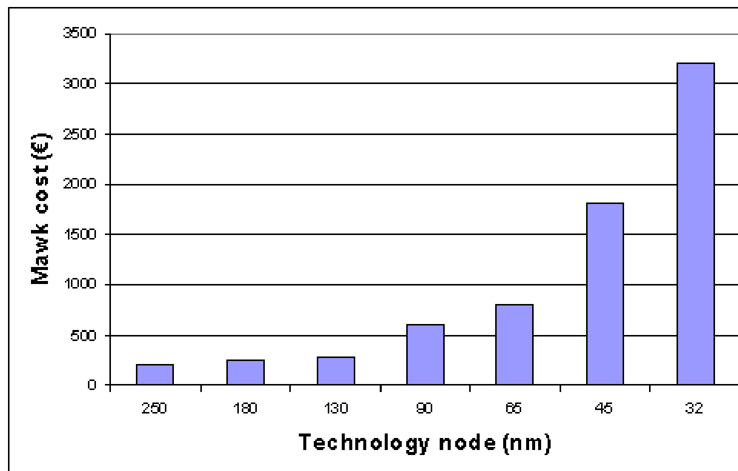


Figure 1.6: Increase in the cost of a set of masks according to technological nodes

Heterogeneous 3D integration and 3D same-die stacking approaches are not conflicting, but complementary. Indeed, it is possible to design a 3D SoC that includes several functions such as digital, memory, RF, analogue... Thanks to the 3D heterogeneous integration approach, each function may be fabricated using the most adapted technology in order to get better cost-performance tradeoffs. The 3D same-die stacking approach could be used for the digital part to boost the computational performance of the 3D system as needed by the targeted application (figure 1.7).

2.2 Enhancing performance and form factor

One of the most obvious advantages of 3D integration is to replace long horizontal wires with short vertical interconnects (TSV-Trough Silicon Via). Figure 1.8 illustrates the overall reduction of interconnections. The global inter-block wiring in 2D circuits (the longest wires) are replaced here by short vertical interconnections.

These shorter wires will decrease the average load capacitance and resistance (wire's capacitance and resistance are proportional to wire length) and reduce the

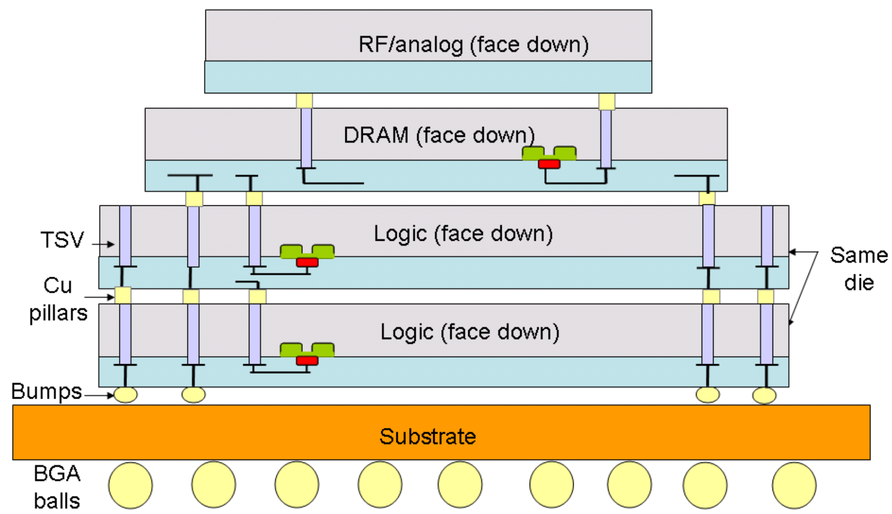


Figure 1.7: A 3D SoC designed with heterogeneous integration and same-die stacking approaches

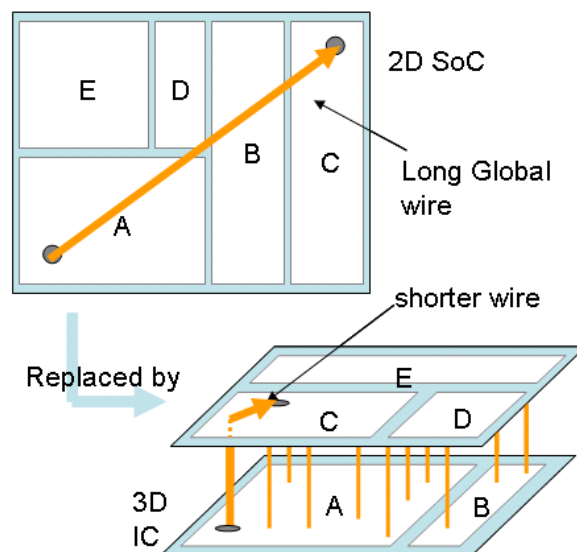


Figure 1.8: Interconnects' length shortening in 3D ICs

number of repeaters needed by long wires. Since interconnect wires with their supporting repeaters consume a significant portion of total active power, the average interconnect length reduction in 3D IC will significantly reduce overall power consumption.

Moreover, shorter interconnects in 3D ICs (with consequent reduction of load capacitance and reduced numbers of repeaters) will reduce the noise resulting from simultaneous switching events and coupling between signal lines. This should provide better signal integrity.

Another major consequence of the reduced wire resistance and capacitance is the significant reduction of signal propagation delay (proportional to the product resistance times capacitance), which results in significant system performance gain.

As shown in figure 1.8, 3D integration technologies allow reducing chip area and thus enhancing form factor. Therefore, it would be possible to continue chip miniaturization without necessarily following Moore's Law.

In conclusion, 3D stacking allows having shorter global interconnects within the 3D circuit, and thus reducing its total active power, coupling noise, and signal propagation delay . Besides, 3D integration technologies allow enhancing the circuit form factor.

3 3D circuit manufacturing technologies

A 3D integrated circuit may be fabricated according to several technological options [7]. A critical issue is to choose the way to assemble chips (figure 1.9):

- Die-to-die (D2D): This approach requires a stringent alignment effort since it seems difficult to handle small dies. Besides, chips' assembly is time consuming (Pick and Place).

- Wafer-to-wafer (W2W): In this case, the time necessary for chips' assembly is much shorter. Further, alignment is easier since assembly is performed on bigger objects. All dies of on these stacked wafers must have the same size to be separated after assembly.

- Die-to-wafer (D2W): The time needed for stacking is less significant than in the D2D option. Alignment issue is also less critical than in the die-to-die approach. Another advantage of this technique is that the stacked chips can be different sizes: it is possible to stack a small-sized circuit on top a larger circuit. Although the D2W approach is more expensive the W2W approach, it could be used when circuit fabrication and stacking are not performed by the same foundry.

Another important topic is to decide on how active levels are oriented (figure 1.10):

- Face-to-face (F2F): stacked circuits are assembled so that the metal layers are face to face. This option gives a high integration density, since there is no TSV (Through Silicon Via) in the active area. However, it is not possible to stack more than two layers this way.

Figure 1.9: Three ways to assemble the circuits vertically

- Face-to-back (F2B): In this case, integration density is lower, but stacking more than two layers is possible.

Figure 1.10: F2F and F2B approaches in 3D stacking

To achieve 3D stacking, some key technological steps must be fully involved: wafer thinning, bonding and TSV forming.

3.1 Wafer thinning

The wafer can be either a bulk Si or SOI (Silicon on Insulator). Although more expensive, the buried oxide layer (BOX) in SOI wafers provides a selective etch stop for the uniform removal of the Si substrate. This step is more critical on bulk Si wafer. The chips have to be thinned in order to have a good form factor and to reduce vertical interconnections' length. The substrate thinning is mandatory to fabricate high density TSV with an aspect ratio from 1 to 10 [2]; it is performed in two steps:

- The grinding step is to remove a large amount of material at high speed using an abrasive wheel.

- The polishing step allows getting rid of roughness and obtaining a good quality uniform surface. This is indispensable to enable the subsequent technological process. This step is often carried out by combining chemical mechanical polishing (CMP) and wet chemical surface treatment.

Current technologies allow reaching up to 10μm for wafer thickness.

3.2 TSV forming

The development of TSVs may be performed at different stages of the manufacturing process:

- Via first: before the front-end (transistors),
- Via middle: after the front-end (transistors) and before the back-end (interconnection metal layers),
- Via last: after the back-end.

In all these cases, TSVs are directly accessible by low metal layers in the same manner as transistors. TSVs may be filled with copper (most often used), tungsten or even poly-silicon. Depending on applications, TSVs' diameter may vary from 1μm to 100μm and their form factor (height to width ratio) from 1 to 30 (figure 4.12). Smaller TSV sizes give the highest integration density. Therefore, reducing vias' size is a critical challenge for both manufacturer and designer in order to reduce 3D circuit cost.

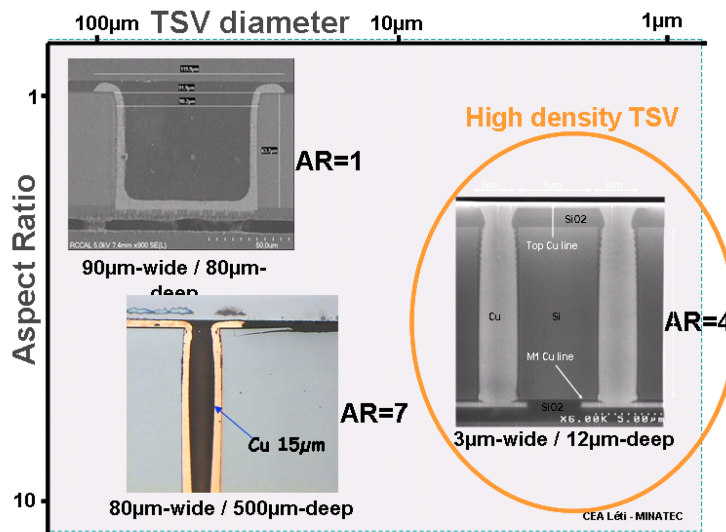


Figure 1.11: Different TSV sizes and form factors

3.3 Bonding

A variety of bonding techniques are being considered. Nevertheless, there is no rework solution so far. For all types of bonding methods, the quality of the bonded interface strongly depends on surface roughness and cleanliness. Bonding may be achieved in two different ways: metallic bonding or molecular (direct) bonding.

For example, a bonding technique is to use copper pillars to interconnect TSVs of the upper die and metal layers of the lower die. They are formed by developing copper layers on both sides of the two circuits to be stacked. Copper pillars of the 2 circuits are bonded thanks to a solder paste (mixture of tin, silver and copper).

After that, the 2 circuits are aligned. The alignment step may be performed using either optical or infrared microscopy. Depending on the chosen integration scheme (F2F or F2B), the alignment is performed according to the substrate face or the substrate back. Besides, the higher the integration density will be the greater alignment accuracy will be required. The alignment precision depends not only on equipment but also on substrate deformation generated by the technological steps preceding alignment. If the bonding interface is not smoothed sufficiently, alignment can be degraded by the following process (figure 1.12). Currently, obtained alignment precision may reach 1μm.

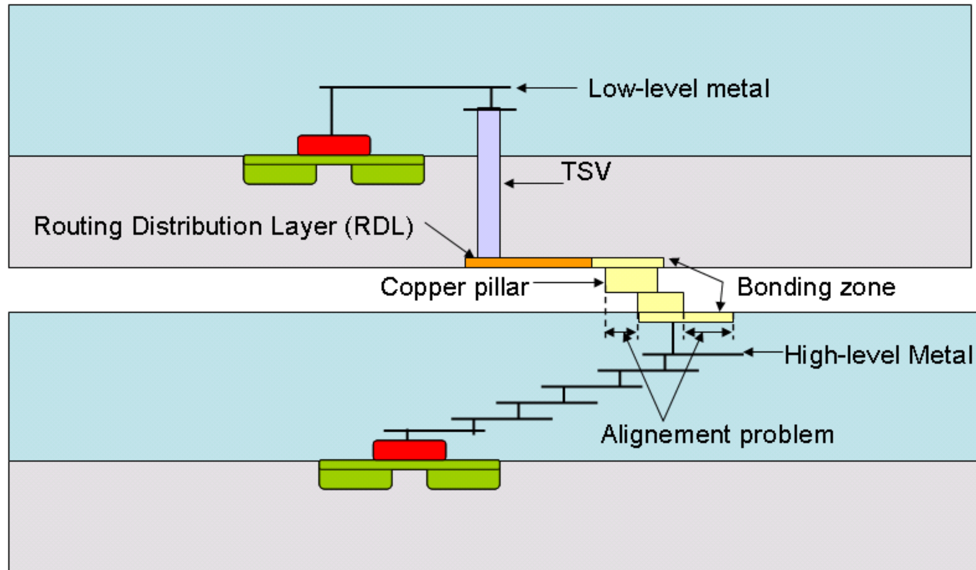


Figure 1.12: Die bonding using the copper pillar technique

3.4 3D IC manufacturing yield

For a new technology to be economically viable, yield is a critical issue for the chip designer and producer.

In general, the larger chip size provides the lower yield. Hypothetically, we can suppose that a wafer has a known number of fatal defects that are spread randomly over the wafer surface. Thus, the average number of defects per chip would be $A \times D_0$, where A is the chip area and D_0 is the number of defects on the wafer divided by the surface of the wafer. The yield can be defined as [8]:

$$Y = (1 + \frac{A \times D_0}{\alpha})^{-\alpha}$$

where:

- A is the 2D IC area,
- D_0 is the density of point-defects per unit area,
- α is a model parameter, and typically ranges from 1.0 to 5.0.

Figure 1.13 shows the yield variation (for $\alpha = 1$) according to chip area for different technology nodes: 130nm, 65nm and 32 nm (with D_0= 0.02, 0.002, and 0.0002 respectively). When the chip size increases, yield falls with rates that depend on the technology maturity. As seen is section 2, 3D integration allows reducing total chip area and thus enables significant enhancement in individual chip yield.

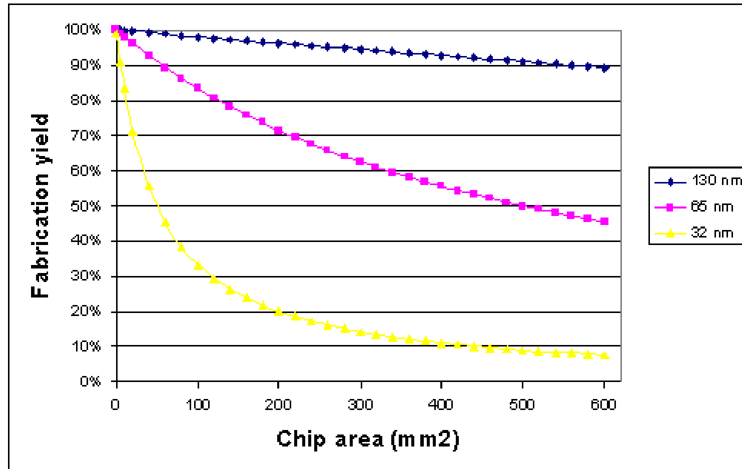


Figure 1.13: Yield according to area

In reality, total manufacturing yield in 3D stacking technology depends not only on the yield of individual die production but also on the chosen assembly scheme: die-to-die, wafer-to-wafer or die-to-wafer. Indeed, when stacking n dies made with a technology yield Y_d, the final yield Y_f of the 3D system enclosing these n stacked chips becomes:

$$Y_f = Y_s \times Y_d^n$$

with Y_s being the 3D integration (interconnections and assemblies) technology yield. For example, when stacking 4 wafers having a yield of 80%, using an assembly technology with a yield of 90%, the final yield drops to 37% with the W2W approach. In the case of assembly schemes D2D or D2W, it is possible to test the chip in advance and select only the functional chips (known good die); the final yield reaches 72%.

4 Challenges in 3D integration technologies

In spite of all its benefits relevant to cost and performance, 3D stacking is currently facing some major challenges that hinder its industrial application. This section focuses on the most important challenges and presents some potential solutions.

4.1 Thermal management

One of the most critical challenges of 3D design is heat dissipation. The thermal issue in the vertical stacking is caused by two main reasons. Firstly, in 3D circuits more devices are packed into a smaller volume, resulting in a rapid increase of power density, especially if the stacked tiers are highly active (typically logic circuits). In addition, heat from the top-most core of a 3D chip has to travel through several

layers, which are inserted between device layers for insulation, to reach a heat sink. The thermal conductivity of these dielectric layers is very low compared to metal layers or even silicon.

High temperature has two main drawbacks: it can limit the operating frequencies of vertically-stacked chip and degrades chip reliability. These two factors have made thermal management to be identified as a critical challenge in 3D devices. Many academic and industrial researches propose several promising methods to solve thermal challenges [9].

Thermal management without thermal vias

Several thermal-driven floorplanning, placement and routing algorithms have been proposed to deal with thermal problems in 3D ICs. These algorithms make use of 3D IC thermal models in order to take into consideration high temperatures [10]. Thermal-driven floorplanning, placement and routing algorithms target the optimization of an objective function that includes total circuit area, total length of inter-block wires, total number of vertical vias (to decrease fabrication cost and silicon area), and finally a 3D IC thermal model. The objective function of a 3D IC may be resolved using several well-known techniques such as simulated annealing and force directed methods.

Thermal management with thermal vias

An efficient technological solution for thermal issues is to insert thermal vias, which are used to create thermal paths helping heat dissipation from a core on a stacked chip to the heat sink (figure 1.14) [11]. Several ways are proposed in the literature to place thermal vias. These techniques include inserting thermal vias within certain regions, or dispersing thermal vias across the stacked layers (known as thermal vias planning). It is also possible to connect regions with different thermal vias densities using special routing channels called thermal wires.

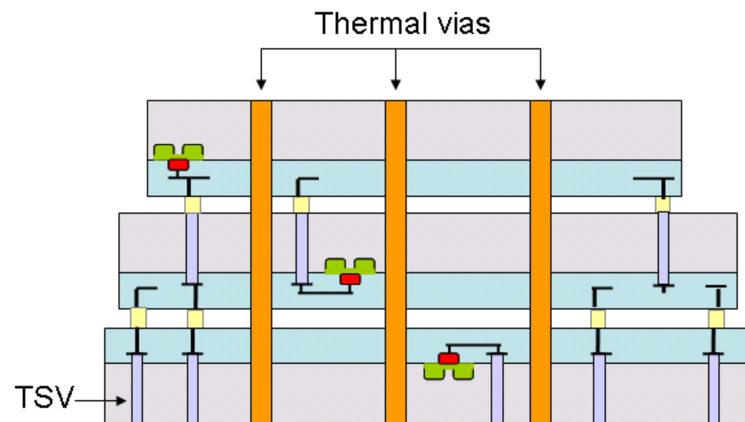


Figure 1.14: Yield according to area

4.2 TSV area overhead

With current 3D integration technologies, TSV diameter may range from 1μm to 100μm. This size is considered huge compared to the size of transistors. For example, the area of a 4μm diameter TSV is superior to the area of 500 SRAM cells in 45nm technology (figure 1.15). Therefore, it is important to minimize total TSV area by reducing their size (manufacturer role) and their number (designer role) in order to reduce 3D circuit cost.

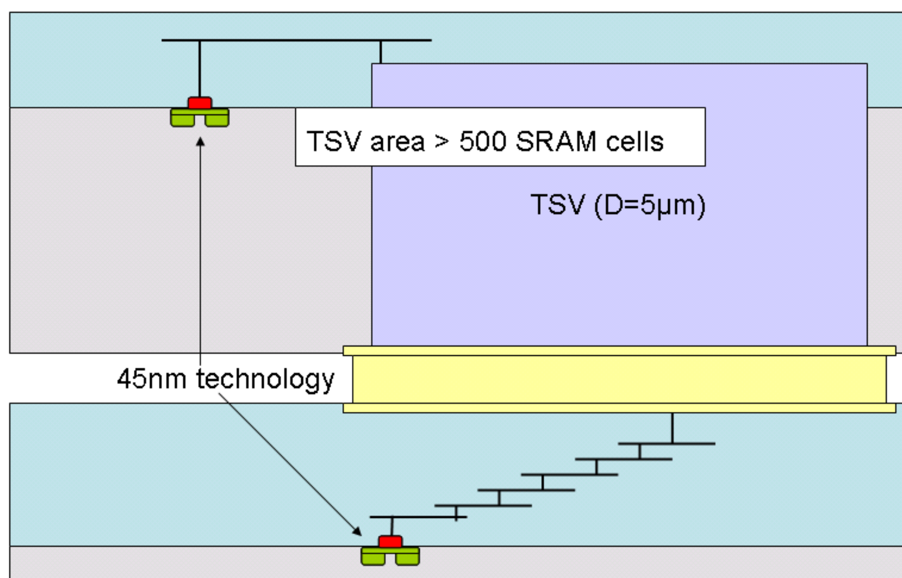


Figure 1.15: TSV area vs transistor area

4.3 CAD tools

Present CAD tools used for IC design have to be adapted and several features must be added, in order to support the new challenges of 3D systems such as circuit partitioning, thermal management techniques, global clock distribution in digital synchronous systems... To do so, a lot of efforts is needed to be spent on both algorithms and visualization options of classic CAD tools [12]. Algorithms should be extended by appending new behavioural models to take into account the new specifications of 3D ICs. Visualization options should consider the third dimension, so that the designer could understand and manage more effortlessly the characteristics of such 3D systems. Finally, these extensions in both algorithms and graphics imply that more computational power is required, to model and assess efficiently the whole 3D system.

4.4 Test challenges

A major obstacle for 3D integration technologies is the not enough comprehension of 3D testing problems. Indeed, design-for-testability techniques for 3D ICs are still insufficiently explored by the research community. Moreover, several major test challenges are identified by industrial experts and include the lack of probe access for wafers, the test access to modules in stacked wafers/dies, the thermal concerns, the

design testability, the test economics, and the new defects arising from 3D-specific processing steps such as wafer thinning, alignment, and bonding [13].

5 Potential applications

Both research and industrial communities are working on 3D integration technologies. Several demos have been made to prove the validity of the 3D fabrication process. Figure 1.16 depicts a roadmap of 3D applications depending on advances in TSV size reduction.

The first achievements looked at homogeneous technologies, especially memory. Since 2006, Samsung Electronics has been the first to announce the stacking of 8 wafer-level processed 2Gb NAND flash memory, for a total height of 0.56mm using TSV interconnection technology. This miniaturization of memory presents several potential benefits especially for mobile applications.

In 2006, Intel published the wafer-level stacking of SRAM memory operating at 4Gb. The bonding was done using 330mm Si bulk wafers processed in 65nm technology and face to face assembly. The top wafers were thinned to different thicknesses ranging from 5 to 28μm [14].

As shown in figure 1.16, first 3D heterogeneous technologies integration began in 2006 with Tohoku University proposing a novel retinal prosthesis system consisting of several LSI (Large-Scale Integration) chips vertically stacked and electrically connected using 3D integration technology. The retinal prosthesis chip is made including photo-detectors [15].

In 2007, MIT and Yale University presented the design and measurement results of 3D photo-detectors made using 0.18μm SOI CMOS technology [16].

In 2009, IMEC, with many players in the 3D integration supply chain, announced a new 3D demonstrator integrating a commercial DRAM chip on top of a logic IC [17]. The thickness of the logic die was a 25μm. The chip contained test structures for monitoring thermo-mechanical stress in a 3D stack, ESD (electro-static discharge) hazards, electrical characteristics of TSVs and micro-bumps, fault models for TSVs. Heaters were integrated to test the impact of hotspot on DRAM refresh times. Unfortunately, no information was given later about the success of the chip given later about the success of the chip.

From the year 2009 on, advances in 3D integration technologies would allow drastic reduction of TSV size, enabling several other applications. It would be possible to make a complete 3D heterogeneous system consisting of IO, digital, memory, RF, MEMS, sensors... This depends especially on advances in terms of TSV size diminution and thermal management.

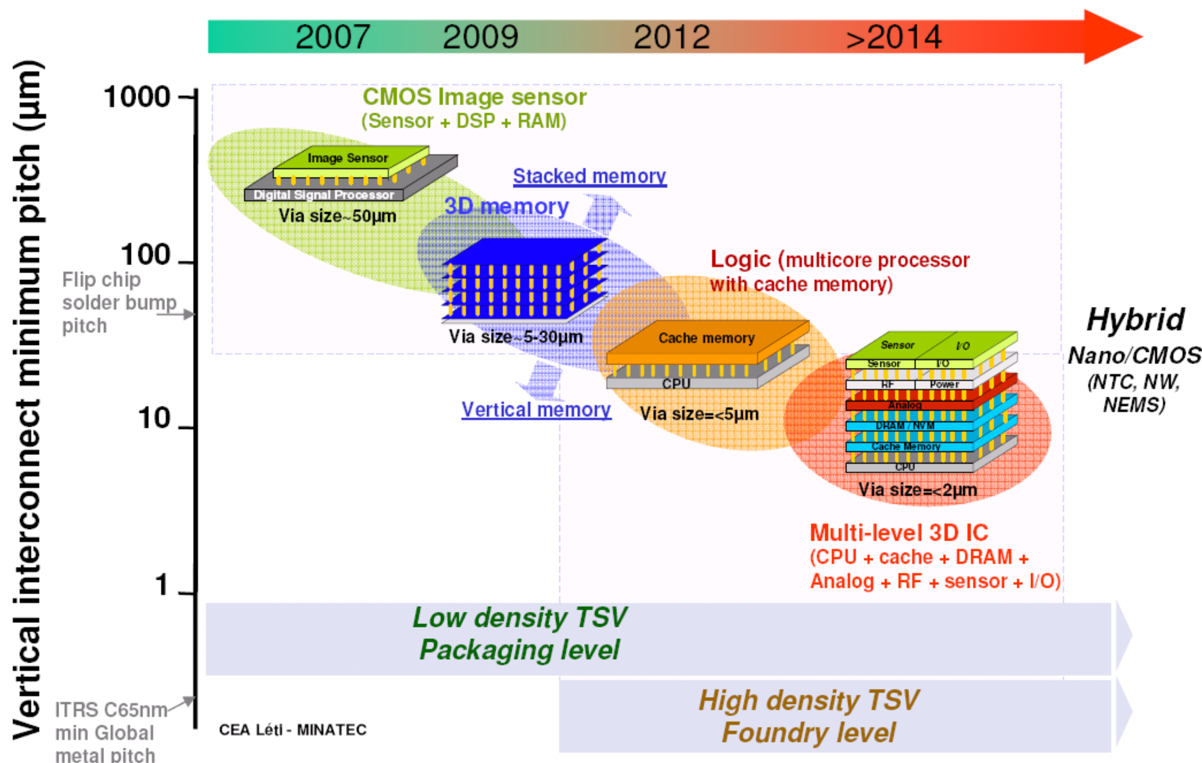


Figure 1.16: 3D applications roadmap

6 Conclusion

In this chapter, some main aspects of 3D integration technologies were presented and explained.

3D stacking technology is rising as a promising solution allowing to continue chip miniaturization and to integrate extended functionalities. 3D integration technology is to stack many circuits vertically and connecting those using TSVs. This results in smaller circuit footprint and shorter vertical interconnections, which improves overall system performance and power. Thanks to this 3D die stacking, it would be also possible to manufacture each system function with its most adapted technology. Finally, the same mask set can be reused to build many digital systems adapted to different applications.

When fabricating a 3D circuit, it is necessary to choose the way to assemble chips (die-to-die, wafer-to-wafer or die-to-wafer) and to decide on how active levels are oriented (face-to-face, face-to-back). The fabrication includes 4 major steps: wafer thinning, bonding, alignment and TSVs' formation. The manufacturing yield of a 3D circuit depends on the die yield and the bonding yield.

In spite of all its benefits relevant to cost and performance, 3D stacking is facing some major challenges. The first one is the thermal management challenge caused by the increased power density. Current solutions for this problem consist of thermal-driven floorplanning, placement and routing techniques and inserting thermal vias within the 3D stack. Other major challenges of 3D integration are TSV area overhead, the insufficiency of CAD tools, the not enough comprehension of 3D testing problems and the lack of design-for-testability (DFT) solutions.

Next chapter focuses on challenges that arises when moving from a 2D multiprocessor architecture to its 3D version. It deals mainly with the partitioning granularities and the related cost challenges.

Chapter 2

From 2D to 3D architectures: partitioning and cost challenges

The aim of the previous chapter is to introduce the 3D die stacking in terms of advantages, technologies and major limitations. This background is necessary to understand the challenges that arise when moving from classical 2D to new 3D multiprocessor architectures. When designing a new 3D digital circuit, it is obvious to focus on how to partition the original 2D circuit. The partitioning approach is intended to take full advantage of the 3D integration potential. Moreover, it has to take into account the economical limitations described in the first chapter. Therefore, cost analysis on early design stages is very important to decide on the best 3D technology options to choose.

This chapter is intended to deal with two main challenges of 3D die stacking in the case of multiprocessor architectures: partitioning granularity and cost-effectiveness. It is organized in 2 sections. The first one deals with partitioning aspects of 3D digital ICs. It introduces 3 levels of partitioning: core-level (or macro-bloc level), functional unit level and basic operator level. Based on several relevant state-of-the-art works, the pros and cons of each partitioning granularity are explained in details and illustrated by some examples of 3D digital architectures. The second section focuses on cost challenges. After presenting some pertinent related works regarding cost analysis for 3D ICs, a system-level cost estimation model is presented in details. This cost model takes into account costs of mask set design, wafer fabrication, 3D fabrication process steps, and circuit test. Based on this model, we explore the cost trends of a 3D digital circuit according to the variation of its area, the number of its layers. This help us decide on the best options to choose in order to improve the cost-effectiveness of 3D digital circuits.

1 Partitioning challenges

In this section, based on state-of-the-art works, we present in details the different partitioning granularities of 3D digital circuits, and we explain the advantages and limitations of each partitioning approach.

The partitioning of a 2D digital circuit across two or more layers may be performed according to 3 different granularities (figure 2.1) [18]. The first one is at the core level: it is to stack macroscopic blocks such as memory and processor. Examples include staking cache or main memory on top of a CPU, or stacking a CPU on top of another CPU. It is also possible to partition a circuit according to functional units such as splitting the processor itself across two or more layers. This means stacking the arithmetic and logical unit directly above the register file for example. The finest partitioning granularity is to stack basic operators such as multiplexers and logic gates. This enables splitting functional units like the instruction scheduler or the register file across several layers.

Each one of these partitioning types has its own advantages and limitations. Choosing one of these partitioning possibilities depends on the performance gain we are expecting and the cost we are willing to pay. Here after some relevant examples of each partitioning possibility, helping us to better illustrate their pros and cons.

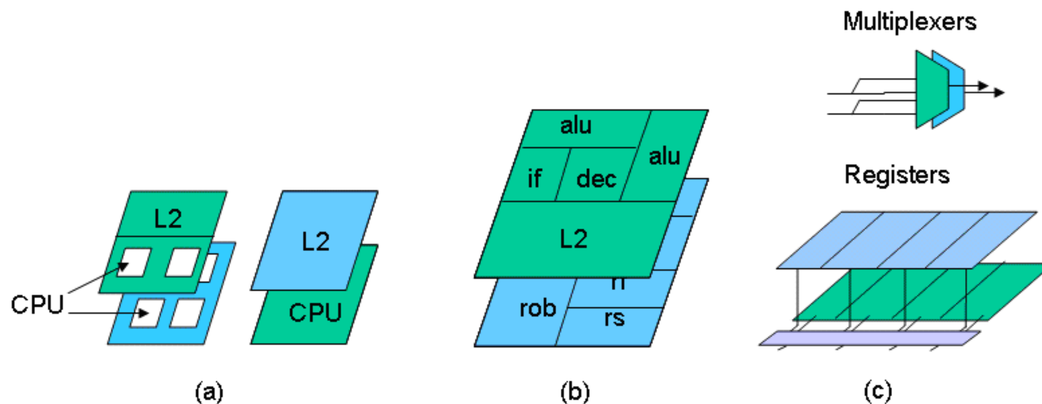


Figure 2.1: Partitioning granularities: (a) macroscopic blocks (b) functional units (c) basic operators

1.1 Macro-block stacking

Several studies are carried out to explore this type of partitioning in order to reveal its contributions in terms of performance, power and thermals, and to identify the most promising architectures. We present the proposed architectures and the obtained results for some relevant researches. It is worth noting that these studies are performed based on simulations, not on real physical implementations.

Stacking L2 cache on top of a processor

A study performed by a team of researchers from Intel Corporation [19] compares three schemes to improve the performance of a planar circuit composed of a Core 2 Duo processor, with an L1 cache for each core and a shared L2 SRAM cache (figure

2.2). The first 3D scheme is to retain the original planar circuit and expanding the L2 cache by stacking more SRAM on top. The second option proposes to replace the L2 SRAM cache by a denser (therefore larger) L2 DRAM cache. In this case, cache memory is stacked on top of the circuit which is composed only of the CPU and L1 cache. The last design id to stack DRAM memory (as L2 cache) above the initial circuit.

Performances of the proposed circuits are measured in terms of cycles per memory access CPMA and off-die bandwidth BW. According to simulations, the three proposed schemes outperform the original planer circuit. In particular, the second 3D system (L2 DRAM cache on top of the CPU) can reduce CPMA by 13% and off-die BW by 66% on average compared to the original 2D circuit. Simulations also register an average 66% power reduction in average bus power. In the same time, none of the stacking schemes considerably impacts thermals. Indeed, temperature increase is not prohibitive and varies between 0.08°.C (stacking DRAM on top of the CPU) and 4.5°.C (stacking more SRAM above the initial circuit). This can be explained by the higher power density of SRAM compared to DRAM.

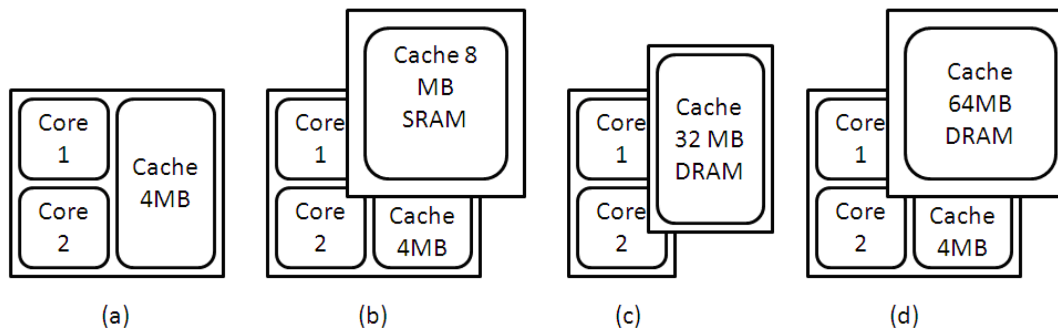


Figure 2.2: Memory stacking schemes: (a) original 2D circuit, (b) stacking more SRAM above the initial circuit, (c) stacking L2 DRAM cache on top of the CPU, (d) stacking more DRAM on top of the initial 2D circuit

Stacking main memory and L2 cache on top of processor

A group of researchers from the University of California [20] present a 3D system composed of a processor, a cache and a main memory, all integrated on the same chip (figure 2.3). The CPU and L1 cache are implemented on the first layer in 130nm technology, while the L2 cache is on the second layer. SDRAM main memory, implemented in 150nm technology, is partitioned across 16 layers above the first two layers.

According to simulations, the considerable increase in memory bus frequency and bus width contribute to a significant decrease in execution time for the 3-D system. Indeed, for a typical 1 MB L2 cache configuration, execution time improvement over 2-D system is found to reach 57.7% with a 16-byte bus width. However, temperature increase in this case is relatively higher than that of the previously presented system; it reaches 12°C for a 300 MHz frequency. Consequently, thermal constraint in this 3D design imposes a maximum allowed operating frequency lower than that of 2-D designs. In spite of this, the overall system performance is still significantly better than conventional planar designs, especially for memory intensive applications.

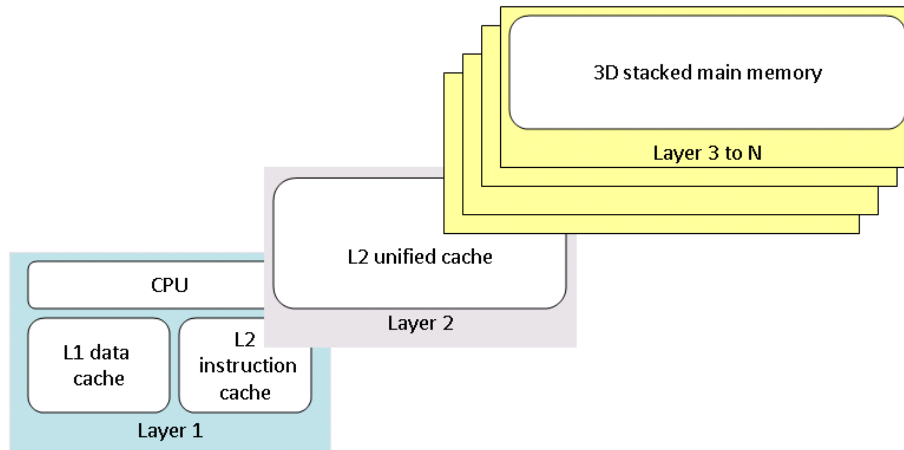


Figure 2.3: The 3D stacked processor-cache-memory system

Stacking main memory on top of a processor without L2 cache

Another architecture called PicoServer is proposed by researchers from Michigan University and ARM Company [21]. As illustrated by figure 2.4, the proposed circuit is composed of a first layer containing several slow processor cores, and several other layers forming the DRAM main memory. This architecture has no L2 cache; the main memory is connected directly through a shared bus to the L1 cache of each core. This approach is justified by the fact that latency and bandwidth obtained using stacked DRAM are comparable to those of the L2 cache. Therefore, it is more advantageous to remove the L2 cache and replace it with additional processor cores. The additional cores allow the clock frequency to be lowered without affecting performance. Lower clock frequency in turn reduces power and thus allows decreasing thermal constraints which are a concern with 3D stacking.

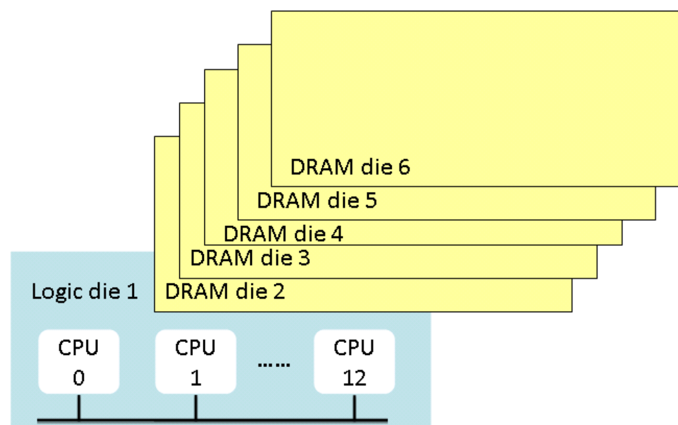


Figure 2.4: PicoServer: a multi-processor architecture connected to a conventional DRAM using 3D integration technology

Simulations confirm that for a similar logic die area, a 12 CPU system with 3D stacking and no L2 cache outperforms an 8 CPU system with a large on-chip L2 cache by about 14% while consuming 55% less power. Besides, PicoServer shows the same performance as a Pentium 4-like class device while consuming only about 1/10 of the power. According to simulations, temperature increase is not a major

limitation in the PicoServer platform since the power density is relatively low (It does not exceed $5W/cm^2$).

In conclusion, in spite of the diversity of the previously exposed architectures, they all confirm the role that 3D integration can play to improve system performance (in terms of cycles per memory access, bandwidth and execution time), compared to classic planar architectures. Stacking memory on top of logic leads to significant reduction in the overall circuit power by decreasing global interconnects' length. Given that memory is not a highly active layer, reached temperatures in this type of architecture are imperceptibly higher than those of planar circuits.

Although all the previously presented examples deal with stacking memory on top of processor, core-level partitioned 3D architectures may include also staking processors on top of processors: we are talking here about 3D multiprocessor circuits. It is worth noting that 3D multiprocessor circuits are complex and require efficient interconnection architectures to ensure communication between the logic cores located on different tiers. Interconnection architectures proposed in the literature include mainly bus and network-on-chip.

1.2 Functional unit stacking

In order to get more advantages from 3D die stacking, it is possible to partition the processor core itself across many layers according to a finer granularity, called functional unit level. This alternative means stacking several functional unit blocks such as stacking register file directly above the arithmetic and logic (ALU) units. This helps decreasing the length of the wires used for bringing operands to the ALU and those used for writing results back to the register file. Since these data paths are usually large (many operands, each at 32, 64, or even 128 bits) with high activity factors, the 3D organization would result in both performance and power gains. Besides, with this partitioning approach, each functional unit is still basically a planar component, and then some design reuse is still possible.

The most relevant study concerning this partitioning type was carried out by B. Black et al. from Intel Corporation [22]. They focus on implementing the iA32 microprocessor (a real deeply pipelined high performance x86 processor) in a 2-layer stack using face-to-face bonding as it provides a higher density die-to-die inter-connect compared to face-to-back option. Using the 3D design approach, the obtained floor-plan requires only 50% of the original footprint.

As expected, the implementation of the microprocessor in 3D provides a substantial improvement in performance and power compared to conventional planar processor. According to [22], performance and power improvement on a 3D implementation of a X86 type processor reaches 15%.

Concerning thermal challenges, several measures have been performed by means of accurate models of the 3D structure for thermal dissipation. It is observed that a naïve floor-plan can raise heat by 10-15%. The hot areas of the die are due to several blocks that make use of aggressive dynamic circuits to make timing. By splitting these hot blocks across two layers, it is possible to decrease internal wire delay sufficiently. This allows reducing power consumption by as much as 50% while maintaining latency and power density. If this technique is applicable for all hot blocks, it would be possible to avoid 3D thermal problems.

In conclusion, unlike the first type of partitioning that does not affect the processor itself, functional unit stacking can significantly improve performance and decrease power of the 3D CPU. However, at the same time it may cause a significant increase in temperature, so as to weaken the reliability of the processor. It is thus necessary to deeply investigate thermal problems associated with this type of partitioning to find appropriate solutions to ensure the intended functionality of the 3D processor.

1.3 Elementary operators stacking

The finest granularity of 3D partitioning is to split functional unit blocks across several layers. This means stacking elementary operators such as logic gates, multiplexers, registers... For example, it is possible to split cache's word-lines or bit-lines, or to stack the entries of a processor's reservation stations.

The main advantage of this partitioning type to remove significantly intra-block wiring, which can afford important power and performance gains for wire-dominated blocks such as the instruction scheduler, and large multi-ported SRAMs (a physical register file or register alias table). Moreover, this approach would decrease each block footprint, which leads to a more compact overall floorplan and shorter global wires. Therefore, this partitioning granularity has the potential for larger performance and power advantages with respect to the coarser-grained partitioning techniques. However, a considerable design effort is required to implement 3D versions of all the functional unit blocks.

Most of the work presented in the literature are concerned with the implementation of various processor components in 3D such as the instruction scheduler, the file register, the arithmetic units [23–25]... In the remainder of this section, we take as examples the instruction scheduler and the file register.

Instruction scheduler

In current superscalar processors, the dynamic instruction scheduler is in charge of exposing instruction level parallelism by identifying instructions that can be executed in parallel. G.H Loh and K. Puttaswamy from Georgia Institute of Technology has proposed two schemes to implement the 3D instruction scheduler: according to source operand addresses and according to entries [23]. According to their study, a 2-layer implementation of an instruction scheduler with 20 entries (60 entries respectively) improves performance by 9% (15% respectively). Improvements reach 12% (24% respectively) if the 3D scheduler is implemented in 4 layers. Concerning power consumption, the 2-layer implementation of a scheduler with 20 entries (60 entries respectively) induced a decrease of 18% (45%, respectively) compared to its planar version. The 4-layer implementation results in a reduction of 25% (67%, respectively).

According to G.H Loh and K. Puttaswamy, the 3D thermal analysis of the scheduler (or any other individual component of the processor) is not practical since the worst-case temperature depends not only on the 3D layout of the scheduler, but also on powers of neighboring components. Thus, they evoke no quantitative estimations of the temperatures reached by the 3D scheduler. Another work carried out by B. Vaidyanathan et al. from the University of Pennsylvania [24], implemented in 3D

the instruction scheduler (using the same approaches) and quantified the reached temperatures. According to their study, a 4-layer implementation causes a 10°C increase in average temperatures compared to the planar version.

File register

The register file is the component that contains all the general purpose registers of the microprocessor. A register file is a set of registers, a decoder, addresses and data inputs. K. Puttaswamy and G. H. Loh [25] propose three different strategies for partitioning the register file across multiple dies: according to registers, to bits or to ports. In term of access latency reduction, best results of 16.8% (respectively 24.1%) are obtained when using the bit-partitioning approach (respectively register-partitioning approach) for a 2-die 128-entry register file (respectively 2-die 256-entry register file approach. In term of power consumption reduction, best results of 21.5% (respectively 58.5%) are obtained when using the bit-partitioning approach for a 2-die128-entry register file (respectively 2-die 256-entry register file approach).

In conclusion, this type of partitioning requires the complete re-design of the functional units. It allows reducing the interconnections lengths inside functional units, and thus improve their performances. Moreover, power consumption is decreased proportionally to the preponderance of wires within the functional unit. However, area overhead due to TSVs is important compared to the size of the stacked basic operators. For example, as mentioned in chapter1, the area of a 4μm diameter TSV is superior to the area of 500 SRAM cells in 45nm technology. Finally, power density increases considerably compared to other partitioning granularities. Thus, thermal constraints are more important.

1.4 Synthesis

The previously presented researches regarding 3D digital architectures help have some important findings about their viabilities and their major trends. It turns out that all of the 3 types of partitioning allow improving the performance and power with rates that increase when the partitioning granularity decreases. In the same time, challenges such as redesign efforts, thermal stresses, and area overhead due to TSVs become more and more noticeable when miniaturizing the size of the stacked elements. Taking into account these limitations, it can be concluded that macro-block stacking seems to be the most economically viable approach, as it requires the least design efforts, depicts the lowest temperature increase and is the most adapted for current TSV size.

For this reason, this thesis is oriented to focus on macro-block stacking. At this level of partitioning, the involvement of 3D integration technology is limited to the vertical inter-block interconnections. As said previously, macro-block-level partitioned 3D multiprocessor architectures require efficient interconnection architectures to ensure communication between the logic macro-blocks located on different layers. A first contribution of this thesis is to propose a new 3D NoC topology that allows performance enhancement. Detailed presentation of this work is provided in chapter 3.

Although macro-block stacking appears as the most viable partitioning approach, its performance and power gains are not so-exciting. Therefore, the use of 3D in-

tegration (at this level of partitioning) could not be well-justified based on performance benefits only, but others motivations, especially economical ones, should be mentioned. Next section presents a simple cost analysis model for 3D integration technology that helps us analyse the cost-effectiveness of 3D architectures.

2 Cost challenges

3D integration is made in several steps, each of which includes a wide range of technological choices. Choosing the optimal process flow depends mainly on cost. According to [26], decisions taken during the first 20% of total design cycle time influence by 80% on the final product cost. Therefore, cost estimation of 3D ICs in the early stages of the design cycle is so important. This section deals with 3D ICs cost challenges. After presenting some relevant researches about cost analysis, we propose a system-level cost model, which allows analysing the cost-effectiveness of 3D ICs.

2.1 3D IC cost analysis: state of the art

Several works have focused on 3D integration cost analysis. Their approaches range from system level to technology detailed assessment.

R. Mercier et al. focus on 3D integration yield modelling, based on technology-dependent parameters. This yield model is used to perform a cost estimation of 3D ICs [27].

R. Weerasekera et al. propose a whole cost modelling flow that makes use of parameterized analytical models, to assess area, yield, cost, thermals and interconnect performance [26]. Using on these models, they carry out a comparison between SoC, SiP and 3D implementations in terms of cost and performance.

X. Dong and Y. Xie present a system-level analysis for 3D ICs [8]. Based on Rent's rule, they perform an estimation of the number of wires inside a 2D chip, and deduce the number of TSVs within the resulting 3D stack after partitioning. Then, they propose parameterized cost models, which take into account several aspects of 3D ICs such as bonding yield, known-good-die test, assembly options... Based on these models, some important trends about 3D ICs cost are found out. This helps the authors of [8] to propose a cost driven design flow for 3D ICs.

P. Marchal et al. introduce a methodology to assess 3D integration technology called path-finding the technology/design sweet-spot [28]. The basic idea is to perform a system-level co-optimization of both design and technology options in order to take full advantage of the 3D integration technology potential. To do so, the proposed methodology takes as inputs a block-level description of the system (with performance and communication requirements) and predictive design rules for technology, and outputs a 3D model including power, performance, and cost estimation. The cost of a 3-D stack is composed of the silicon wafer cost and the cost for realizing the TSVs. The costs of performing the 3-D process (TSV and bonding) are estimated based on a detailed analysis of the process flow and the required process time and cost per wafer.

Based on the previously described works, we propose a system-level cost analysis model that allows having a preliminary cost estimation of a 3D system, and deciding

on the best options to choose in order to optimize cost.

2.2 A system-level cost model for 3D ICs

The proposed model allows making comparison between a 2D system and its 3D versions in terms of cost. In this work, we compare 3 stacking approaches: the W2W, the D2W and the interposer-based stacking (IbS) (figure 2.5). As introduced in chapter 1, the IbS approach is to stack several dies (fabricated in an aggressive technology such as 32nm) on top of a silicon interposer (fabricated in a mature technology such as 130nm) in order to improve fabrication yield. When using the W2W approach, it is necessary that the stacked dies have the same size. Otherwise, it would be impossible to slice the wafer to obtain the final 3D circuits. This problem does not arise when using the D2W or the IbS approach. In this comparison, we assume that all the stacked dies are different (each die has its own mask set).

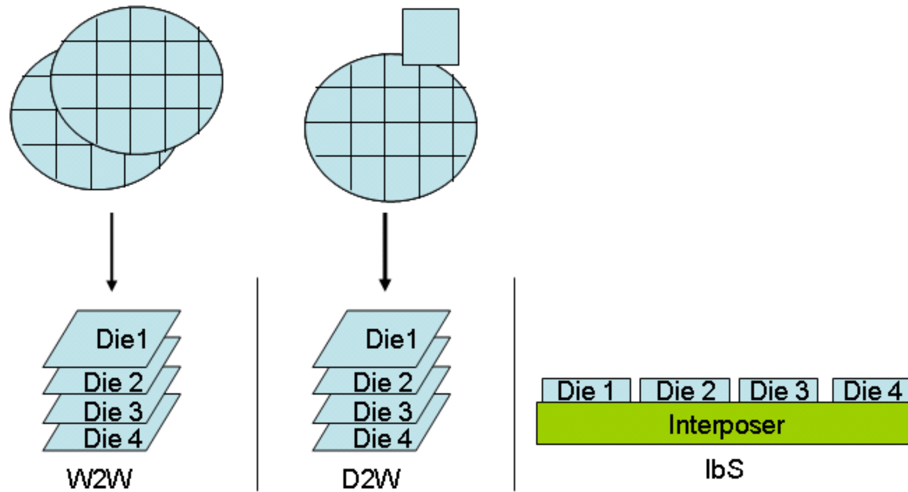


Figure 2.5: The investigated stacking approaches

3D integration requires extra-fabrication including TSV forming, wafer/die thinning, and wafer/die bonding. In order to separate the die cost model and the 3D stacking cost model, we assume TSV-last approach is used in 3-D IC fabrication process. In order to support multiple-layer stacking, the chosen stacking mode is F2B. In addition, the entire 3D-stacked chip cost depends on whether die-to-wafer (D2W), wafer-to-wafer (W2W) or interposer-based (IbS) stacking is used. If D2W or IbS approaches are selected, cost of known-good-die (KGD) test should also be included. To get rid of fabrication details common to 2D and 3D, we assume that the wafer fabrication cost is constant for a specific foundry using a specific technology node. The number of dies per wafer may be given by:

$$N_{die/wafer} = \frac{\pi \times (D_{wafer}/2)}{A_{die}} - \frac{\pi \times D_{wafer}}{\sqrt(2A_{die})}$$

where D_{wafer} is the wafer diameter and A_{die} is the die area. Hereafter, we present our cost models for the 2D and the different 3D approaches.

2D IC cost model

The final 2D IC cost my be given by:

$$C_{2D} = \frac{C_{fab} + C_{test}}{Y_{2D}}$$

where:

- C_{fab} is the 2D IC fabrication cost,
- C_{test} is the final test cost,
- Y_{2D} is the fabrication yield.

The fabrication yield and the 2D IC fabrication cost may be given by:

$$Y_{2D} = (1 + \frac{A_{2D} \times D_0}{\alpha})^{-\alpha}$$

$$C_{fab} = \frac{C_{wafer}}{N_{die/wafer}} + \frac{C_{mask}}{N}$$

where:

- A_{2D} is the 2D IC area,
- C_{mask} is the mask cost,
- C_{wafer} is the wafer fabrication cost for a specific foundry using a specific technology node,
- N is the total number of 2D ICs (or total production volume),
- D_0 is the density of point-defects per unit area,
- α is a model parameter, and typically ranges from 1.0 to 5.0.

3D IC cost model

$C_{3D,W2W}$, $C_{3D,D2W}$ and $C_{3D,IbS}$ are the final 3D IC costs when using the wafer-to-wafer, the die-to-wafer and the interposer-based stacking approaches respectively. They are given by:

$$C_{3D,W2W} = \frac{\sum_{i=1}^{L} C_{die_i} + (L-1) \times C_{stacking,W2W} + C_{test,W2W}}{(Y_{stacking,W2W})^{L-1} \times (\prod_{i=1}^{L} Y_{die_i})}$$

$$C_{3D,D2W} = \frac{\sum_{i=1}^{L} (C_{die_i} + C_{test,die_i})/Y_{die_i} + (L-1) \times (C_{stacking,D2W} + C_{test,stacking,D2W})}{(Y_{stacking,D2W})^{L-1}}$$

$$C_{3D,IbS} = \frac{\sum_{i=1}^{L} (C_{die_i} + C_{test,die_i})/Y_{die_i} + (C_I + C_{test,I})/Y_I + L \times (C_{stacking,IbS} + C_{test,stacking,IbS})}{(Y_{stacking,IbS})^{L}}$$

where:

- $C_{stacking,W2W}$, $C_{stacking,D2W}$ and $C_{stacking,IbS}$ are the stacking costs (including all the steps of the 3D fabrication process) when using the wafer-to-wafer, the die-to-wafer and the interposer-based stacking approaches respectively,
- $Y_{stacking,W2W}$, $Y_{stacking,D2W}$ and $Y_{stacking,IbS}$ are the stacking yields when using the wafer-to-wafer, the die-to-wafer and the interposer-based stacking approaches respectively
- $C_{test,W2W}$ is the final test costs of the 3D IC when using the wafer-to-wafer approach,
- $C_{test,die}$ and $C_{test,I}$ are the costs of testing the die and the interposer respectively,
- C_{die} and C_I are the fabrication costs of the die and the interposer respectively,
- Y_{die} and Y_I are the fabrication yields of the die and the interposer respectively,
- L is the number of stacked dies,
- A_{3D} is the final die area (including TSV area overhead),
- A_{TSV} is the TSV area overhead,

C_{die} and C_I are given by:

$$C_{die} = \frac{C_{wafer}}{N_{die/wafer}} + \frac{C_{mask}}{N}$$

$$C_I = \frac{C_{wafer}}{N_{I/wafer}} + \frac{C_{mask}}{N}$$

Y_{die} and Y_I are given by:

$$Y_I = (1 + \frac{A_I \times D_0}{\alpha})^{-\alpha}$$

$$Y_{die} = (1 + \frac{A_{3D} \times D_0}{\alpha})^{-\alpha}$$

A_{TSV} is given by:

$$A_{3D} = \frac{A_{2D}}{L} + A_{TSV}$$

Test cost model

According to [29], test cost may be modelled as the product of the cost of tester use per second and the average IC test time. Tester-use cost (per die) may be then given by:

$$C_{test} = R.T_{test}$$

where R is the cost rate (euros per second) for a tester, and T_{test} is the average IC test time. According to [29], test execution time may be considered as proportional to the die area. Besides, the average test time may be considered as depending on yield, because test time is shorter for a failing die than for a good die. Indeed,

testing usually terminates upon first failures. As a result, the average time required to test a single IC is:

$$T_{test} = T_{setup} + [Y + \beta(1 - Y)]K.A$$

where T_{setup} is the setup time for an IC on the tester, β is the average ratio between good-die test time and defective IC test time, and K is a constant multiplier that relates test time to IC die area A. Both β and K may be extracted based on regression analysis of historical data on test execution times of various products. A less-than-1 β means that the entire test sequence needs not to be applied to a failing die.

In the case of 3D ICs, test time includes also time required to test vertical interconnects (TSVs). This time may be considered as proportional to the number vertical interconnects per die. Consequently, TSVs test time may be given by:

$$T_{test,TSV} = H.N_{TSV}$$

Where N_{TSV} is the number of TSVs per die, and H is a constant multiplier that relates TSV test time to their number.

For the W2W stacking, test time may be given by:

$$T_{test} = T_{setup} + L[Y + \beta(1 - Y)]K.A_{die} + (L - 1)H.N_{TSV}$$

For the D2W stacking, die testing and TSVs testing times may be given by:

$$T_{test,die} = T_{setup} + [Y + \beta(1 - Y)]K.A_{die}$$

$$T_{test,TSV} = T_{setup} + H.N_{TSV}$$

Finally, for the IbS stacking, die testing and stacking testing times may be given by:

$$T_{test,die} = T_{setup} + [Y + \beta(1 - Y)]K.A_{die}$$

$$T_{test,TSV} = T_{setup} + H.N_{Microbumps}$$

where $N_{Microbumps}$ is the number of microbumps per die. Table 2.1 indicates the values of R, β, K and H used for our cost analysis.

Table 2.1: Constant values used for test cost model

Parameter	Value
β	0.3
R (€/second)	2
K (second/mm^2)	0.05
H (second/TSV)	0.001
T_{setup} (second)	0.5

3D stacking cost model

Unlike the test cost, it is a complicated task to elaborate a system-level cost model for 3D stacking. This is due to the variety of 3D technological options, which makes the final cost of 3D stacking depending directly on the 3D fabrication process used. Besides, it is quite hard to find realistic information about the cost of the different steps of a particular 3D fabrication process, since these information are rarely published by the industrial community. One of the few publications that provides this type of information is [30] of John H. Lau from the Industrial Technology Research Institute of Taiwan. This paper deals with TSV manufacturing yield and cost for 3D IC integration. It presents some realistic values of TSV forming and wafer bumping costs for a typical 3D fabrication process. Further details about this process are provided in the paper [30]. Our stacking cost model is based on information provided by this paper [30]. The cost of 3D stacking includes the costs of the different steps of a typical 3D fabrication process: wafer thinning, TSV forming and wafer (or die) bonding. Therefore, the 3D stacking cost model for the W2W, the D2W and the IbS may be given by:

$$C_{stacking,W2W} = \frac{C_{TSV,wafer} + C_{bumping} + C_{bonding}}{N_{die/wafer}}$$

$$C_{stacking,D2W} = \frac{C_{TSV,wafer} + C_{bumping}}{N_{die/wafer}} + C_{bonding}$$

$$C_{stacking,IbS} = \frac{C_{TSV,wafer} + C_{bumping}}{N_{die/wafer}} + C_{bonding}$$

where $C_{TSV,wafer}$ is the TSV manufacturing cost per wafer, $C_{bumping}$ is the is the wafer bumping cost, and $C_{bonding}$ is the die or wafer bonding cost. We consider that bonding a single die on top of a wafer takes the same time as bonding a wafer on top of another wafer. Therefore, bonding a single die or a whole wafer on top of another wafer has almost the same cost. Table 2.2 depicts the values of $C_{TSV,wafer}$, $C_{bumping}$ and $C_{bonding}$ used for our cost analysis, considering a 300mm wafer [30].

Table 2.2: Costs of different 3D stacking steps

Step	Value
$C_{TSV,wafer}$	280 €
$C_{bumping}$	200 €
$C_{bonding}$	10 €

Implementation of the cost model

The previously described cost models were implemented using the Excel tool. Excel is a well-known widely used tool that provides a simple and easy-to-use interface. As it offers all the mathematical functions needed by our cost models, it is simpler and quicker to use compared to standard programming languages (JAVA, Visual Basic...). In order to allow the exploration of the economical trends of 3D integration, the developed Excel sheet may be easily configured by new values for the technological parameters.

2.3 Cost analysis of 3D ICs

This analysis is based on data of tables 2.3 and 2.4. The stacking yields for all the 3D approaches are set to 99%, and production volume is set to 1 million.

Table 2.3: Technological parameters for 32nm-technology die

Parameter	Value
α	1
Defect density	$2.10^{-2}/mm^2$
Mask cost	3,500,000 €
Wafer cost (300mm)	8,000 €

Table 2.4: Technological parameters for 130nm-technology interposer

Parameter	Value
α	1
Defect density	$2.10^{-4}/mm^2$
Mask cost	400,000 €
Wafer cost (300mm)	2,000 €

Cost variation according to area

Figure 2.6 shows the variation of unit cost for a 2D chip and its equivalent W2W, D2W and IbS 3D 2-layer circuits, when die area increases.

As depicted in figure 2.6, for small-sized circuit, the 2D approach is more economical than any of the 3 stacking approaches. For example, considering a $50mm^2$ design, the W2W, D2W and IbS 3D schemes increase cost by 90%, 56% and 120% respectively compared to the 2D approach. This may be explained by the very high yield of small-sized circuits that makes inefficient any further reduction of the circuit size. 3D stacking includes additional technological steps in manufacturing process (and then extra-fees), without significantly improving fabrication yield.

In the case of large-sized design, the D2W and IbS 3D stacking become the most cost-effective approaches. Besides, cost gain increases when the design area increases. As an illustration, the D2W stacking allows reducing cost (compared to the 2D approach) by 12% and 20% when the design area is $300mm^2$ and $600mm^2$ respectively. This can be explained by the low yield of large circuits, which makes so advantageous to partition the 2D circuit into smaller dies, and to test these obtained dies before stacking (known-good die test). However, the W2W scheme remains more expensive than the 2D approach. Indeed, because of low yield (due to large circuit area), die stacking without testing turns out to be economically inefficient.

It could be concluded that 3D stacking involves extra-fees due to additional steps of the 3D fabrication process (such as bonding, TSV formation, known-good-die test...), but also allows cost reduction by reducing the area of the stacked dies and then improving yield. Therefore, the 3D approach is more cost effective in the case of large-sized circuits, when using the D2W or the IbS integration schemes.

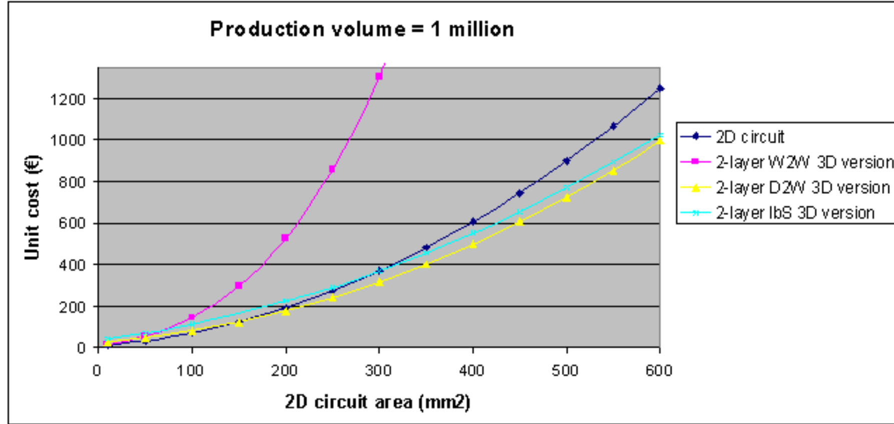


Figure 2.6: Unit cost for a 2D circuit and its W2W, D2W and IbS 3D versions for different die areas

Cost variation according to the number of layers

Figures 2.7, 2.8 and 2.9 depict cost variation for different numbers of layers when using the W2W, D2W and IbS 3D schemes.

For the W2W approach, the 3D cost increases considerably when the number of layers increases. For example, considering a $200mm^2$ design, the cost of the W2W 3D IC increases by 2 times, 7 times and 12 times (compared to 2D approach) when the 2D initial design is partitioned across 2, 4 and 6 layers respectively. This is due to low yield of aggressive technologies that makes it indispensable to test the dies before stacking them, especially for large-sized circuits.

For the D2W and IbS approaches, the 3D cost increases with the number of layers in the case of small-sized circuit, due to their high fabrication yield. For the D2W approach, the cost of a $50mm^2$ design increases by 55% for the 2-layer 3D version, and by 300% for the 6-layer 3D version (compared to the 2D approach). However, in the case of large-sized design, the cost of the D2W or IbS decreases when the number of layers becomes more and more important. When using the D2W approach, the cost of a $600mm^2$ circuit decreases by 20% for the 2-layer 3D version, and by 40% for the 6-layer 3D version (compared to the 2D approach). This could be explained by the low yield of large designs, which makes it so beneficial to partition the circuit into several small dies, and to test them before stacking.

It could be concluded that the optimal number of 3D layers (in terms of cost) depends on how large the design it is.

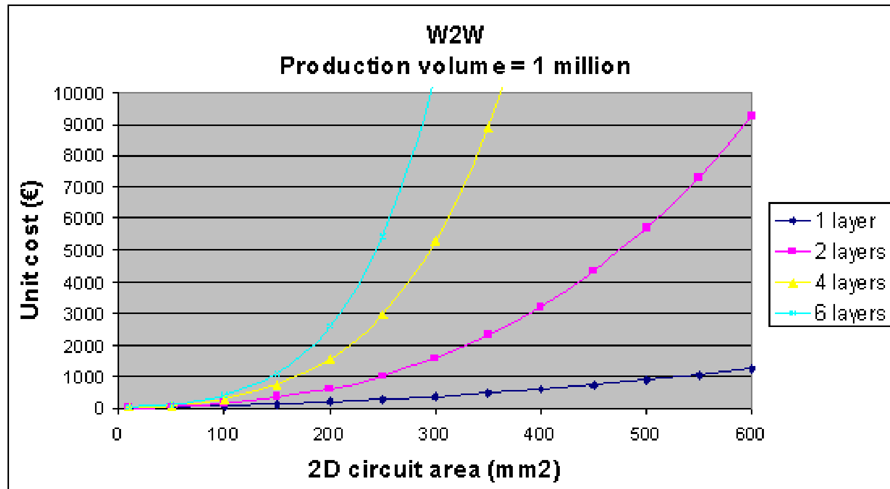


Figure 2.7: Unit cost for different numbers of layers using the W2W scheme

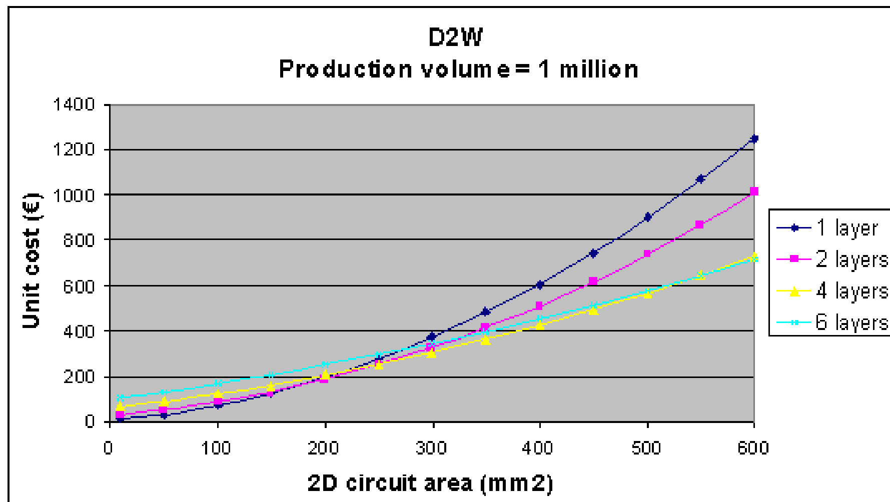


Figure 2.8: Unit cost for different numbers of layers using the D2W scheme

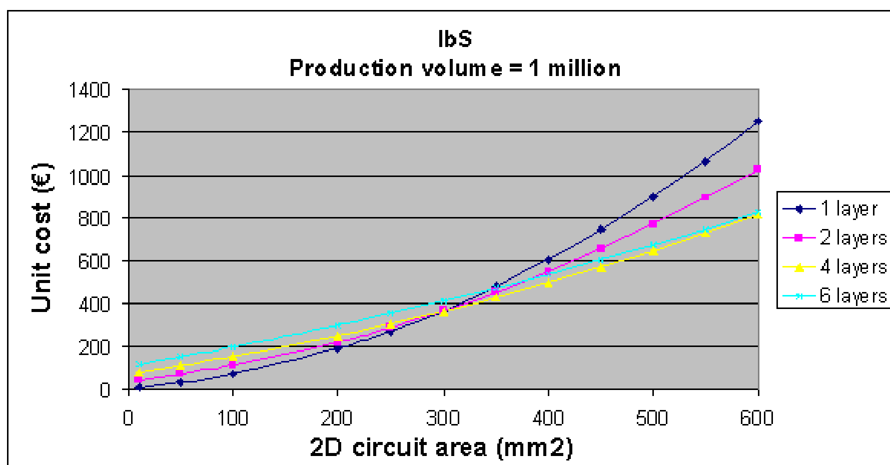


Figure 2.9: Unit cost for different numbers of layers using the IbS scheme

3 Conclusion

In this chapter, some major challenges related to 3D IC partitioning and cost are introduced and discussed.

3D IC partitioning may be performed according to 3 different granularities: macroscopic blocks, functional units and basic operators. All of these 3 types of partitioning allow improving the performance and power with rates that increase when the partitioning granularity decreases. In the same time, challenges such as redesign efforts, thermal stress... become more and more noticeable when miniaturizing the size of the stacked elements. Taking into account 3D IC economical and technical limitations, macro-block stacking seems to be the most viable approach. For this reason, this thesis is oriented to focus on this partitioning granularity, and more specifically with interconnection architecture inside macro-block partitioned 3D systems. We propose a new 3D NoC topology that allows performance enhancement. Further details concerning this work are provided in chapter 3.

Another critical challenge for 3D ICs is cost. A system-level cost analysis model is proposed to investigate the economical trends of 3D integration. Based on this cost model, 3D stacking turns out to be cost-effective (compared to classical 2D approach) in the case of large-sized circuits when using the D2W or the IbS integration schemes. Another observation is that the optimal number of layers (in terms of cost) depends on how large the design is when using the D2W or the IbS integration approaches. The second contribution of this thesis focuses on cost-aware 3D architectures. Using this same cost model, we investigate the cost-effectiveness of the 3D same-die stacking approach, with a real case study on 4G telecom applications. More details about this work are provided in chapter 4.

As said previously, next chapter proposes a 3D NoC architecture in order to improve communication performance within macro-block partitioned 3D circuits.

Chapter 3

A hierarchical NoC for 3D MPSoCs

The previous chapter deals with the pros and cons of the 3 different partitioning granularities of 3D ICs. For economical and technical viability reasons, this thesis focuses on macro-block partitioning approach. At this level of partitioning, the involvement of 3D integration technology is limited to the vertical inter-block interconnections. Several interconnect architectures could be used to perform communication between cores on different tiers. Depending on the circuit requirements in terms of performance and cost, each architecture present advantages and drawbacks. Nevertheless, considering the increasing complexity of current MPSoC, and the substantial communication needs within 3D architectures, 3D Network-on-Chip is becoming more and more accepted as a promising solution to meet the performance requirements of 3D MPSoCs.

This chapter presents a new 3D NoC router in order to enhance throughput and latency compared to classic 3D mesh NoC. The proposed router is hierarchical as it is composed of 2 completely decoupled modules: one for inter-layer communication and one for intra-layer communication. It is fully implemented in asynchronous logic in order to allow low latency transfer. This chapter is organized on 5 sections. The first one explains the motivations for NoC architecture, and some of its main principles. The second section presents the pros and cons of several state-of-the-art 3D NoC architectures. Section 3 is devoted to describe the proposed 3D hierarchical NoC and the design of the asynchronous router. Finally, sections 4 and 5 present an evaluation of the proposed NoC in terms of area, power and performance.

1 NoC paradigm

Technological advances and the increasing complexity of applications lead designers to propose electronic systems that integrate several functions and processing units. Initially, the possibility to design a complete system on a chip (SoC) has overcome the performance limitations of classic printed circuits, where several discrete components are interconnected by printed wires. Today, with the growing number of integrated blocks and the increasing exchange of information on the chip, innovative interconnect structures are being studied in order to meet those new needs [31, 32].

1.1 Evolution of interconnection structures

On-chip communications have been traditionally provided by direct connections and shared buses (figure 3.1 (a) and (b)). A direct connection is to physically connect a transmitter to a receiver. Direct connections provide the best performance in term of bandwidth. However, they are difficult to design and to reuse in the case of complex systems that have a large number of processing units [33]. A bus on the contrary may be easily reused, but it allows only a single communication on a time between modules that are connected to it (thanks to the Time Division Multiplexing technique). As a result, the bandwidth is shared between those modules. Consequently, communication performance decreases when the number of modules increases. An arbitration mechanism is needed to manage simultaneous access [34]. Moreover, a bus requires significant wire lengths in the case of large-sized systems. Therefore, its power consumption is important since data is propagated along the bus. Besides, wire capacitive effects cause synchronization problems and limit the bus frequency [35]. To conclude, direct connection and bus solutions are not scalable with the number of connected components.

The network-on-chip (NoC) has been proposed to solve reusability and scalability problems. It provides a communication architecture that can be standardized and then easily reusable. Besides, the NoC offers a distributed communication architecture without overall control on the system state. Therefore, it allows integrating a large number of modules having different functionalities, while keeping the same bandwidth.

Table 3.1 summarizes key elements of comparison between bus and NoC architectures. These comparison criteria are frequently cited in articles dealing with the subject [32, 36].

1.2 Principles of NoC

The principles of NoCs are based mainly on theory and experience of computer network architectures and multiprocessor systems. They take into account also silicon technology constraints.

Communication system

As depicted in figure 3.1, a NoC is composed of several nodes (also called switches or routers) connected by direct communication links (or channels). Each node includes several ports for connection with its neighbor nodes and the functional element

Table 3.1: Comparison between NoC and bus architectures

Criteria	Bus	NoC
Layout	(-) Each unit creates parasitic capacitances that degrade the bus performance	(+) Physical and electrical properties are predictable thanks to its regular geometry
	(-) Wires are global	(+) Wires are point-to-point connections
Frequency	(-) The frequency is limited by the length of interconnect wires	(+) The unidirectional and point-to-point wires allow a high frequency functioning
Power	(-) Data must be sent to all potential receivers resulting in an unnecessary high power consumption	(+) It is possible to cut power to some areas of the network
Time	(+) The transfer time is constant and relatively short	(-) The transfer time increases proportionally to the number of nodes
	(-) The access time is proportional to the number of masters connected to the bus	(+) The node access time is short
Arbitration	(-) Arbitration mechanisms degrade bus performance	(+) Routing decisions are distributed and based on local information
	(-) The arbitration delay increases with the number of connected modules	(+) The node routing delay is the same regardless the network size
Design	(+) The design is simple and well understood. Different standards may be used to integrate many compatible IPs	(-) Designers have to get used to new concepts brought by the NoC
Blocking	(+) There is no deadlock problems	(-) Deadlocks are possible

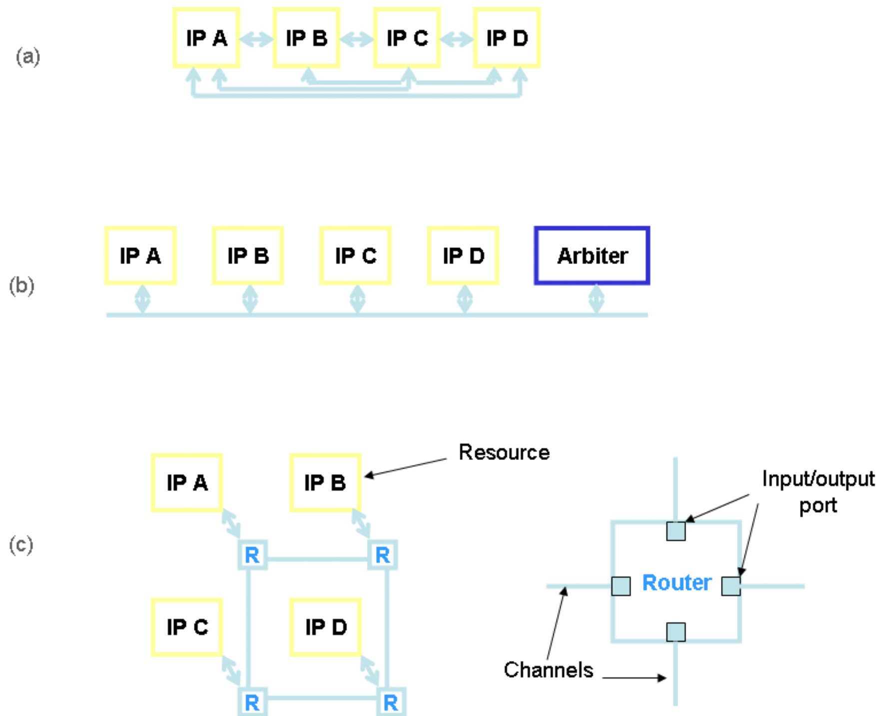


Figure 3.1: Interconnect structures: (a) direct connections, (b) bus, (c) NoC

of the NoC (resources). The resources (the functional elements) communicate by exchanging messages. At the NoC level, the message is called a packet. A packet consists of a header that often contains information on the packet nature and its destination, a data field (payload) that includes the useful information carried by the packet, and a trailer that indicates the end of the packet.

Topologies

The topology specifies the physical organization of the NoC, ie. how network nodes are connected together. Several network topologies have been proposed in the state-of-the-art works (figure 3.2): ring, tree, mesh, torus, fat-tree...

These topologies can be compared according to different criteria such as technological implementation constraints. Indeed, the NoC is proposed as a new flexible and scalable interconnection structure. It is so important to limit its cost in terms of implementation complexity and hardware resources. Ring topologies are easy to implement, but do not provide good performance when multiple messages are transmitted simultaneously. Fat-tree topologies have a high bandwidth, but require a more complex routing and longer interconnections between nodes. Mesh networks seem to provide the best performance-complexity compromise and to be the most adapted to current needs of NoC architectures. Indeed, they allow more easily closing resources that communicate intensively together, which reduces routing complexity. Therefore, these resources would make use of only a small number of NoC routers, which improves performance.

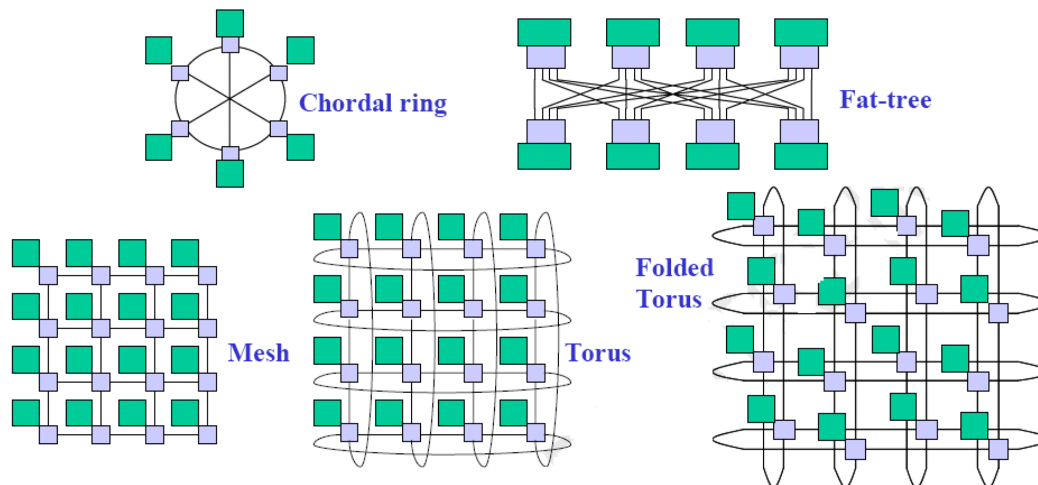


Figure 3.2: Different NoC topologies

Communication mechanism

The communication mechanism specifies how messages are transmitted over the network. There are two main methods to perform this transfer: circuit switching and packet switching.

In the case of circuit switching, several network resources (links and nodes) are allocated for a given transfer between the source node and the destination node. These resources can not be used by other users during the transfer time. Thus, there is no blocking. Physically, the nodes are simple as they only provide a link between an input port and an output port. The control information (start / end of transfer) are separated from data. This mechanism is suitable for important data transfers in order to minimize time spent negotiating the connection. Circuit switching is used for service guarantees.

In the case of packet switching, packets are sent without any previously established connection. This allows sharing network resources. Packets must contain information about their destinations: the control information is sent along with the data. Nodes are often more complex: they sometimes have to modify the contents of packages to update routing information, for example. Consequently, the network latency is greater. Specific mechanisms must be introduced to manage bottlenecks and priorities. As packets may arrive erratically, transfer time is less guaranteed.

Switching mode

In the case of packet switching, the switching mode means how packets will pass from one network node to the next [37].

In the case of store-and-forward switching mode, a network node can send a packet only after receiving it completely. Therefore, communication latency is large. Each node must be able to store an entire packet, and thus requires large memory capacities.

In the case of virtual cut-through switching mode, a node can send a packet if the next node guarantees that it can store entirely this packet, otherwise the node must be able to keep the packet. Thus, the storage capacity of the node is the same

as for the store-and-forward mode. However, latency is reduced as there is no need to await the receipt of the complete packet.

The wormhole mode is intended to minimize memory size of the network nodes. The packets are divided into smaller units called flits (flow control unit). The flits pass from node to node as soon as there is memory space for a flit and not necessarily for a complete packet. The header flit contains the destination information; all the remaining flits of the same packet must then follow the same path. The same packet can be distributed over several network nodes. Wormhole switching therefore reduces latency and memory requirements, but increases the risk of bottlenecks and contention in the network.

Storage strategy

As seen in the preceding paragraph, the routing nodes of a network have storage buffers to meet their memory requirements. There are different ways to position the storage buffers with respect to the input / output ports [37].

In the output queuing storage strategy, each output port has as much buffers as the number of input ports. It is possible for each input port to write in a buffer of any output port. This strategy offers the best performance, but it requires a lot of connection wires and an important number of buffers (For a node with N input/output ports, N^2 buffers are required).

In the input queuing strategy, a buffer queue is placed at each input port. An arbitration system deals with connecting these buffers to the output ports while avoiding contentions. Therefore, a node with N input /output ports require only N buffers. The disadvantage of this strategy is that in the case of heavy traffic, it quickly saturates at less than 60% of the theoretical utilization capacity because of head-of-line blocking [37]. This occurs when data in the lead does not have access to an output port and blocks following data in the queue. This strategy does not allow optimum use of resources.

The virtual output queuing strategy aims to combine the advantages of the two previously mentioned approaches. The idea is to use multiple buffer queues on input ports. Packets are distributed on these buffer queues according to their destinations or a priority level. This increases the utilization rate of the node by reducing the phenomenon of head-of-line blocking. This strategy can reduce Static deadlocks thanks to virtual channels.

Static deadlocks occur when multiple packets are mutually blocked due to a cyclical dependency: each node is waiting to send a packet to a node that is already waiting. For example, in a wormhole routing, a packet may occupy several nodes on the same time. If multiple packets have flits across multiple nodes, the situation can be blocked as shown in figure 3.3. Virtual channels are utilized to avoid cyclical dependencies. To do so, we use the virtual output queuing storage strategy. Hence, for each physical channel, there are many buffer queues that correspond to virtual channels which share the same physical channel bandwidth. In this way, a packet temporarily blocked and stored on multiple nodes can be overpassed by another packet using the same physical channel but stored on another channel virtual (buffer queue).

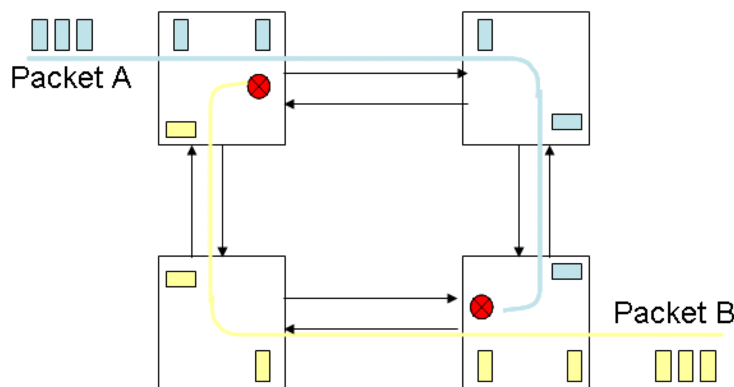


Figure 3.3: A static deadlock

Routing algorithm

Routing determines which set of network nodes a packet will pass through. Routing can be determined at the source (source routing) if the transmitter node defines in the packet header the nodes it will follow. Routing can be also distributed, if each node chooses the packet direction based on its destination address and other criteria. The routing is deterministic if it depends only on the sender and the receiver. On the contrary, it is adaptive if the path can vary depending on traffic. Adaptive routing is used in the case of irregular traffic applications. If data exchanges are highly predictable, deterministic routing is preferable and simpler to implement.

NoC quality of service

A NoC have to guarantee several services such as data integrity (meaning that data is transmitted without alteration), data ordering (meaning that data are received in the order they were issued) and no data loss during packet transmission through the network [37].

In this work, we focus on mesh-topology NoCs, with wormhole packet switching mode, virtual output queuing storage, and deterministic source routing.

2 3D NoC: state of the art and challenges

In order to ensure effective communication between the cores on different tiers, designing inter-layer interconnects requires a significant attention. Recently, several studies carried out in academia and industry have looked at designing efficient interconnect architectures, based mainly on the NoC approach [38–40]. 3D NoCs are rising as a good solution to manage efficiently complex interconnections in 3D MPSoCs. This section briefly relates recently proposed 3D on-chip network architectures to the hierarchical 3D NoC presented in this chapter.

2.1 3D mesh NoC

A simple way to extend the 2D 5x5 router (5 input ports x 5 output ports) to the vertical dimension is to add one port for upward traffic and one port for downward traffic [41] (figure 3.4). Of course, it would be necessary to extend also buffers,

arbiters, and crossbar. Although this 3D mesh approach is simple to design and implement, it has several major problems. Indeed, adding two ports to each router would require a larger 7x7 crossbar which leads to significant area, power and latency overheads compared to a 2D 5x5 router. Many researches have been carried out to avoid these issues and to improve the performance of 3D NoCs. They can be mostly categorized into router improvement and topology optimization.

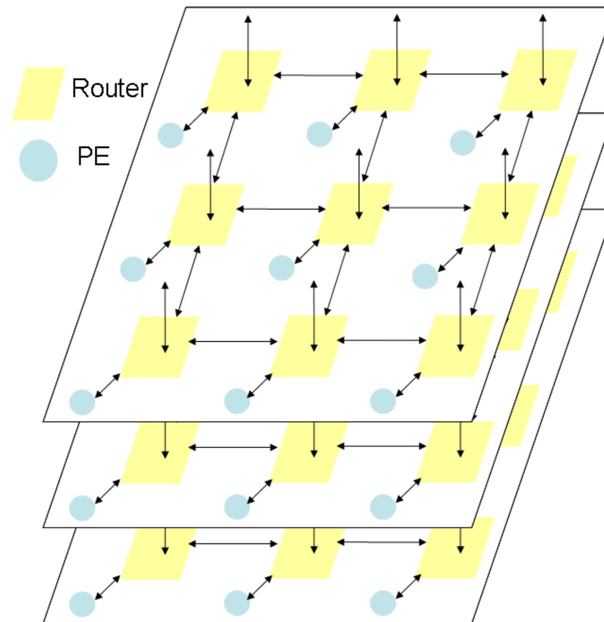


Figure 3.4: 3D mesh NoC architecture

2.2 3D NoC-Bus

To solve the challenges of 3D mesh architecture, Li et al. propose to use a 6x6 router for intra- layer communication and a vertical bus for inter-layer communication [42] (figure 3.5). This approach requires adding only one physical port to each 5x5 router to connect it to the vertical bus. Nevertheless, given that the bus is a shared medium, it cannot carry more than one flit at a time. Under high inter-layer load, the vertical shared bus becomes the throughput bottleneck.

2.3 DimDe router

Kim et al. [41] present a dimensionally decomposed 3D crossbar design called DimDe (figure 3.6). The main idea is to decompose the incoming traffic into three independent streams: east-west traffic, north-south traffic and up-down traffic, and to use three smaller crossbars for the three flows. The resulting three compact crossbars are more efficient in terms of area, power and performance than the conventional monolithic approach. However, as mentioned in [41], the DimDe design can use 2-stage arbitration only when using deterministic dimension-order-routing, which is known to be a non-adaptive algorithm unable to avoid congestion. If a different routing algorithm is used with the DimDe architecture, a more complex 3-stage

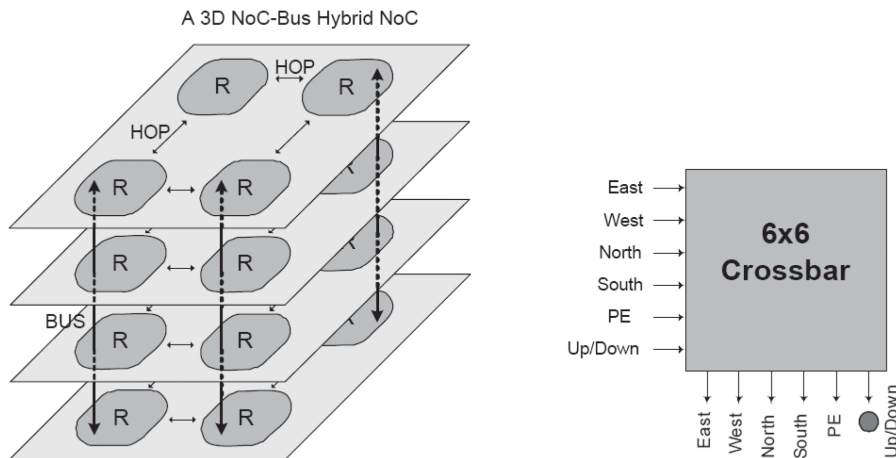


Figure 3.5: 3D NoC-bus router

arbitration scheme would be required. This would affect the performance of the DimDe approach.

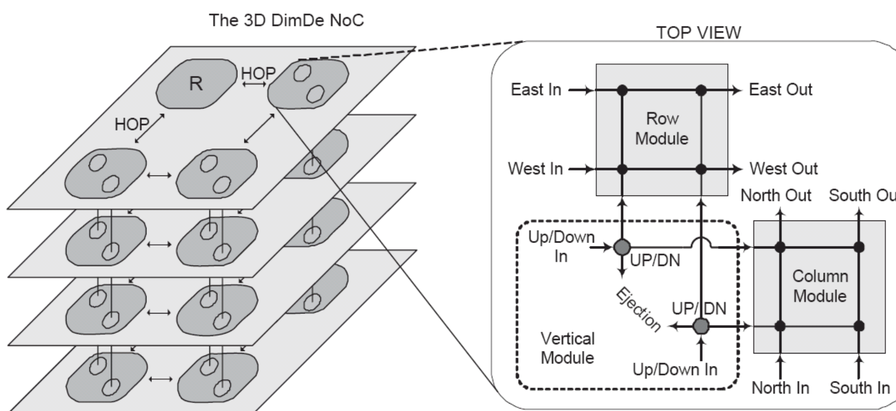


Figure 3.6: DimDe router

2.4 MIRA router

Park et al. [43] propose a 3D stacked router architecture 3.7. This router is partitioned across multiple layers in order to reduce the overall area and power consumption. This means the distribution of the different router functionalities across multiple layers. As mentioned previously, with current 3D integration technology, such a fine-granularity partitioning is considered not viable.

2.5 Other 3D NoC architectures

Chen et al. [44] introduce a new 3D NoC architecture based on De Bruijn graph, which is a topology with small diameter, simple routing and good reliability. Although the De Bruijn graph based architecture can achieve smaller network latency than Mesh based architectures, it has higher power consumption than Mesh topologies in most cases.

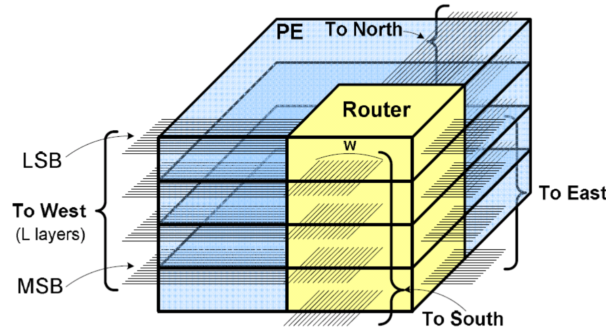


Figure 3.7: MIRA router

Xu et al. [45] deal with developing a 3D NoC topology of low latency. This purpose is achieved using a low-diameter network with long wires and low-radix routers. As this topology uses long links, it can be scaled to larger networks only when using techniques such as network concentration.

Yan et al. [46] focus on designing 3D-NoC architectures optimized for a given application. They present 3D-NoC synthesis algorithms based on the rip-up and reroute concept that has been used in the VLSI routing problem. These algorithms use power and delay models for 3D wiring with through-silicon vias in order to build application-specific 3D NoC topologies.

3 Architecture of the asynchronous hierarchical 3D NoC

3.1 Architecture of the 3D NoC

The architecture of the classic 3D mesh consists of several routers interconnected by direct links as depicted in figure 3.4. Each router has 7 input/output ports to perform communications with the up, down, north, south, east and west routers and the processing element.

The hierarchical 3D NoC is to replace the 7x7 router by 2 totally decoupled routers: a 5x5 router (called the horizontal module) used for intra-layer communication and a 4x4 router (called the vertical module) used for inter-layer communication. All the 7x7, 5x5 and 4x4 routers have the same design; the only difference between them is the number of I/O ports. The 5x5 and 4x4 routers communicate together. The processing element is connected to the 4x4 router. An abstract view of the hierarchical 3D hierarchical router and the equivalent monolithic router is illustrated in figure 3.8.

The architecture of the hierarchical 3D NoC is depicted in figure 3.9. For inter-layer communication, a flit moving between layers has to cross a 4x4 router instead of a 7x7 router in the classic 3D mesh. In order to guarantee this advantage, the deterministic routing used in this work imposes that flits whose target resource is located on another layer are directed firstly to the targeted layer, and then to the target resource within this layer. For intra-layer communication, assume that a flit will cross n routers. In the 3D mesh NoC, this flit would cross n 7x7 routers. In the hierarchical architecture, it would cross a 4x4 router, n 5x5 routers and finally a 4x4

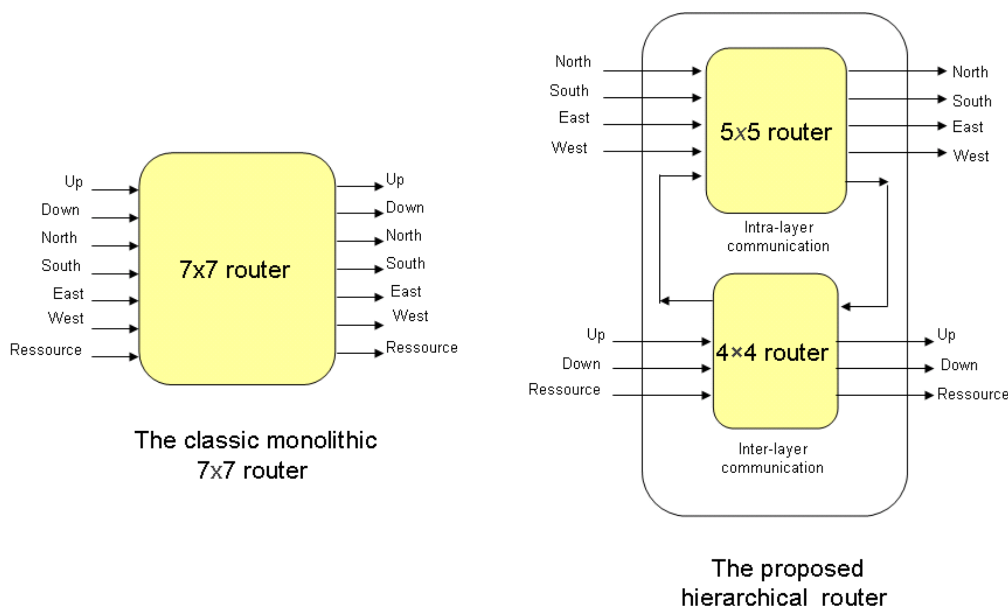


Figure 3.8: Conceptual view of the 3D hierarchical router

router. Thanks to this hierarchical and modular architecture, it would be possible to use the vertical module or the horizontal module (or both of them) according to the requirements of our application. For example, if a processing element need not to communicate with the upper or the downer layer, it will be connected only to the horizontal module (the 5x5 router). Similarly, if this processing element communicates only with cores located on other layers, it will be connected only to the vertical module (the 4x4 router). The major issue with the hierarchical approach is that a totally decoupled vertical module would force all packets moving within a particular layer to be twice (once at the sender and once at the receiver) buffered and arbitrated for access to the processing element. Our SystemC-TLM simulations show that, in large NoCs, this buffering and arbitration overhead can be compensated compared to classic 3D mesh due to the fact that the intrinsic latency of the 7x7 router is superior to those of the 4x4 or the 5x5 router.

The 3D hierarchical NoC is designed as a GALS (Globally Asynchronous Locally Synchronous) system: processing elements are synchronous units while routers are implemented in Quasi-Delay Insensitive asynchronous logic. The GALS architecture allows avoiding an important design challenge related to 3D circuit, which is global clock signal distribution (to perform multi-plane synchronization). Indeed, In order to distribute clock signal across a classic 2D circuit, symmetric interconnect structures such as H-tree are commonly used. Thanks to the symmetry of such structures, clock signal can arrive simultaneously at the leaves of the tree. This symmetry must be maintained within 3D circuits in order to ensure synchronous data processing. When simply extending the clock tree to the vertical dimension, the clock signal propagates through TSVs from the output of the clock driver to the centre of the clock tree on the other planes. Because the impedance of the TSVs, time required by the clock signal to arrive at the leaves of the tree on these planes may be larger than the time required by the clock signal to arrive at the leaves of the tree located on the same plane as the clock driver. Therefore, simply extending the clock tree to

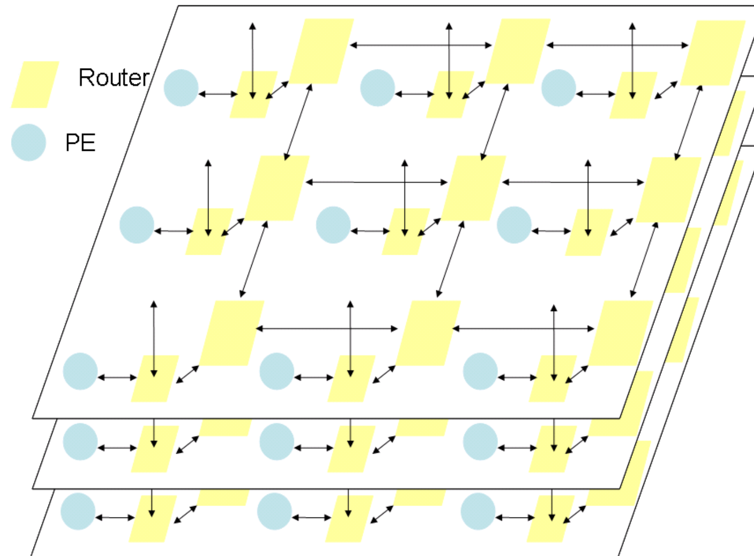


Figure 3.9: Architecture of the 3D hierarchical NoC

the vertical dimension does not guarantee equidistant interconnect paths from the root to all the leaves of the tree [47]. On the other hand, the heat generated by all layers of the 3D stack has to be dissipated through a heat sink. As layers away from the heat sink are thermally more insulated than other layers, temperature profiles across the layers of a 3D stack may vary. Therefore, the clock network that spans over several layers is exposed to different temperatures. According to [48], the clock skew increases when the temperature differences between the different chip parts increase. It would be necessary then to propose new solutions to mitigate the effect of temperature on clock skew.

Each processing element is connected to the Noc router through a network interface, which performs synchronization between the synchronous and the asynchronous domains. The two asynchronous 5x5 and 4x4 routers and their communication channels are presented in figure 3.10.

The whole network synchronization mechanism is based on a basic handshake between routers or between routers and resources to exchange a data flit (Figure 3.11). The meaning of the control signals is the following:

(i) send[i]=1: the data flit is valid for virtual channel i in the current cycle;

(ii) accept [i]=1: the receiver is ready to accept one data flit for virtual channel i at the next cycle.

The flit transfer condition is the following: the emitter block is allowed to transmit a new flit on virtual channel i with send[i]=1, if and only if, the receptor indicated accept[i]=1 at previous cycle [49].

Data are sent out in packets composed of several flits having the format shown in figure 3.12. These flits transit in the NoC according to a wormhole routing, following the path indicated in the header flit.

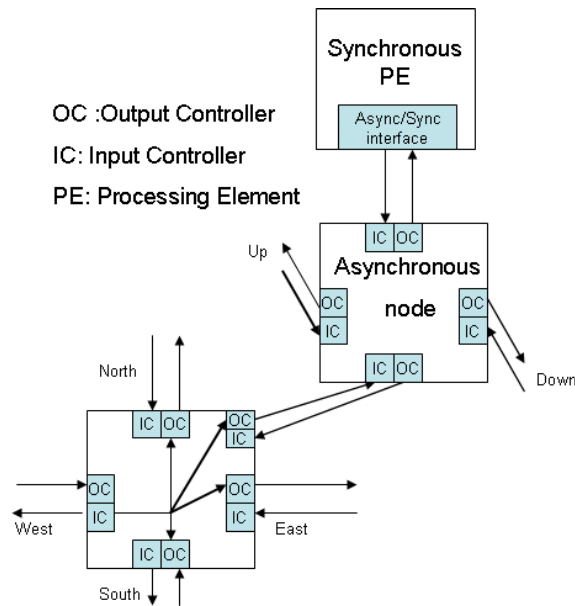


Figure 3.10: Architecture of the 3D hierarchical router

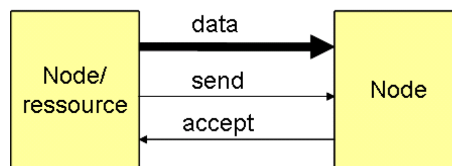


Figure 3.11: Send/Accept protocol

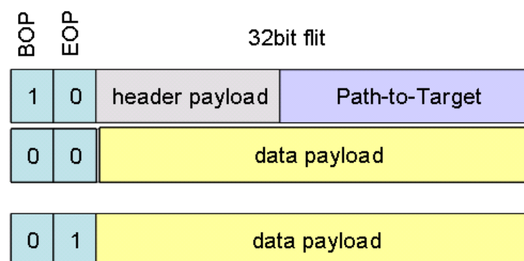


Figure 3.12: NoC packet format

3.2 Design of the asynchronous router

Motivations for asynchronous design

The majority of existing digital circuits are synchronous. This type of circuits is based essentially on a fundamental statement that significantly simplifies its design: all circuit modules have a common view of time, as defined by a single clock signal that is distributed all over the circuit.

On the contrary, asynchronous circuits are basically different as they do not share a common notion of time [50]. Instead, they make use of a handshaking protocol between their modules in order to achieve the required sequencing, communication and synchronization of elementary operations. As a consequence, asynchronous circuits present several intrinsic advantages compared to synchronous circuits:

- low power consumption thanks to the zero standby power consumption of modules,
- high operating speed since operating speed is determined by actual local latencies rather than global worst-case latency,
- better modularity because of the simple handshake interfaces and the local timing,
- no clock distribution and clock skew problems.

On the other hand, asynchronous circuits present also some disadvantages. Indeed, the asynchronous control logic that implements the handshaking protocol involves an overhead in terms of silicon area, circuit speed, and power consumption. Therefore, it is imperative to know whether the improvement (in terms of modularity, power consumption, operating speed, or clock distribution) resulting from the use of asynchronous techniques would cover these overheads. Other limitations are the lack of CAD tools mainly for testing and test vector generation.

Research in asynchronous design has begun since the mid 1950s. Nevertheless, it was not until the late 1990s that projects in academia and industry confirmed the possibility to design asynchronous circuits which show significant benefits in complex real-life applications. Thereby, commercialization of the asynchronous technology began to take place.

From clocking to handshaking

As depicted in figure 3.13, a synchronous circuit consists of several registers clocked by the same clock signal and used to stock data produced by combinatorial blocks. Sequential blocks are always running (and then consuming) when they are receiving the clock signal. However, this running and power consumption are not always required. Clock gating is a well-known solution to reduce power consumption due to clocking. Other important challenges are skew and power consumption of the clock tree, which must be controlled and minimized. Besides, timing constraints (due to the setup to hold time window around the clock edge) necessitates inserting buffers along the clock wires. Then, a lot of effort is spent on designing the clock gating circuitry, minimizing the skew and inserting buffers. All these challenges become more and more serious with today's large-sized and wire-dominated circuits.

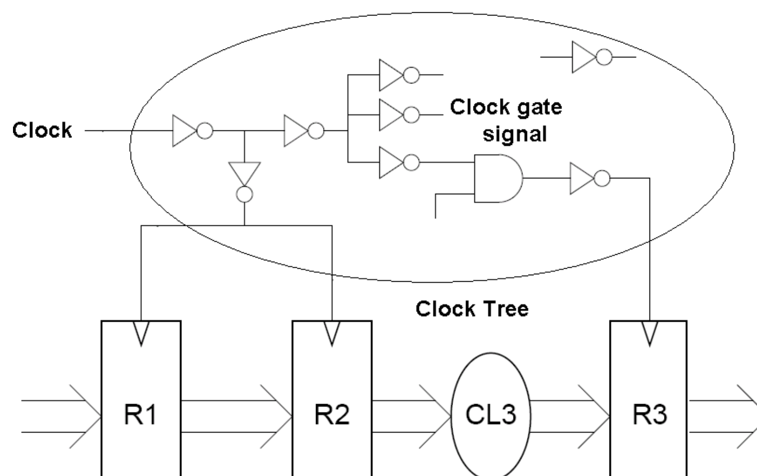


Figure 3.13: Synchronous circuit

In an asynchronous circuit, the clock signal is replaced by some form of handshaking between neighbouring registers. There are several handshake protocols [50].

A well-known one is the 4-phase bundled-data protocol depicted in figure 3.14. It is a simple return-to-zero request-acknowledge based handshake protocol. The term bundled-data means that data signals use boolean levels to encode information, and that request and acknowledge wires are bundled with the data signals. The term 4-phase refers to the number of communication actions. In the 4-phase protocol illustrated in figure 3.14: (1) the sender issues data and sets request high, (2) the receiver takes in the data and sets acknowledge high, (3) the sender responds by taking request low (at which point data is no longer guaranteed to be valid) and (4) the receiver acknowledges this by taking acknowledge low. At this point the sender may initiate the next communication cycle.

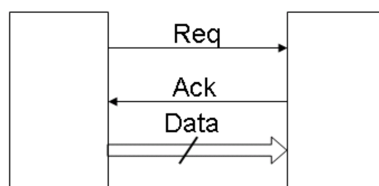


Figure 3.14: A bundled-data channel

Although 4-phase bundled data protocol is familiar to most digital designers, its major drawback is the unnecessary return-to-zero transitions that cost needless time and energy. The 2-phase bundled-data protocol shown in figure 3.15 avoids this. The information on the request and acknowledge wires is now encoded as signal transitions on the wires and there is no difference between a $(0 \rightarrow 1)$ and a $(1 \rightarrow 0)$ transition. They both represent a *signal event*. Therefore, the are only 2 actions: (1) the sender issues data and makes en event on the request signal, (2) the receiver takes in the data and makes an event on the acknowledge signal. Ideally, the 2-phase bundled-data protocol should lead to faster circuits than the 4-phase bundled-data protocol. Nevertheless, as the implementation of event-responding circuits is complex, no general answer as to which protocol is best is provided.

In the bundled-data protocols, the order of signal events at the sender's end

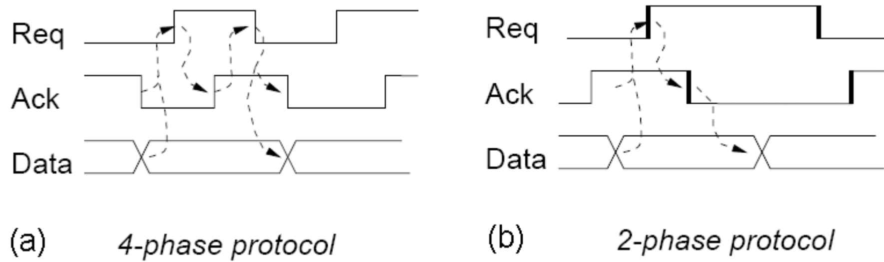
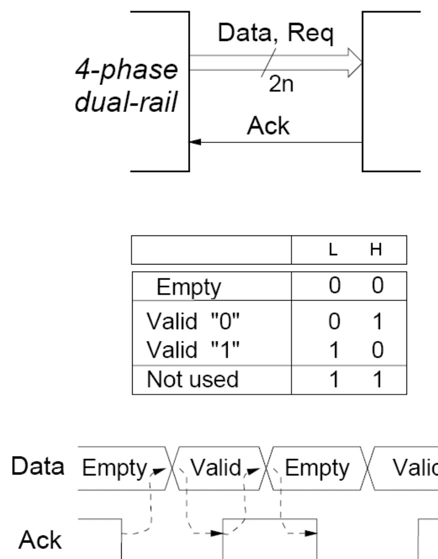


Figure 3.15: The 4-phase and 2-phase bundled-data protocols

should be preserved at the receiver's end (delay matching). Data is valid before request is set high, and this ordering should also be conserved at the receiver's end. To guarantee that, some possible physical solutions (performed when physically implementing such circuits) are to control the placement and routing of the wires (possibly by routing all signals in a channel as a bundle), to have a safety margin at the sender's end, or to insert and resize buffers after layout. An alternative to these physical solutions is to use a more sophisticated protocol that is robust to wire delays.

Instead of separating data and request signals, the 4-phase dual-rail protocol encodes the request signal into the data signals using two wires H and L per bit of information as depicted in figure 3.16. To signal a valid logic 1 (valid logic 0 respectively), H is set to 1 (0 respectively) and L is set to 0 (1 respectively). This protocol is very robust since both sender and receiver can communicate reliably regardless of delays in the wires connecting them. The protocol is then delay-insensitive.



	L	H
Empty	0	0
Valid "0"	0	1
Valid "1"	1	0
Not used	1	1

Figure 3.16: The delay-insensitive 4-phase dual-rail protocol

The previously described protocols are the most commonly used. Choosing the appropriate protocol depends on the performance we want to obtain (speed, robustness) and the cost we are willing to pay (area, power).

Micoarchitecture of the asynchronous router

The router (implemented in Quasi-Delay Insensitive asynchronous logic) deals with routing and arbitrating the packets within the NoC according to the information enclosed in the packet format depicted in figure 3.12. Given the hierarchical topology of the NoC, the router is made of n input ports and n output ports (n=5, 4); there is no need to a central arbitration. The router has 2 independent virtual channels (VC) to improve quality of service.

The input port is in charge of routing the successive packet flits according to the path to target specified in the header flit (marked with a BOP). As depicted in figure 3.17, a first de-multiplexing stage (VC Demux) routes flits to their corresponding VC queues. Then, a shifter stage modifies the routing information as needed in order to be used in the next router. The flit is stored in a buffer stage waiting for consumption by one of the output ports. The notification from the input port to the good output port is performed only once at the beginning of packet in Signal Packet.

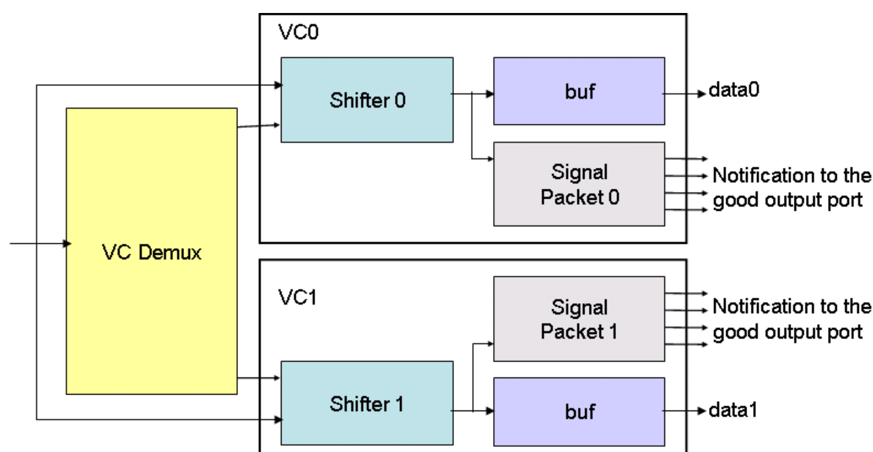


Figure 3.17: Micro-architecture of the input port

The output port is in charge of arbitrating and multiplexing the packets incoming from all input ports. This step begins when receiving the header flit marked with BOP. The path is kept open until receiving the tail flit with the EOP. Firstly, the output port achieves arbitration between directions inside each VC as shown in figure 3.18. After that, it performs arbitration only between VCs. Direction Arbiter performs arbitration of the new packet requests from the input ports, and generates a single command token to the Direction Switch. This token will be consumed only at the end of packet. Finally, the VC Arbiter arbitrates at flit level between the two VCs, and commands the VC Switch.

Given that the synchronous processing units plugged into the NoC may have a significant area, the inter-router links may have an important length that would impact the NoC performance. In order to overcome this issue, asynchronous pipelining is added to the NoC links. Compared to classical inverter/buffer wire buffering, this approach allows reducing the cycle time of the longest paths without adding a latency penalty.

The synchronous IP cores are connected to the routers by means of a GALS interface, which deals with: (a) re-synchronizing the asynchronous NoC protocol

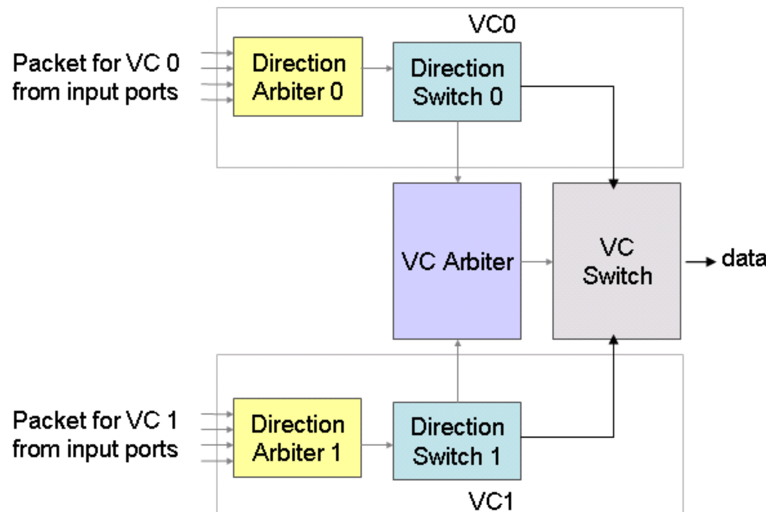


Figure 3.18: Micro-architecture of the output port

with the synchronous domain; (b) generating a local clock with a programmable frequency.

4 Area and power evaluation of the hierarchical 3D router

Table 3.2 depicts the area and power results for the 7x7, 5x5 and 4x4 routers. The 5x5 router has been deployed and fabricated using the STMicroelectronics CMOS 65 nm LP technology in a chip dedicated to flexible baseband processing for 3G/4G wireless telecommunication applications. Data of the 5x5 router are extracted from this chip [51].

Data of the 7x7 and the 4x4 routers are extrapolated based on our knowledge of the NoC router used in this work. This extrapolation is performed as follows. The nxn router includes a nxn crossbar, n input controllers and n output controllers. First, we estimate the area and power of the input controller, the output controller and the 5x5 crossbar. The crossbar represents about 45% of total 5x5 router power and area. The area and power of an input or an output controller do not vary with the Input/Output ports'number. According to [41], crossbar power and area are proportional to the square of Input/Output ports' number. Therefore, the area of the whole nxn router may be estimated by the following formula 3.1:

$$n \times \alpha + n^2 \times \beta = A_n \tag{3.1}$$

where: α is the area of the IO port, $n^2 \times \beta$ is the area of the nxn crossbar, and A_n is the total area of the nxn router. Knowing the area of the 5x5 router and the 5x5 crossbar, we can deduce α and β. Using the same formula (1), it would be possible to estimate the areas of the 4x4 and 7x7 routers. The same approach is applied to estimate powers of the 4x4 and 7x7 routers.

Based on data in table 3.2, the hierarchical router has almost the same area and power as the 3D mesh router.

Table 3.2: Area and power results for the 4x4, 5x5 and 7x7 routers in 65nm technology

Router type	Area (mm^2)	Power (mW)
4x4	0.12	8.66
5x5	0.17	11.9
7x7	0.28	19.65

5 Performance of the hierarchical 3D NoC

5.1 Simulation platform

In order to model and simulate the hierarchical 3D NoC, a class representing the asynchronous router is developed using SystemC-TLM language. The main purpose of the Transaction-Level-Modeling (TLM) in SystemC is to speed-up simulation. To do so, the main idea consists in modeling system transactions at high level of abstraction. Therefore, the proposed protocol is modeled at flit level.

A typical transaction can be either of data or accept type, and consists of many symbolic C++ fields:

```
t_access access; // transfer type: ACCEPT or DATA
int channel; // virtual channel number
t_bit accept; // for accept transfer: accept value
t_bit bop; // to indicate begin-of-packet
t_bit eop; // to indicate end-of-packet
t_uint8 srcid; // header: src id
t_uint8 dstid; // header: dest id
t_aNOC_dir *path; // header: path to destination
t_uint16 control; // header: message control
t_uint32 data; // packet data value
```

The router model is fully event-driven. It consists in function which stores incoming transactions (for data transfers) or incoming accept values (for accept transfers), and of a unique SystemC thread process which wakes-up with incoming transfers. It can be written as a simple C++ function:

```
while(true) {
  wait(node_evt); // wait incoming event
  arbitrate();
  wait(node_wait_state, SC_NS); // wait some time to model router delay
  for (int i=0;i<nb_input;i++)
    for (int j=0;j<nb_output;j++)
      for (int k=0;k<nb_channel;k++)
        if arbitrated(i,j,k) {
        node_out[j].write_data(input_data[i][k]);
        input_valid[i][k]=false;
        node_in[i].write_accept(k,accepted);
        output_accept[j][k]=false;
```

```
    }
}
```

The router model waits for incoming transactions, arbitrates according to each node output state (virtual channel policy), waits some time (to model node real transfer), and with a simple for-loop performs flit transfers on each arbitrated node output (write data on output port, write accept on input port). Since this model is generic in terms of number of input/output ports, it could be easily used to simulate all the 4x4, 5x5 and 7x7 routers. Such a functional TLM model of the router is extremely fast in simulation. Besides, it is accurate enough to get precise system performance evaluation. The delay model for the nxn router is the sum of the input port delay, the nxn crossbar delay and the output port delay:

$$D_{n \times nRouter} = D_{InputPort} + D_{n \times nCrossbar} + D_{OutputPort} \tag{3.2}$$

$D_{InputPort}$, $D_{OutputPort}$ and delay of the 5x5 crossbar are estimated from the 5x5 router (after place and route). Knowing the micro-architecture of the 4x4, 5x5 and 7x7 crossbars, and the multiplexer delay in 65nm technology, we were able to estimate delays of the 4x4 and the 7x7 crossbars. Using formula 3.2, we calculate delays of the 4x4 and 7x7 routers (see table 3.3).

Table 3.3: Delay results for the 4x4, 5x5 and 7x7 routers in 65nm technology

Router type	Delay (ns)
4x4	2.185
5x5	2.3
7x7	2.5

In this work, we use a deterministic source routing. For a 3-Dimension NoC, each resource (or processing element) can be identified by its coordinate (X, Y, Z). Let (Xs, Ys, Zs) and (Xd, Yd, Zd) be the coordinates of source and destination resources respectively. N, S, E, W, U, D and R are characters indicating respectively north, south, east, west, up, down, and resource directions. P is a character chain indicating the path to target. concat() means the concatenation operation. (n)U means repeating n times the character U. For the mesh NoC, we use the following ZYX routing scheme:

```
if (Zd>Zs)
  P= (Zd-Zs)U;
else if (Zd<Zs)
  P= (Zs-Zd)D ;
else if (Zd=Zs) {}

if (Yd>Ys)
  P= concat (P ,(Yd-Ys)E) ;
else if (Yd<Ys)
  P= concat (P ,(Ys-Yd)W) ;
else if (Yd=Ys) {}
```

```
if (Xd>Xs)
  P= concat(P ,(Xd-Xs)S , R) ;
else if (Xd<Xs)
  P= concat(P ,(Xs-Xd)N , R) ;
else if (Xd=Xs)
  P= concat(P , R);

return P;
```

For the hierarchical NoC, we consider 2 additional directions: a first direction from the 5x5 to the 4x4 router (referred as to_4x4) and a second direction from the 4x4 to the 5x5 router (referred as to_5x5). The adapted ZYX algorithm used to calculate the path to destination in the hierarchical NoC is as follows:

```
if (Zd>Zs)
  P= (Zd-Zs)U;
else if (Zd<Zs)
  P= (Zs-Zd)D ;
else if (Zd=Zs) {}

if (Yd>Ys)
  P= concat(P , to_5x5, (Yd-Ys)N) ;
else if (Yd<Ys)
  P= concat(P , to_5x5, (Ys-Yd)S) ;
else if (Yd=Ys){
  if (Xd=Xs){}
  else
    P = concat(P , to_5x5);
}

if (Xd>Xs)
  P= concat (P , (Xd-Xs)E , to_4x4 , R);
else if (Xd<Xs)
  P= concat (P ,(Xs-Xd)W , to_4x4 , R);
else if (Xd=Xs){
  if (Yd=Ys)
    P= concat (P, R);
  else
    P = concat(P , to_4x4, R);
}

return P;
```

To assess the performance of the proposed NoC architecture, simulations with different traffic patterns should be carried out. According to [52], traffic models may be classified as either synthetic or realistic. Synthetic traffic patterns are abstract models of message passing in NoCs whereas realistic traffic patterns are traces of real applications running on NoCs. Contrary to realistic traffic which is representative of a more specific class of applications, synthetic traffic should cover a broad class of

applications running on the NoCs. For this reason, we opt here for synthetic traffic patterns to cover a wide range of applications. In this work, we use uniform random and transpose traffic profiles:

(i) Uniform traffic model is a standard benchmark used in network routing studies. Packets are sent to a destination chosen at uniform random. All routers receive approximately the same number of packets. This model can be considered as the traffic model for of well-balanced shared memory computations.

(ii) Transpose traffic can model traces of applications related to numerical computations. Packets originating from router (x, y, z) have as destination the router (X - z, Y- y, Z - z), where X, Y, Z are the dimensions of the NoC.

Since the proposed NoC is intended to be used in large NoCs, the test-bench used for simulation consists of 256 routers (32 routers x 8 layers) connected to 256 traffic generators. Traffic generators inject packets in the network with random activation of virtual channels and random data values. Packet length (number of flits per packet) is set to 4.

5.2 Throughput

Injection load refers to the rate at which the functional IP blocks are injecting data into the network. In our case, the random data generators wait some time $WriteTime$ between the generations of two successive packets. We vary this $WriteTime$ in order to vary the injection load. Thus, injection load is calculated according to formula 3.3:

$$IL = \frac{GeneratedFlits}{WriteTime} \tag{3.3}$$

where $GeneratedFlits$ is the number of flits generated by each random data generators.

Throughput means the rate that message traffic can be sent across the network. It could be calculated using formula 3.4:

$$Th = \frac{TotalPacketCompleted \times PacketLength}{NumberOfIPs \times TotalTime} \tag{3.4}$$

where $TotalPacketCompleted$ is the number of messages that successfully reach their target IPs, $PacketLength$ refers to the number of flits per packet, $NumberOfIPs$ is the number of IP blocks implicated in the communication, and $TotalTime$ is the time (in clock cycles) that elapses between the first message sending and the last message reception [53].

Figure 3.19 depicts the variation of the throughput with the injection load for two different traffics: uniform and transpose. Since the hierarchical NoC is asynchronous, the cycle is replaced by the time unit (ns) in the presented results. According to figure 3.19, the saturation value of the proposed NoC is superior by either 15% (when using uniform traffic) or 25% (when using transpose traffic) compared to the classic 3D mesh. Thus, we can conclude that the hierarchical router outperforms the classic 3D mesh in term of throughput.

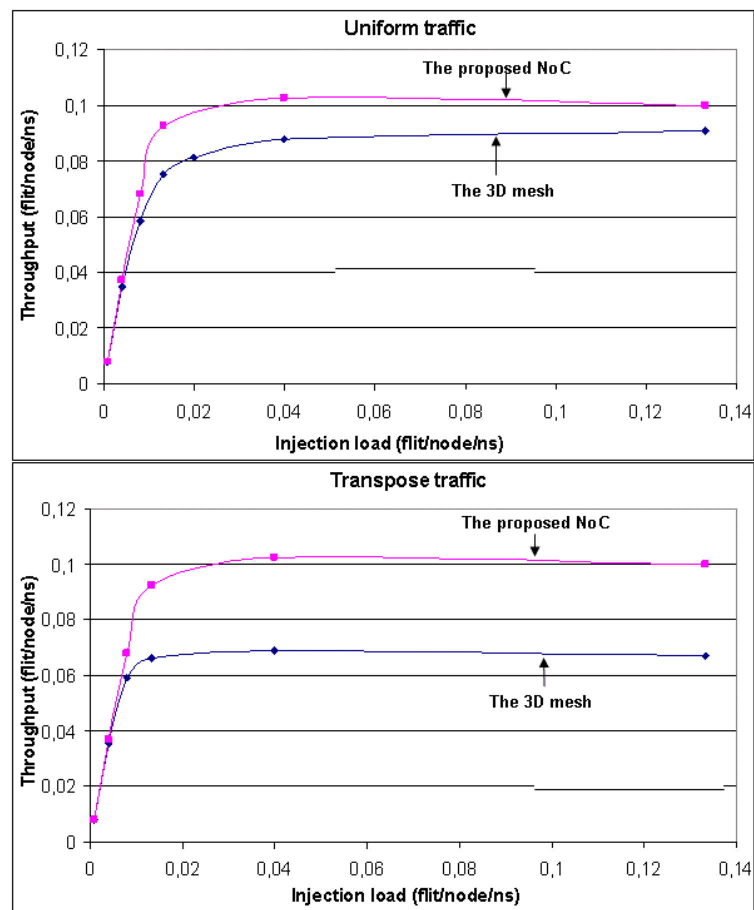


Figure 3.19: Throughput variation under different synthetic traffic patterns

5.3 Latency

Average latency is defined as the average time (in clock cycles) that elapses between the message header injection into the network at the source router and the tail flit reception at the destination router. It is given by the formula 3.5:

$$L = \frac{\sum_{i=0}^{N} L_i}{N} \tag{3.5}$$

with N being the total number of packets reaching their destination IPs and Li is the latency of each packet i [53].

Figure 3.20 shows the variation of the average network latency according to the throughput variation for the uniform and transposes traffics. For the two traffic patterns, the proposed NoC outperforms the classic 3D mesh in term of average latency by more than 25% when using transpose traffic and more than 15% when using uniform traffic. Thus, we can conclude that the hierarchical router outperforms the classic 3D mesh in term of latency.

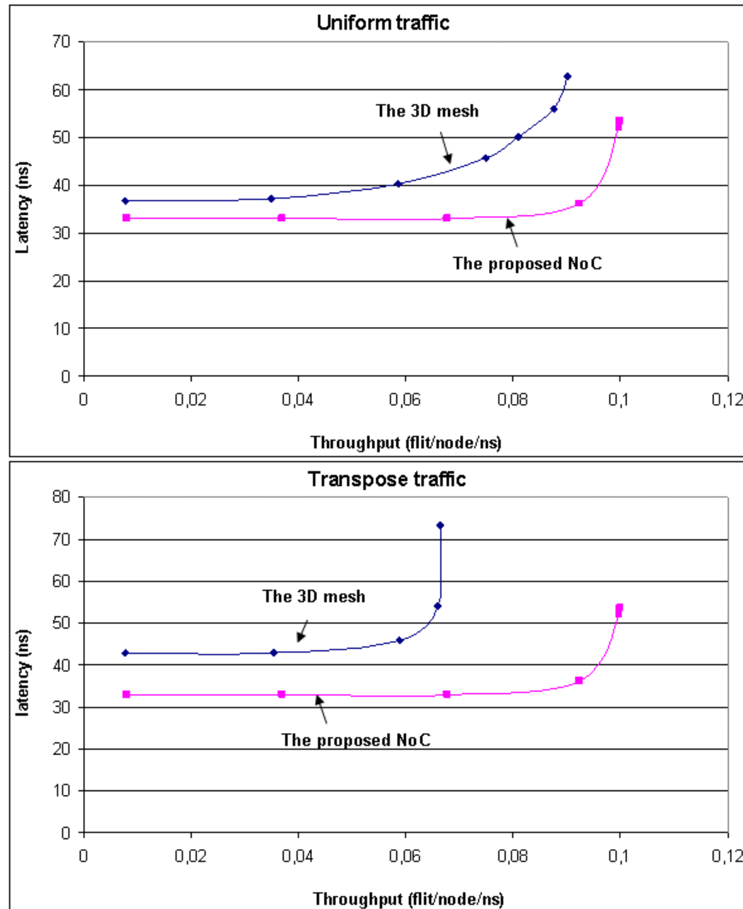


Figure 3.20: Latency variation under different synthetic traffic patterns

5.4 Result analysis

This section presents an explanation to the advantages of the hierarchical router over the classic 3D mesh router. We perform an analytical comparison of the hierarchical

router against classic 3D mesh NoC in terms of intrinsic latency. This analytical comparison is purely intrinsic and does not take NoC traffic contentions into account. Indeed, under the same traffic type, the only factor that affects global latency is the intrinsic latency of each router. Let:
-n be the number of routers crossed by a flit,
-v be the number of horizontal routers including the router used to access to the targeted layer (the layer that contains the destination IP),
-h be the number of vertical routers without including the router used to access to the targeted layer,
-$L1$ and $L2$ be the intrinsic latencies (without contentions) of n 7x7 routers and n hierarchical routers respectively,
-$d7$, $d5$ and $d4$ be the intrinsic latencies of the 7x7, 5x5 and 4x4 routers respectively.

Since a 7x7 router has more ports than a 5x5 router, its intrinsic latency exceeds the intrinsic latency of the 4x4 or the 5x5 router. We can assume that:
$d7 = (1 + y)d$,
$d5 = (1 + x)d$,
$d4 = d$,
$0 < x < y < 1$.

If traffic is a purely vertical, we obtain:
$n = v$;
$L1 = vd7 = v(1 + y)d$;
$L2 = vd4 = vd$;
$L1 - L2 = vyd > 0$.
We can conclude that the hierarchical NoC outperform the 3D mesh in term of latency when the traffic is a purely vertical.
If traffic is a mixture of vertical and horizontal communication, we obtain:
$n = h + v$;
$L1 = (h + v)d7 = (h + v)(1 + y)d$;
$L2 = vd4 + d4 + hd5 + d4 = (v + 2)d + h(1 + x)d$;
$L1 - L2 = [y(h + v) - 2 - xh]d$.

We consider now that n is the average number of hops within traffic. Let P be the percentage of the horizontal traffic. P is always non-zero since traffic is not purely vertical. We obtain:
$0 < P < 1 \Longrightarrow y > xP$;
$h = Pn$;
$L1 - L2 = [n(y - xP) - 2]d$;
So that the hierarchical router is beneficial in term of latency, we should have:

$$L1 - L2 > 0 \Longrightarrow n > \frac{2}{y - xP}$$

.

When the average number of hops exceeds a certain value, the hierarchical approach is beneficial in term of intrinsic latency. This value depends on the percentage of the horizontal communication (which depends on traffic type).

The previous analytical model can be intuitively clarified as follows considering n the number of routers crossed by a flit. In the case of purely vertical communication,

a flit has to cross n 4x4 routers instead of n 7x7 routers in the classic 3D mesh. Since the latency of the 7x7 router is superior to the latency of the 4x4 router, the proposed NoC will be advantageous.
In the case of purely horizontal communication, a flit in the 3D mesh NoC would cross n 7x7 routers. In the hierarchical NoC, it would cross a 4x4 router, n 5x5 routers and finally a 4x4 router. When n (the number of crossed routers) increases, the latencies of the two 4x4 routers will be compensated since the latency of the 7x7 router is superior to the latency of the 5x5 router. When n exceeds a certain value, the proposed hierarchical NoC will be more efficient than the 3D mesh. In the case of mixed (both horizontal and vertical) communication, the same analysis is applicable.

We can conclude that when the number of crossed routers is superior to a certain value $\frac{2}{y-xP}$, the proposed hierarchical NoC will be more advantageous than the 3D mesh. That's why the proposed NoC is beneficial in the case of large NoCs, since the average number of crossed routers is large enough.

6 Conclusion

In this chapter, a new hierarchical router for NoC-based 3D MPSoCs is presented and evaluated. This router is fully implemented in asynchronous logic in order to allow low latency transfer. This router is hierarchical as it is composed of 2 totally decoupled modules: one for intra-layer communication and one for inter-layer communication.

A comparison of the hierarchical 3D NoC against 3D mesh NoC shows that the hierarchical has almost the same area and power compared to 3D mesh. A fast SystemC- TLM simulator is used to evaluate the performance of the hierarchical router in term of throughput and latency. Simulation results that the proposed hierarchical router can outperform the 3D mesh by more than 25% in terms of throughput and latency using transpose traffic. The proposed NoC architecture will be implemented on a real 3D IC demonstrator. This allows to measure performance and to validate simulation results.

The proposed 3D NoC architecture is useful only for macro-block partitioned 3D circuits. Performance enhancement in this type of 3D ICs is limited to interconnect. Therefore, overall performance and power gains (due to 3D technology) are not so-exciting, especially in the case of processing-dominated applications such as telecommunications and image processing. The use of 3D integration (at this level of partitioning) could not be well-justified based on performance benefits only, but others motivations, especially economical ones, should be mentioned. Next chapter focuses on cost-aware 3D architectures. It investigates the cost-effectiveness of the 3D same-die stacking approach, with a real case study on 4G telecom applications.

Chapter 4

A reconfigurable and stackable circuit for 4G telecom applications

Today's complex SoC designers are facing several problems that limit the economic benefits of advanced technology nodes. In addition to the prohibitive cost of masks (which has already exceeded 1 million euro according to the ITRS), wafer fabrication is becoming more and more expensive mainly due to huge circuit size that reduces manufacturing yield. A good solution to develop economically competitive products is to reuse masks to address a wide range of systems and to fabricate small-sized circuits to increase yield. To do so, our proposal is to design a modular circuit that could be stacked using 3D integration technologies in order to build 3D systems with processing performance adapted to several application requirements. This type of circuits is referred to as 3D same-die stacked architectures in this thesis.

In this work, we focus on modular architectures for 4G telecom applications, which are the latest standard in the mobile network technology with important performance requirements. We propose a reconfigurable circuit that meets the SISO (Single Input Single Output) mode of transmission (1 antenna) in a stand-alone. By stacking multiple instances of this same circuit, it would be possible to boost overall system performance and address several MIMO (Multiple Input Multiple Output) modes.

The reminder of this chapter is organized as follows. Section 1 presents the different algorithms of the LTE standard and their computational requirements. Section 2 presents the reconfigurable and stackable circuit, and proves its adequacy to meet the performance needs of the LTE standard. Section 3 explains how the hardware components of the stackable circuit are reconfigured and programmed. Some technological considerations of 3D integration are described in section 4. A case study of the 4x2 LTE mode of transmission with performance and power evaluation is presented in section 5. Finally, section 6 presents a cost analysis of the proposed circuit.

1 Implementation of 4G terminals: algorithms and requirements

The 4G standard is the latest standard in the mobile network technology. It is known also as 3GPP LTE: 3rd Generation Partnership Project Long Term Evolution. The 3GPP is collaboration between groups of telecommunications associations that aims to make a globally applicable third-generation (3G) mobile phone system specification within the scope of the International Mobile Telecommunications-2000 project of the International Telecommunication Union (ITU). Long Term Evolution (LTE) is a project of the 3GPP that produces the latest standard in the mobile network technology tree in order to move forward from the cellular 3G services to the 4G services. The main objectives for 3GPP LTE (or 4G) are to increase downlink and uplink peak data rates (100Mbps for DL with 20MHz, 50Mbps for UL with 20MHz), to improve spectral efficiency (5bps/Hz for DL and 2.5bps/Hz for UL), to reduce latency, to improve bandwidth scalability, and to establish a standard's based interface that can support a multitude of user types [54, 55].

The LTE physical layer is in charge of transmitting data and control information between an LTE base station and the user equipment (a mobile phone typically). The LTE physical layer is based on Orthogonal Frequency Division Multiplexing scheme OFDM to meet the targets of high data rate and improved spectral efficiency. OFDM makes use of a large number of closely-spaced orthogonal sub-carriers to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying). The modulation schemes supported in the downlink and uplink are QPSK, 16QAM and 64QAM. In order to improve communication robustness and throughput, the LTE physical layer supports several Multiple Input Multiple Output (MIMO is the use of multiple antennas at both the transmitter and receiver) options with 1, 2 or 4 antennas [56].

In this section, we focus only on the baseband processing of the downlink part (reception chain) and omit all other components of LTE user equipment UE such as radio-frequency and analogue functions, higher layer protocols and multimedia processing. Figure 4.1 depicts a functional block diagram of the internal data flows of the downlink part within an LTE UE with four receive (Rx) antennas [57]. First, the RF signal is received by the receiver antennas, converted to an electrical quantity, and digitized by an analogue to digital converter (ADC). Then, the baseband processor receives the digitized signal as complex samples and performs OFDM demodulation, channel estimation and finally MIMO decoding [58]. In the following subsections, we will analyse each of these processing steps in order to determine their computational efforts.

1.1 OFDM demodulation

This step performs a fast Fourier transform FFT in order to transform the received signal into frequency domain. The FFT complexity depends on the bandwidth (the number of sub-carriers) that is used to carry data. Therefore, for a given bandwidth B (number of sub-carriers), the computational complexity is the same

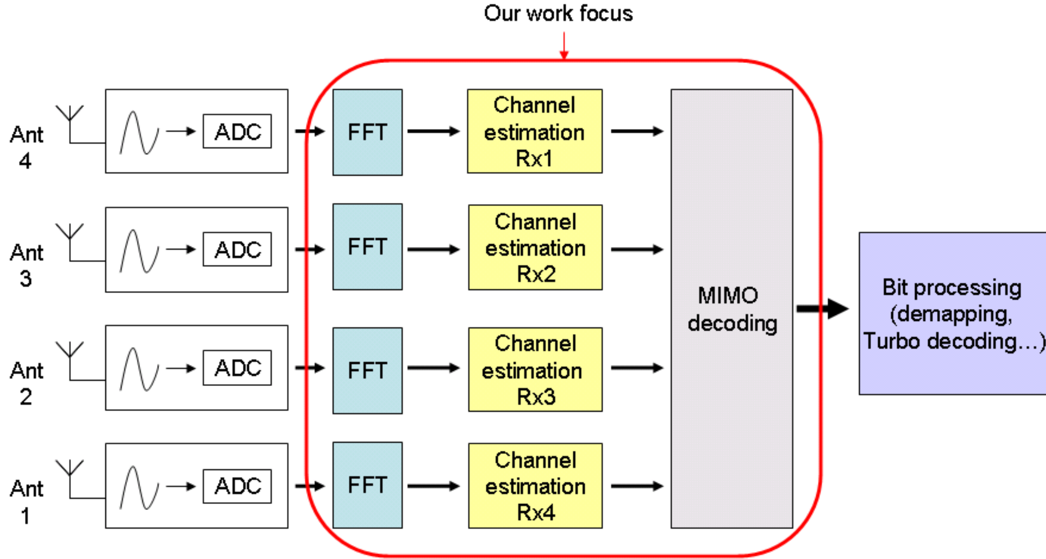


Figure 4.1: A functional block diagram of an LTE UE reception chain with 4 receive (Rx) antennas

for all LTE modes of transmission (or antenna configuration: number of transmission antenna/number of reception antenna). It requires B^2 complex multiplication for each reception antenna. Given the large bandwidth used in the LTE standard, the FFT is usually performed by specialized blocs within the LTE terminals.

1.2 Channel estimation

A MIMO-OFDM system with N_T TX (transmission) antennas and N_R RX (reception) antennas is depicted in figure 4.2. At a discrete transmission time n, a stream of binary bits is coded into N_T OFDM symbol blocks. The signal on k^{th} subcarrier is denoted by $x_i[n,k]$, where i=1,...,N_T, k=0,..., K-1 (K is the total number of subcarriers). The received signal at RX antenna j is given by equation 4.1 [59]:

$$y_j[n,k] = \sum_{i=1}^{N_T} x_i[n,k]h_{ij}[n,k] + w_j[n,k] \tag{4.1}$$

where $h_{ij}[n,k]$ is the channel transfer function between antennas i and j, $w_j[n,k]$ is the additive Gaussian noise with zero mean.

For simplicity, the time index n and carrier number k will be henceforth omitted. Equation 4.1 becomes then 4.2:

$$X = [x_1, x_2, ..., x_{N_T}]$$

$$h_j = [h_{1j}, h_{2j}, ..., h_{N_T j}]$$

$$y_j = Xh_j^T + w_j \tag{4.2}$$

X may be a data symbol X_d or a pilot symbol X_p. Pilot symbols with predetermined values are used for channel estimation (determining h_j using equation 4.2). Channel estimation may be performed using several techniques. In this work, we use

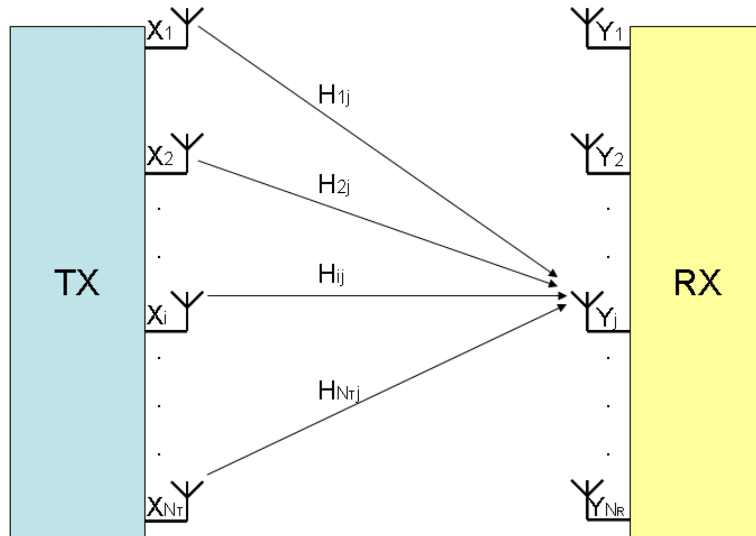


Figure 4.2: A MIMO-OFDM system with N_T TX antennas and N_R RX antennas

a modified version of the Linear Minimum-Mean-Square-Error (LMMSE) channel estimation. This algorithm requires the knowledge of the second order statistics of the channel and the noise. It provides good results, but it requires a high computational complexity. Using the MMSE, the channel coefficients matrix for subcarriers on which pilot symbols are located (for each antenna), is given by equation 4.3 [59]:

$$h_p^{MMSE} = R_{hh}(R_{hh} + \frac{\beta}{SNR}I)^{-1}(X_p^H X_p)^{-1}X_p^H y_p \tag{4.3}$$

where X_p^H is the complex conjugate transpose of X_p, R_{hh} is the autocorrelation matrix of the channel at the pilot symbols position, β is a constant depending on the training signal's constellation, and SNR is the signal to noise ratio. Since the pilot data X_p are known, $(X_p^H X_p)^{-1}$ in 4.3 can be calculated in advance. Therefore, only 1 matrix inversion is required in 4.3. The remaining channel coefficients for data symbols have to be obtained by linear interpolation.

In order to simplify the evaluation of MMSE complexity, we will omit arithmetic operations (additions and multiplications) that are performed to obtain R_{hh}, β and SNR. Table 4.1 depicts the number of complex additions and complex multiplications that are required to perform channel estimation for subcarriers on which pilot symbols are located (using the MMSE algorithm).

1.3 MIMO decoding

As seen in the previous subsection, the received signal at RX antenna j is given by equation 4.2. Let consider the following matrix notations:

$$H = [h_1, h_2, ..., h_{N_R}]^T$$

$$Y = [y_1, y_2, ..., y_{N_R}]^T$$

$$W = [w_1, w_2, ..., w_{N_R}]^T$$

The received signals at all RX antennas are given by equation 4.4:

$$Y = HX + W \tag{4.4}$$

H is an $N_R \times N_T$ matrix. MIMO decoding is determining X knowing H. To do so in this work, we use the MMSE algorithm. X is obtained by equation 4.5 [60–62]:

$$X_{est} = H^H(HH^H + \sigma^2 I)^{-1} Y \tag{4.5}$$

where H^H is the conjugate transpose of H, and σ is a constant depending on the SNR.

Table 4.1 depicts the number of complex additions and complex multiplications that are required to perform channel estimation and MIMO decoding (using the MMSE algorithm). $Inv(N \times N)$ indicates the complexity of matrix inversion according to the matrix size $N \times N$.

Table 4.1: The theoretical complexity of the LTE terminal with NT TX antennas and NR RX antennas

Step	(+)	(×)	matrix inversion
Channel estimation	$N_R(3N_T^2 - 2N_T)$	$N_R(3N_T^2 + N_T)$	$N_R Inv(N_T \times N_T)$
MIMO decoding	$N_T N_R^2 + N_R N_T - N_T$	$N_T N_R + N_R^2 + N_T N_R^2$	$1 Inv(N_R \times N_R)$

For the LTE standard, N_T and N_R (the number of TX antennas and RX antennas respectively) may be 1, 2 or 4. Table 4.2 depicts the number of additions and multiplications required to perform the inversion of 2x2 and 4x4 matrixs.

Table 4.2: The theoretical complexity of the inversion of 2x2 and 4x4 matrixs

Matrix	(+)	(×)
2x2	1	6
4x4	103	280

Finally, table 4.3 depicts the number of additions and multiplications required by channel estimation and MIMO decoding for different LTE modes ($N_T \times N_R$): 4x4, 4x2, 2x2, 2x1 and 1x1. These amounts of arithmetic operations correspond to the processing of only 1 subcarrier. Therefore, total number of complex additions and multiplications is obtained by multiplying the data of table 4.3 by the total number of subcarriers.

Table 4.3: The theoretical complexity of channel estimation and MIMO decoding for different LTE modes

Transmission mode	Channel estimation		MIMO decoding	
	(+)	(×)	(+)	(×)
4x4	572	1,328	179	376
4x2	286	664	21	34
2x2	18	40	11	22
2x1	9	20	3	11
1x1	0	1	0	1

When it comes to real implementation, channel estimation and MIMO decoding algorithms are simplified (especially matrix inversion) in order to reduce computational efforts and power consumption of the LTE terminals. Therefore, the theoretic

complexities presented below do not correspond to the actual requirements of an LTE user equipment. Nevertheless, in this work we consider these theoretic complexities in order to have an upper limit of the performance requirements of LTE terminals.

2 A stackable circuit for LTE applications

After analyzing the performance requirements on the LTE terminals, in this section, we present a reconfigurable NoC-based circuit for LTE applications. When used alone, the proposed circuit (henceforth called basic circuit) can meet the requirements of the SISO transmission mode. By stacking multiple instances of this basic circuit and performing some software reconfigurations, it will be possible to boost system performance and address several MIMO modes (figure 4.3). This section presents the hardware components of the basic circuit such as processing units and the NoC-based communication structure, and provides synthesis results in 65nm technology.

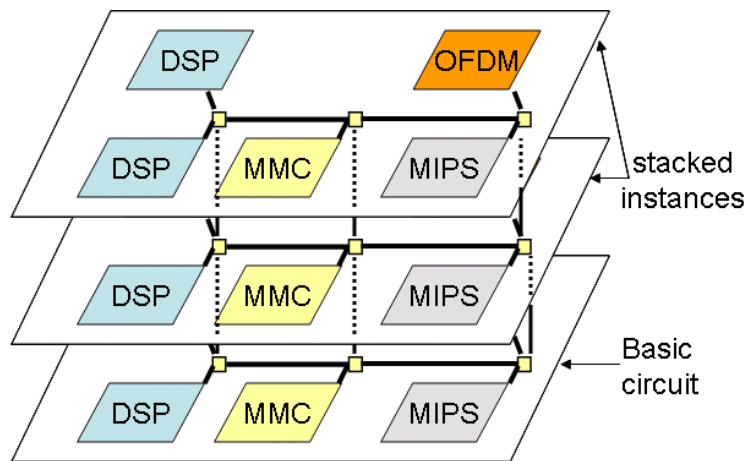


Figure 4.3: 3D reconfigurable circuit obtained by stacking multiple instances of a same basic circuit

2.1 Processing units

To be able to satisfy the needs of several telecom applications, the basic circuit has to support data manipulation and data processing at the same time. To do so, three reconfigurable units are designed.

Smart Memory Engine (SME)

The SME unit is entirely developed by the LISAN team [63]. It is a Micro-programmable Memory Controller (MMC) designed to perform data synchronization and distribution in dataflow systems. The SME allows separating data synchronization from data processing, and thus reducing the complexity of processing units and helping their reuse. A subset of the C programming language and a dedicated compiler are used for flow programming in the SME.

An SME is composed of several logical buffers like the one presented in figure 4.4. Each one has its own controlling hardware and uses a configurable section of the MMC RAM as a storage area. The use of a shared memory for all buffers improves flexibility and drastically reduces the global needed memory, because each buffer may be sized, or resized, precisely according to applicative needs. Each logical buffer is mapped into a physical memory area defined by a base pointer and a top pointer, which determine the maximum amount of data that the buffer may store at a given time. Two controllers, named Read Process and Write Process, define the valid set of data by controlling two increasing pointers. An increment of the write pointer (WP) means that new data have been stored in the buffer, whereas a move of the read pointer (RP) means that some data are no more available in the buffer.

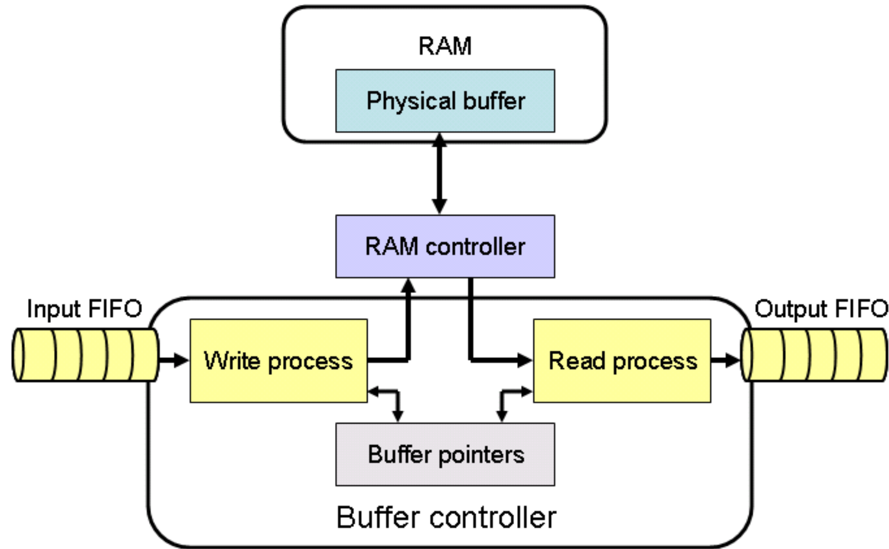


Figure 4.4: SME buffer functional model

Mephisto digital signal prosessor

The second unit used in the proposed basic circuit is a digital signal processor (DSP) called MEPHISTO [64]. It is a high-performance reconfigurable core designed by the LISAN team to perform complex matrixs computation, useful for channel estimation, advanced MIMO coding/decoding... MEPHISTO is designed as a 32-bit data path Very-Long-Instruction-Word (VLIW) structure composed of a MAC (Multiplier/Accumulator) unit dedicated to complex arithmetic operations, a compare/select unit for branch operations, and a cordic/divider unit for special computations.

This DSP is composed of a control part and a data path (figure 4.5). The control part uses the instructions stored in the 1Kx64bits memory to command the data path and the address generators. The data path includes a MAC unit used to perform complex and scalar operations with 2 independent sub-parts (16b I, 16b Q). The MAC unit is designed as a 4-stage pipeline composed of many operators: i) four 16bx16b multipliers (32b result), ii) 2 partial adders and 2 pairs of 40b accumulators, iii) a conversion stage in 2-complement notation and iv) finally a programmable shift and saturation. For the whole operation, the MAC offers 3.2 scalar GOPS.

Two 1Kword single port RAM and a 64-word Register Array (RA) with 2 read and 1 write ports are accessible in parallel. Each RAM and RA port has its own Address Generator (AG) to schedule the operands. There are 2 shift registers (SR), one for bit sequences and another for address sequences. Finally, Compare/Select, 12b cordics and 32b inverter (1/x) operators are provided to allow more flexibility for specialized operations. The MEPHISTO inputs/outputs consist in FIFOs to receive/send data from/to the Network-on-Chip (NoC) through a Network Interface (NI).

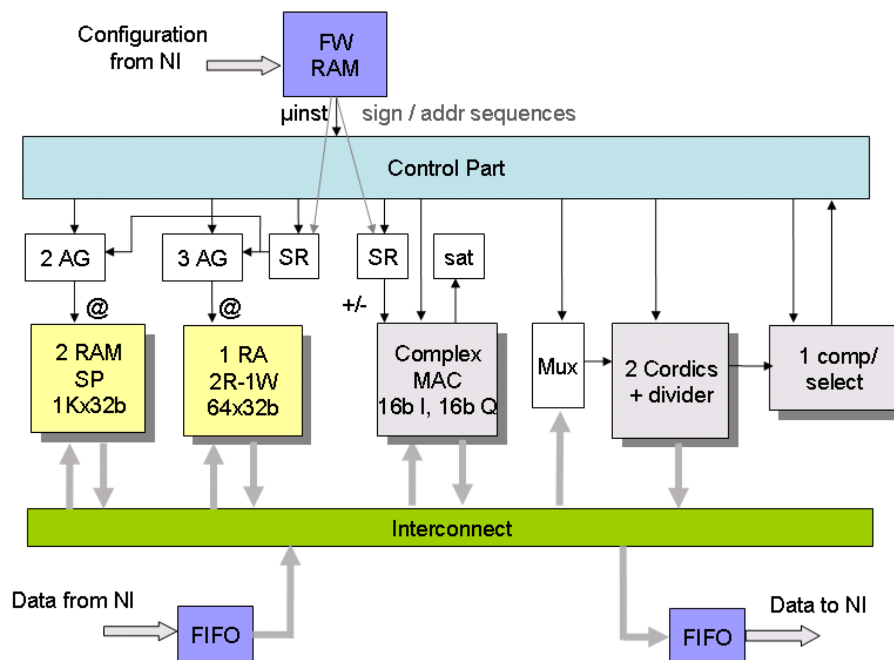


Figure 4.5: Architecture of the Mephisto DSP

OFDM core

The OFDM core is designed by the LISAN team to perform direct and inverse fast Fourier transform (FFT and IFFT). It also incorporates features to achieve a formatting of incoming OFDM symbols (framing i.e insertion of pilots and zeros) and a separation of outgoing pilot and data symbols (deframing). Therefore, this block can be used for both OFDM transmission (framing + IFFT + inserting guard interval) and reception OFDM (FFT + deframing).

To perform OFDM symbols processing, the OFDM core requires complete blocks (32 to 2048 words) to be present at the input. A double buffering (at the input and at the output) is used to avoid penalizing the computing time by the communication time. The 3 phases load / computing / unload are then clearly separated thanks to an execution pipeline (figure 4.6). Consequently, the processing of an OFDM symbol is spread over 3 configurable phases.

Figure 4.7 depicts the resulting basic circuit. It is composed of 1 OFDM core, 1 SME and 2 MEPHISTOs (to meet real time constraints) interconnected via 3 routers. Each unit is plugged on the NoC via a network interface (NI). The NI deals with packetization, depacketization and flow control using credits, handled by input/output communication controllers.

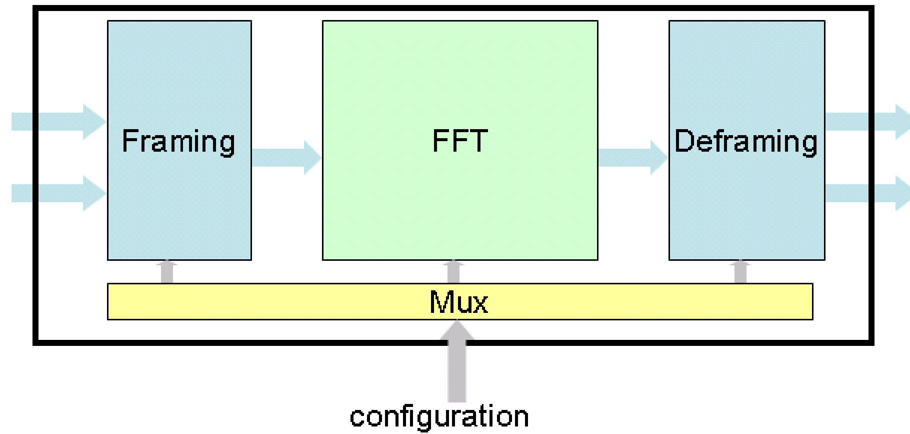


Figure 4.6: Architecture of the OFDM core

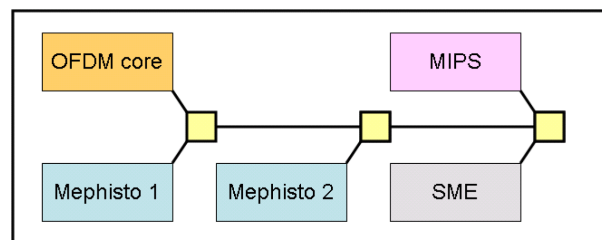


Figure 4.7: The basic circuit

2.2 MIPS-based semi-distributed control

The previously described units require an efficient control mechanism to deal with scheduling and configuration. In this work, we use a semi-distributed control for the whole basic circuit. This allows alleviating the load of the host processor [65]. In addition to the local configuration and communication controller performed by the NI, a global control is performed by a 32-bit MIPS processor, by means of direct addressing and interrupts mechanisms. The MIPS is chosen for its compactness. It is in charge of dynamic reconfigurations, real time scheduling and synchronizations. As depicted in figure 4.10, the MIPS processor has several extensions useful to interact and communicate with other basic circuit's components. These extensions include:

- an output extension managing the generation of data and configuration packets from the MIPS,
- an input extension allowing to read (to dump) data and configuration values from any of the circuit's units at the request of the MIPS,
- an interrupt controller in charge of handling interrupts generated in the NoC such as end-of-task notifications,
- and finally a local 16 KB RAM (32-bit word) used to store both instructions (the embedded control software), and data (configurations). An arbiter is used to allow the NoC to write the embedded control code in the MIPS's memory.

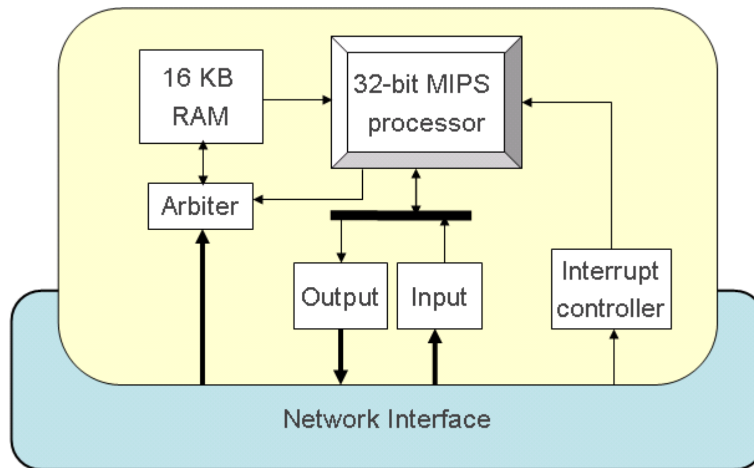


Figure 4.8: The MIPS-based global control unit

Unlike the local configuration and communication control performed by the NI, the global control is software-based to allow more flexibility. When multiple basic circuits are stacked (to build a 3D system), global control is distributed between all the MIPS processors of the resulting 3D stack. Each one is in charge of controlling the 4 components of its layer and communicates with other host processors to exchange information about scheduling. Compared to a centralized host processor approach, this approach allows distributing and then alleviating the global control load. Moreover, such a distributed approach improves scalability by avoiding control bottleneck problem when the number of processing cores increases.

The whole LTE application is composed of several tasks that must be executed in a well-defined succession. Each task is performed by 1 or more processing units, which are not necessarily on the same layer. So that a unit achieves properly its work, it requires some configurations and commands. A MIPS processor is in charge of providing these configurations and commands for all processing units located on the same layer as it. After receiving configurations and the task enable from the MIPS processor, a unit starts its processing. Once processing is finished, the unit informs the control processor by sending an interrupt message. The control processor may send then new configurations and commands for this unit to deal with another task. As said previously, tasks have to be executed in a specified succession. To guarantee this, MIPS control processors need to communicate together in order to exchange information about scheduling and synchronization of the processing units. This interaction is performed by means of software interruptions, which are sent via the NoC. Figure 4.9 depicts the interaction between MIPS processors and the processing units.

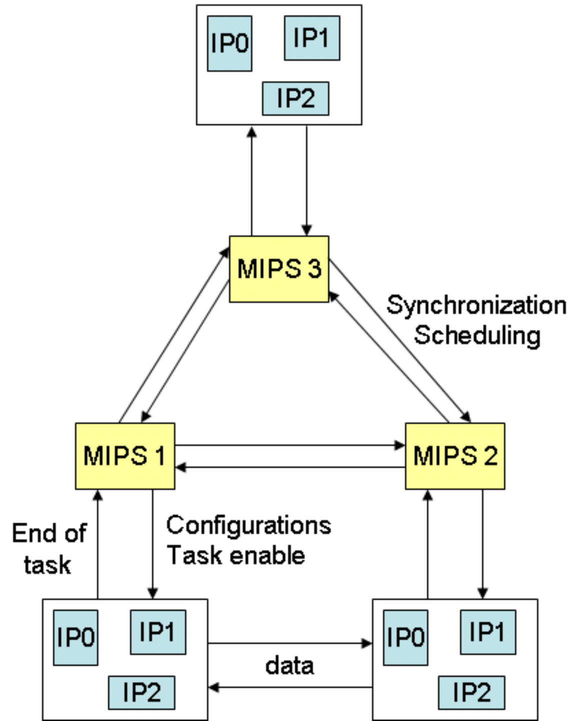


Figure 4.9: Diagram of tasks

2.3 The network interface

In order to avoid performance degradation due to NoC, the interface between the core and the NoC must be as simple as possible. Therefore, it would be judicious to separate configuration and data flow blocs. As depicted in figure 4.10, the network interface is composed of 3 blocs [66].

The first one is the input/output bloc, which is composed of the input and output ports. The input port is in charge of decoding the received packet and forwarding its content to data flow or configuration blocs. The output port is in charge of arbitrating between output packets coming from all the NI blocs, and injecting them in the NoC.

The second bloc is dedicated to data flow control. It can manage multiple input and output flows thanks to a credit mechanism that guarantee that no packet would be blocked in the NoC. The flow control bloc is composed of 2 sub-blocs: an input data manager (IDM) and an output data manager (ODM). The IDM of the destination unit sends credits through the network to inform the source unit that there are some available places in the NI buffer to receive data. The number of these available places is the same as the number of credits.

The third bloc is the communication and configuration controller (CCC) devoted to manage complex dynamic configuration flows (figure 4.11). The CCC makes use of a simple address/data interface to handle configurations for both the NI and the core. It makes use also of an execution interface to launch computations while guaranteeing a correct load of configurations. This interface includes an *exec* signal indicating a start of computation, a *slotid* signal indicating the configuration to be played, and *end of computation (eoc)* signal used by the CCC to start another configuration/execution step.

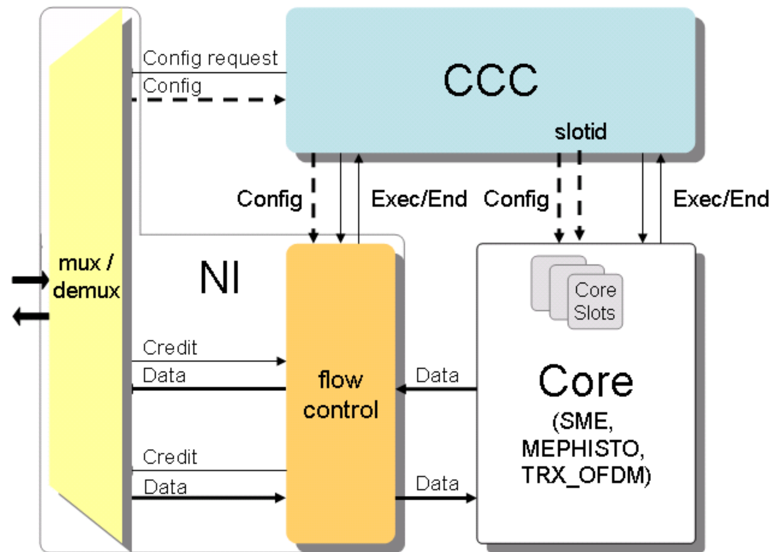


Figure 4.10: The network interface blocs

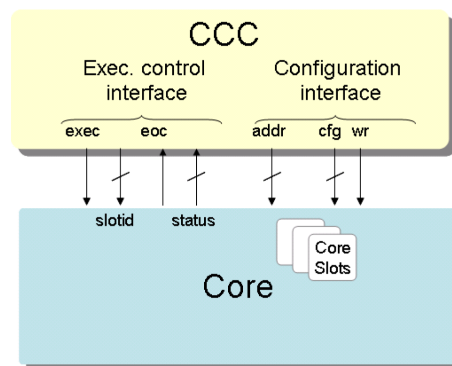


Figure 4.11: The communication and configuration controller

2.4 3D asynchronous mesh NoC

All the units of the basic circuit are interconnected via a NoC. This NoC is also used to connect all the components of a 3D system resulting from stacking 2 or more basic circuits. The basic circuit is designed as Globally Asynchronous Locally Synchronous (GALS) system. Processing units are synchronous (each one has its own clock frequency), while NoC routers are implemented in Quasi-Delay Insensitive asynchronous logic. The NI performs synchronization between the synchronous and asynchronous domains. The implementation details of this asynchronous router are explained in chapter 3. In this work, we use only 5x5 routers (5 Input ports x 5 output ports) to deal with all intra-layer and inter-layer communications as depicted in figure 5. The down ports of the routers located at the bottom layer are used to communicate with the external world.

2.5 Design results

The asynchronous router and all the previously described units were synthesised with 65nm low-power CMOS technology. All units' designs include the NI and test mechanisms like scan chains and memory BIST. TSVs' area overhead is a key challenge limiting the viability of 3D circuits. The asynchronous router needs 184 links at each port to communicate with its neighbors. Considering high-density TSV with $4\mu m$ diameter and $10\mu m$ pitch (figure 4.12), total TSVs' area would be $12800\mu m^2$, which corresponds to 6.5% of the router size.

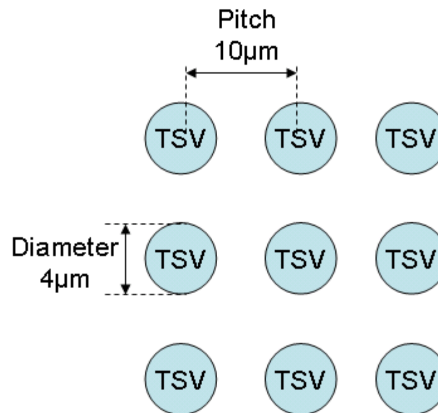


Figure 4.12: TSV characteristics

Table 4.4 depicts area results with the corresponding frequency of each synchronous core. The basic circuit has a total area of $3.6mm^2$. The TSVs' area overhead corresponds to 1.1% of the whole basic circuit size. We consider this overhead as negligible in this work. The MIPS host processor induces a 4.8% silicon impact ($<5\%$). The NoC infrastructure represents 17% of the whole circuit area.

2.6 Adequacy of the basic circuit to meet LTE requirements

After presenting the different components of the basic circuit, we have to prove that processing performance resulting from stacking several instances of this basic circuit is sufficient to satisfy the requirement of the different LTE modes. It is

Table 4.4: Synthesis results in 65nm technology

Bloc	Frequency (MHz)	Area (mm^2)
MIPS	300	0.175
MEPHISTO	400	0.455
SME	400	1.274
OFDM core	400	0.58
184 TSVs/router	-	0.0128
Router	-	0.20
Basic circuit	-	3.58

worthy noting here that all the previously presented processing units and NoC have been already integrated in 2D platform called MAGALI devoted to wireless telecom applications. MAGALI focuses mainly on 3GPP LTE standard and aims multi antennas schemes (MIMO). In order to implement the 4x2 mode of transmission, a subset of the MAGALI platform composed of 2 OFDM cores (1 for each reception antenna), 2 SME and 4 Mephistos were used.

Concerning OFDM demodulation, MAGALI platform allows having an OFDM core for each reception antenna in order to implement the 4x2 LTE mode. For the 4x4 mode, each reception antenna receives the same amount of data (coming from 4 transmission antennas) as in the 4x2 mode. For other LTE modes (2x2, 2x1 and 1x1), each reception antenna receives less data than the 4x2 mode. Consequently, 1 OFDM core would be sufficient to process data received by 1 antenna, whatever is the mode of transmission.

Besides, 4 Mephisto DSPs have been used to implement the 4x2 mode: 2 for MIMO decoding and 2 for channel estimation. The Mephisto's occupation ratio is 70% for MIMO decoding and 96% for channel estimation. Based on these results and by referring to theoretic complexities presented in table 4.3, we were able to deduce the number of Mephisto DSPs that are required by other LTE modes for both MIMO decoding and channel estimation (table 4.5).

Table 4.5: The theoretical complexity of channel estimation and MIMO decoding for different LTE modes

Transmission mode	Number of Mephisto DSPs		
	Channel estimation	MIMO decoding	Total
4x4	4	16	20
4x2	2	2	4
2x2	1	1	2
2x1	1	1	2
1x1	1	1	2

Moreover, some memory is required to store data that is obtained after each processing step (OFDM demodulation, channel estimation and MIMO decoding). For the 4x2 mode, 2 SMEs were sufficient. For the 4x4 mode, the amount of received data is twice as that of the 4x2 mode. Therefore, 4 SMEs at least will be required. For the 2x2 mode, the amount of received data is less than that of the 4x2 mode. 2 SMEs would be then sufficient. For other modes, only 1 SME is required.

Table 4.6 depicts the number of instances (of the basic circuit) required to address the different LTE modes.

Table 4.6: The numbers of instances (of the basic circuit) required to address the different LTE modes

Transmission mode	Number of instances
4x4	10
4x2	2
2x2	1
2x1	1
1x1	1

It is worth reminding here that these results are obtained based on theoretical complexities that are shown in table 4.3. In real implementation, algorithms are simplified in order to reduce computational requirements and power consumption. Therefore, results of tables 4.6 and 4.5 are actually the upper limits to the real required numbers of Mephisto DSPs and basic circuit instances.

3 Units' programming and platform control

As said previously, the processing units of the basic circuit are reconfigurable and programmable to be able to address several telecom applications using the same hardware. This section describes how these units are programmed and reconfigured to be adapted to different application requirements.

As depicted in figure 4.13, data manipulation in the SME is described using a subset of the C++ programming language. A dedicated compiler is used to obtain the binary code.

Computing tasks of the MEPHISTO are independently described and compiled into a binary code that is embedded in the instruction memory. A complete tool chain helps the programmer to transform a high level description into a binary code.

The local Configuration and Communication Controller (CCC) located in the NI handles the configuration data transfers, the storage of micro-programs for the input communications, output communications and the core processing, and the local scheduling of the NI and the core. The CCC is programmed using a configuration file.

The software-based global control allows more flexibility without impacting performance since the MIPS host processor is in charge of controlling only 4 processing units. The host processor is programmed using a standard C code and a GCC cross-compiler tool chain. Basically, the control code has to deal with scheduling and reconfigurations. An Application Programming Interface (API) is developed to abstract low-level details and could be used to program the MIPS. An example of control code is depicted in figure 4.14. This simple code describes communications mechanisms between MIPS processors as well as scheduling and configurations of processing units. As we can see in this example, hardware details are totally abstracted for the programmer.

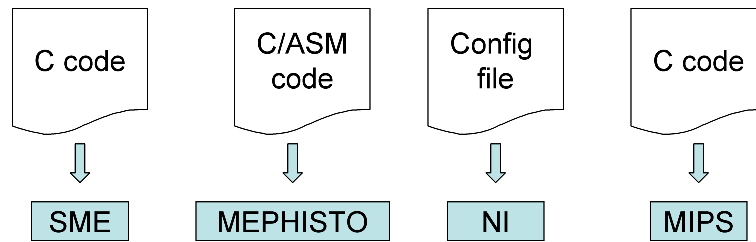


Figure 4.13: Programming of the processing units and the NI

```
// API definition
#include "control.h"

int main(void){

  uint32 it, core;

  it=0;

  // masking interruption
  mask_IT();

  // send enable tasks
  send_enable_task(0, 0, 0, 0, 1, PathTo_trx_ofdm_02);
  send_enable_task(0, 1, 1, 1, 1, PathTo_sme_03);
  send_enable_task(0, 0, 0, 0, 1, PathTo_mep_04);

  // waiting for end of task interruption
  while (it==0){
    //Automatic clock gating
    standby() ;

    if (read_reg_IT() == 2 || read_reg_IT() == 3){
      if (read_noc_it() == 65536){
        it=1;
      }
    }
  }

  //inter-MIPS control event
  send_it(PathTo_mips_15, 11, 0, 0);

  // send reconfigurations to units
  send_move_packet(11, PathTo_sme_13, core + 48 , 2, 0);
  write_packet_data (0x0);
  send_move_packet(11, PathTo_mep_04, 1279 , 2, 3);
  write_packet_data (0x2);
  ...

  return 0;
}
```

Figure 4.14: Example of the control code

4 Technological considerations

When using the macro-block partitioning granularity, each processor core is identical to its original 2D version and therefore has the same performance and power characteristics. Benefits of 3D stacking in terms of performance and power consumption are limited to vertical interconnections (TSVs). This section presents a TSV physical model that we use to evaluate signal propagation delay within a TSV.

In this work, we consider the 3D integration process presented by Cadix et al. [1] from CEA-LETI. Here after, we present a detailed description of this process flow. First, chemical and mechanical polishing (CMP) and cleaning are performed to prepare the surfaces of both top wafer and Si substrate. After that, wafer stacking is performed in a face-to-face approach by direct oxide bonding. The stack is then thinned down to $15\mu m$ thanks to a combination of grinding and chemical and mechanical polishing steps. A $1\mu m$ oxide layer is subsequently deposited to ensure the electrical insulation of future TSV. Next, high density cylindrical $4\mu m$-diameter TSVs are patterned by lithography (104 TSV/mm^2). After several intermediate steps (to ensure good isolation), TSVs are filled by copper. Further details about this process flow are provided in [1]. Figure 4.15 shows the TSV electrical model proposed by Cadix et al.

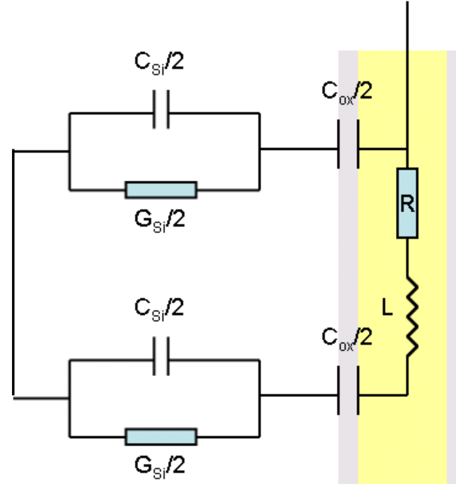


Figure 4.15: Physical model of a TSV as proposed by Cadix and al. [1]

A TSV capacitance is described by formula 4.6:

$$C_{TSV} = \frac{C_{ox}[G_{Si}^2 + C_{Si}w^2(C_{Si} + C_{ox})]}{G_{Si}^2 + w^2(C_{Si} + C_{ox})^2} \tag{4.6}$$

C_{ox} represents oxide capacitance between TSV and silicon due to isolation oxide. G_{Si} and C_{Si} are parasitic conductance and capacitance due to presence of lossy silicon substrate. TSV resistance is given by equation 4.7:

$$R_{TSV} = R_{Barrier} + \frac{\rho_{Cu} t_{Si}}{\pi(D_{TSV}\sigma - \sigma^2)} \tag{4.7}$$

t_{Si} is the TSV depth, ρ_{Cu} is the copper resistivity, D_{TSV} is the TSV diameter and σ is the frequency dependent skin depth. TSV resistance model is made of

two main elements: the first one represents diffusion barrier resistance, and the second one TSV copper resistance, including skin effect. Table 4.7 summarizes the electrical performance of this high density TSV whose integration process is presented previously.

Table 4.7: High density TSV electrical parameters

Electrical parameter	value
C_{ox} (fF)	40
C_{Si} (fF)	7
G_{Si} (mS)	0.4
t_{Si} (μm)	15
D_{Si} (μm)	4
R ($m\Omega$)	170
L (pHm)	10

Buffers (or inverters: an inverter consists of two MOS transistors) serve to regenerate the signal that may fade on the transmission TSV (figure 4.16). As TSVs are too short (the signal may spread rapidly), two buffers are used: a first one on the beginning of the TSV (driver buffer) and a second one at the end of the TSV (receiver buffer).

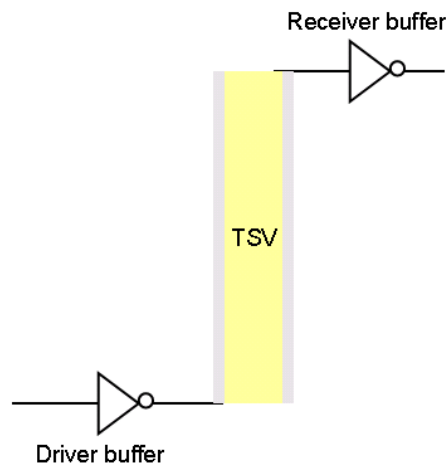


Figure 4.16: Driver and receiver buffers of a TSV

Based on values of table 4.7, the TSV physical model was realized using the Cadence Opus tool and simulated using the analog simulator ELDO. With a buffer driver 13 in 65nm CMOS technology, signal propagation delay is 50 ps. The delay model is used in the next section to perform a comparison between a 3D architecture and its equivalent 2D version in terms of performance.

5 Case study: downlink part of the 4x2 LTE mode

To assess the performance and the power consumption of the proposed platform, we choose to deal with the downlink part of the LTE standard, and more specifically

with the receiver side. The system is designed to transmit on 4 antennas and to receive on 2 antennas (4x2 MIMO), which requires a high performance processing, because of the implementation of diversity and spatial multiplexing schemes. Data are transmitted in 10ms frames equally divided in 10 sub-frames also called TTIs (Time Transmission Intervals). A TTI is composed of 14 OFDM symbols and lasts 1ms (at a sampling frequency of 15.36 MHz) [67]. 4 OFDM symbols contain pilot subcarriers with predetermined values that are used to estimate the transmission channel.

As said previously, the benchmark application is composed of 3 tasks:

1. OFDM demodulation,
2. Channel Estimation for each RX antenna,
3. MIMO MMSE decoding based on a 4x2 double-Alamouti algorithm.

Data processing after MIMO decoding is performed by several demodulation operators (de-mapping, de-interleaving, channel decoding ...) to move from frequency samples (represented as complex numbers) to a stream of binary data. In this work, we consider a 2-layer 3D system resulting from stacking 2 instances of the basic circuit. The targeted application is mapped to this 3D platform as shown in figure 4.17.

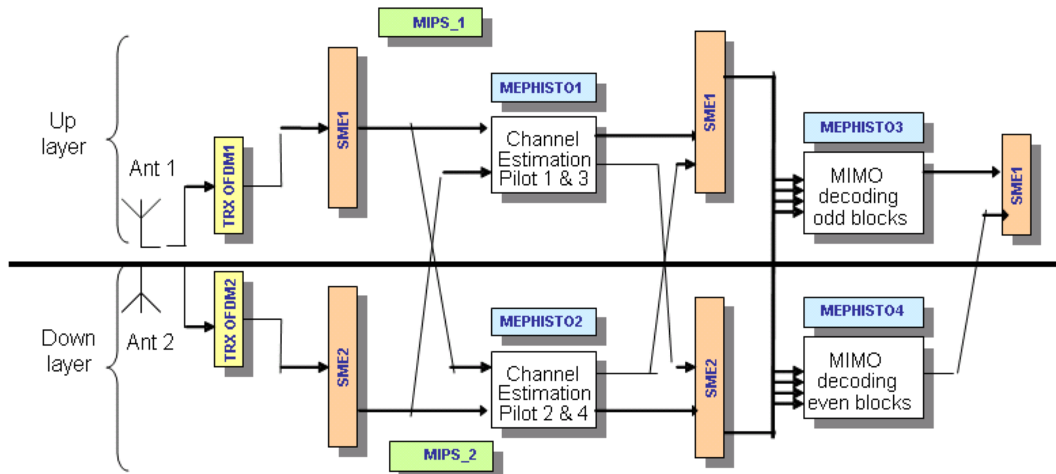


Figure 4.17: Mapping of the 3GPP LTE application on our 3D platform

In this work, we take as 2D reference architecture a platform called MAGALI devoted to wireless telecom applications. A silicon prototype is fabricated using the STMicroelectronics CMOS 65nm LP technology [68]. MAGALI platform focuses mainly on 3GPP LTE standard and aims multi antennas schemes (MIMO). It supports OFDMA/MIMO TX/RX baseband algorithms. MAGALI architecture consists of the same units and the same NoC used in our basic circuit. The MAGALI platform control (scheduling and configuration) is semi-distributed. The global control is centralized and performed by an ARM11 host processor. In order to make comparison with our 3D platform, we use only a sub-set of the MAGALI platform as shown in figure 4.18. This sub-set will be referred to as MAGALI in the rest of the chapter. It has a total area of 7.18mm^2. The 3D system resulting from stacking 2

instances of the basic circuit is functionally equivalent to the MAGALI 2D platform. Thus, the 3D approach achieves up to 50% enhancement in form factor compared to the planar version.

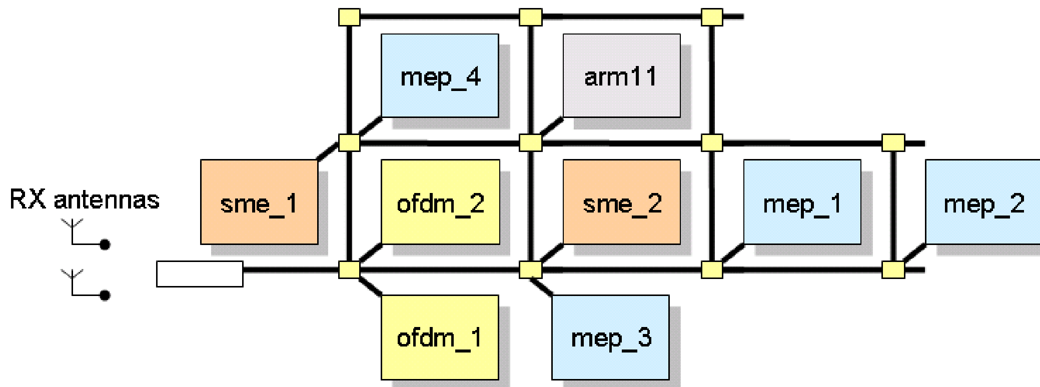


Figure 4.18: 2D reference architecture MAGALI

5.1 Performance results

Simulation environment includes 2 SystemC-TLM data generators to emulate the incoming data-flow from the 2 antennas. To speed up simulation, the asynchronous NoC is modelled in SystemC-TLM with post-layout parameters (the same TLM model described in chapter 3). All processing units of the 2 platforms are modelled at RTL level to provide cycle-accurate results. During simulation, a SystemC-TLM unit, called the recorder, records a trace of the output data-flow. This trace is compared to a reference file to check the obtained values and guarantee the right execution. Figure 4.19 depicts an abstract view of the simulation platform.

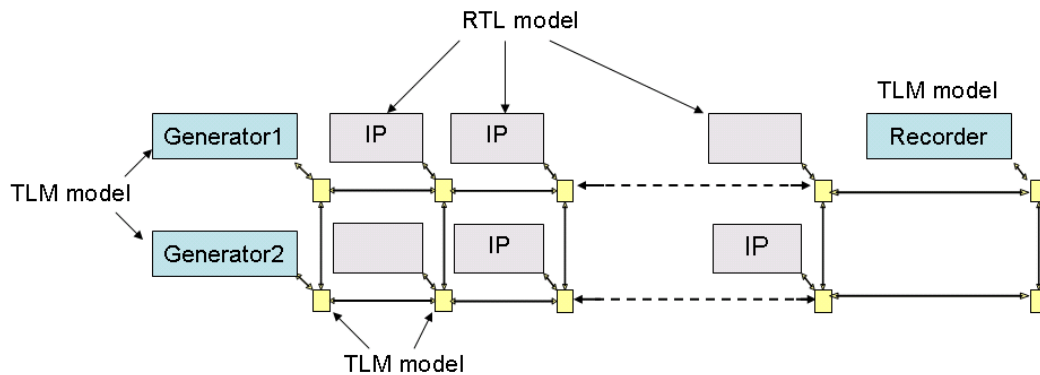


Figure 4.19: An abstract view of the simulation platform

Table 4.8 summarizes the performance results corresponding to the processing time of a complete TTI including scheduling and reconfiguration phases. Host processors of the 2 platforms run at 300MHz clock frequency, while processing units run at 400MHz. In this work, our challenge was to keep at least the same performance to guarantee hard real-time constraints.

As depicted is table 4.8, the execution time is almost the same in the two platforms. This is expected since we are using the same processing units on the 2

platforms. A speed up of 4% is achieved by the 3D approach thanks to the use of short vertical TSV interconnects. From 2D MAGALI to the 3D 2-layer system, the scheduling and reconfiguration management are transferred from centralized host CPU to 2 distributed MIPS processors. By efficiently using the smaller MIPS processor, there is no time overhead due to the communication between the 2 control processors of the 2 layers.

These results based on RTL simulation, confirm that the 3D platform with a distributed control is as efficient as the 2D MAGALI platform with a centralized controller.

Table 4.8: Time to process a TTI

Platform	2D MAGALI	3D platform
Execution time	$500.3\mu s$	$481\mu s$
performance speed up	-	4%

5.2 Power consumption results

To provide a full comparison, we have evaluated the power consumption of the 2 platforms. Each platform was placed and routed in 65nm low-power CMOS technology. A complete TTI processing was simulated with the placed and routed netlist.

Table 4.9 presents the average power consumption of the different processing units of the 2 platforms at gate level. The contributions of the FFT, data processing and data manipulation are the same for the 2 platforms since we are using the same units.

Table 4.9: Power consumption of the processing units

	Power consupmtion (mW)
FFT	172
Processing	207
Data manipulation	258

The control in the 3D platform is performed by two MIPS processors, which deal only with control. Table 4.10 depicts the average power consumption of the MIPS control processor corresponding to different activity rates. With 0% activity rate, the MIPS processor consumes 14.9 mW corresponding to the clock tree power consumption. By efficiently programming the MIPS processor, it is active only 0.4% over the time of TTI period to perform scheduling and reconfiguration, and its power consumption reaches 19mW. Therefore, it is possible to achieve an important power consumption gain (more than 50%) by inserting a simple clock gating mechanism in the MIPS processor. Figure 4.20 depicts a detailed view of the power consumption of the MIPS processor profile during a TTI processing.

The control in the 2D MAGALI platform is performed by one ARM11 processor, which deals also with the MAC (Medium Access Control) layer. Its total power consumption is 150mW. Power consumption due to control is estimated to be 70mW (less than 50% of the total power consumption of the ARM11).

Table 4.10: Power consumption of the MIPS control processor

Platform	Activity rates	Power consumption (mW)
MIPS without clock gating	0%	14.9
MIPS without clock gating	0.4%	19
MIPS with clock gating	0.4%	9

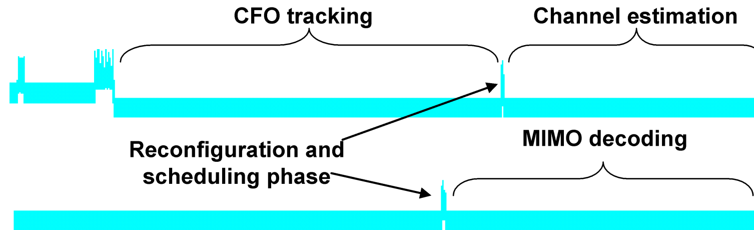


Figure 4.20: Power consumption profile of the up-layer MIPS processor

Table 4.11 depicts power consumption due to control within the 2D and the 3D layer. The MIPS processors are more efficient in term of power consumption than the ARM11 processor.

Table 4.11: Power consumption due to control

Platform	2D MAGALI	3D platform
Power consumption due to control (mW)	70	20

Finally, it could be concluded that, based on post place and route simulation results, sthe distributed control approach introduce a low power consumption overhead to perform the control of the whole 3D platform.

6 Cost analysis

In the previous sections, we present a reconfigurable and stackable circuit for LTE telecom applications. We perform a rigorous comparison between a 3D same-die stacked system (built by stacking 2 instances of the proposed basic circuit) and a 2D reference architecture. Results confirm that the same-die stacking approach is possible in the case of LTE applications, and that overall system performance and power consumption are not affected compared to the 2D approach.

As explained in chapter 1, 3D same-die stacked architectures are obtained by stacking several instances of the same die. This allows building a wide range of systems while reusing always the same mask set. Therefore, it would be possible to avoid designing a new mask set for each new application. As mask cost is prohibitive for aggressive technologies, 3D same-die stacking would reduce considerably final cost in some cases. In this section, we perform a quantitative investigation of the economical benefits of 3D same-die stacking, based on the same cost models and the same numerical values as in chapter 2. The only difference lies in the formula used to calculate the die cost: the mask cost is divided by $N \times L$ in the case of same-die tacking instead of N in the general case (N is the 3D IC production volume, and L

is the number of stacked layers within the 3D IC):

$$C_{die} = \frac{C_{wafer}}{N_{die/wafer}} + \frac{C_{mask}}{N \times L}$$

As already seen in the second chapter, 3D stacking (of different dies) is cost-effective only in the case of large-sized design. However, in this section, it is proven that 3D same-die stacking is beneficial even for small-sized circuits in some cases. To illustrate the economic benefits of the same-die stacking approach, assume that a semiconductor company is designing 3 digital circuits: a low-range, a medium-range and a high-range circuit, to meet market demands in terms of electronics dedicated to LTE applications. Each one of these circuits has its own processing performance. The low-range circuit corresponds to the basic circuit defined earlier in this chapter. The proposed basic circuit has a small area (no more than $4mm^2$). The high range-circuit corresponds to the 4x4 transmission mode. According to table 4.6, its computational performance is 10 times stronger than the basic circuit. Therefore, its 2D version is 10 times larger (in term of area) than the low range-circuit, and its 3D version may be obtained by stacking 10 instances of the low-range circuit. Similarly, the mid-circuit corresponds to the 4x2 transmission mode. According to table 4.6, its computational performance is 2 times stronger than the basic circuit. Therefore, its 2D version is 2 times larger (in term of area) than the low range-circuit, and its 3D version may obtained by stacking 2 instances of the low-range circuit. When using the classical 2D approach, a new mask set has to be designed for each circuit. When using the same-die stacking 3D approach, only the mask set of the low-range circuit (which is the basic circuit) is needed to be designed. All other circuits can be built using this same mask set, by stacking multiple instances of the same low range circuit. Total production volume of the 3 circuit types (high, mid and low-range) is given by:

$$P_{total} = P_H + P_M + P_L$$

where P_H, P_M and P_L are the production volumes of the high, mid and low-range circuits respectively. h, m and l and are the fractions of P_{total} corresponding to production volumes of the high, mid and low-range circuit respectively. They are given by:

$$h = \frac{P_H}{P_{total}}$$

$$m = \frac{P_M}{P_{total}}$$

$$l = \frac{P_L}{P_{total}}$$

Total cost of the 3 types of circuits is given by:

$$C_{total} = P_H \times C_H + P_M \times C_M + P_L \times C_L$$

where C_H, C_M and C_L are unit costs of high, mid and low-range circuits respectively. In this analysis, we keep using the same data as in chapter 2. Figures 4.21 and 4.22 depict total cost of the 3 digital circuits, for 2 different production volumes

(1 million and 10 million respectively) and different values of (h,m,l).

In the case of low total production volume (P_{total}=1 million), the W2W same-die stacking approach is more economical than the D2W and the IbS approaches. As an illustration, for (h,m,l)= (90%,5%,5%), using the W2W approach allows reducing total cost by 3.5 and 4.5 times compared to the D2W and IbS approaches (respectively). This may be explained by the high-yield of the basic circuit (thanks to its small area). Therefore, there is no need to test dies before stacking them.
Besides, the W2W approach is more beneficial than the 2D approach for all (h,m,l) combinations. For example, for (h,m,l)= (5%,90%,5%), total cost when using the 2D approach is twice higher than total cost when using the W2W approach. This is can not be due to 3D stacking since it is beneficial only in the case of large-sized circuit (as seen in chapter 2). The real reason is the mask reuse allowed by the same-die stacking approach.
Moreover, another interesting idea is to design only a 2D high-range circuit that is able to address all the LTE modes ((h,m,l)=(100%,0,0)). In this case, only one mask set would be required (that of the high-range circuit). Figure 4.21 shows that this idea may give better economical results than the W2W same-die stacking approach in some cases, where the high-range circuit is the most produced (h $>$ 50%). For instance, for (h,m,l)= (90%,5%,5%), this approach allows reducing total cost by 1.5 times compared to the W2W approach.
To summarize, in the case of small total volume production, and for low volume production of high-range circuits, the W2W same-die stacking provides the best results in term of cost.

In the case of high total production volume (P_{total}=10 million), classic 2D approach is the most cost-effective, compared to all 3D same-die stacking schemes. As an illustration, for (h,m,l)= (90%,5%,5%), the 2D approach allows reducing total cost by 1.5 times compared to the W2W approach. Indeed, the large number of produced circuits allows amortizing the cost of the 3 mask sets required by the 2D approach. Therefore, the contribution of mask cost to the final circuit cost is drastically reduced. As a result, the cost of the 2D circuits becomes less than any of the 3D circuits.

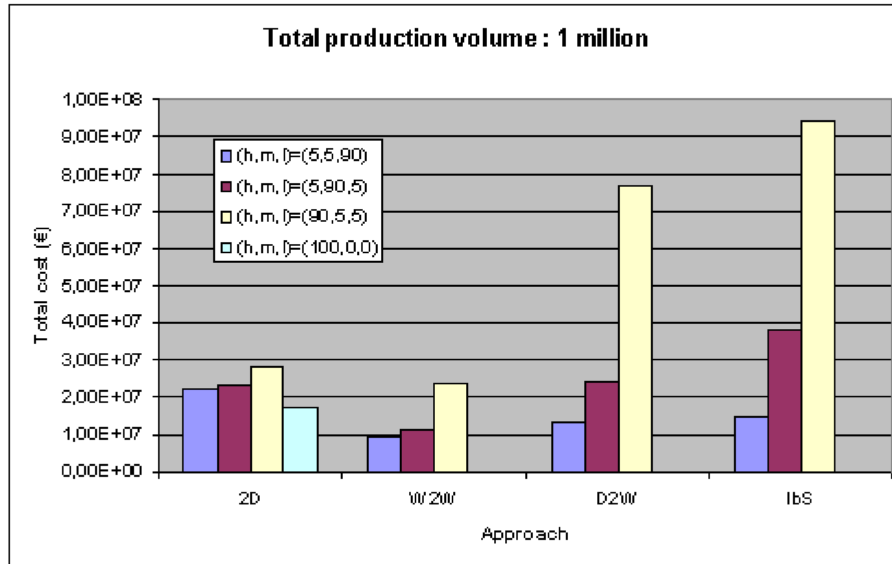


Figure 4.21: Total cost of the 3 digital circuits for a total production volume of 1 million

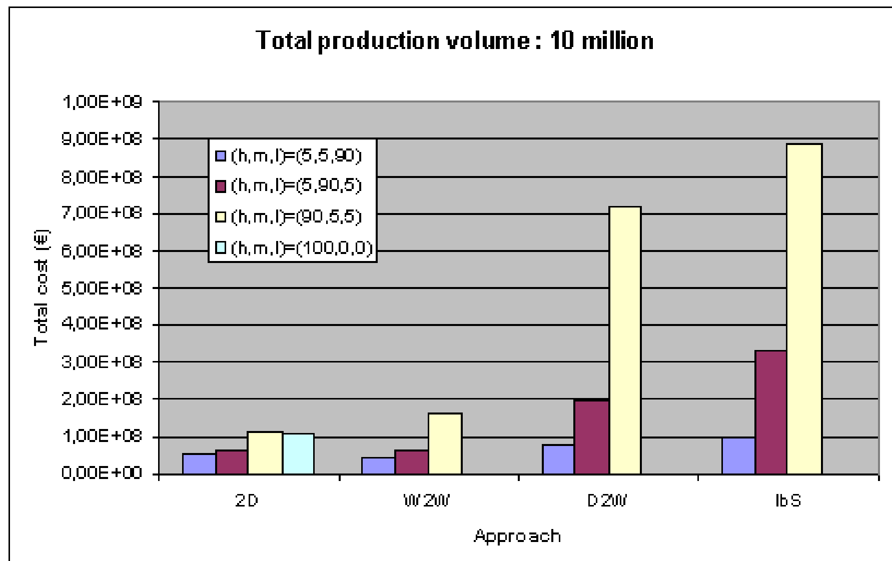


Figure 4.22: Total cost of the 3 digital circuits for a total production volume of 10 million

To conclude, 3D stacking of different dies in cost-effective only in the case of large-sized design. For small-sized circuits, 3D integration becomes cost-effective when using the 3D W2W same-die stacking approach, in the case of low total production volume, and for low production volume of high-range circuits. Figures 4.23 and 4.24 depict the cost-effective options (among the 2D, 3D W2W, 3D D2W and 3D IbS approaches) for different circuit-sizes and production volumes, in the case of 3D stacking of different dies, and 3D same-die stacking respectively.

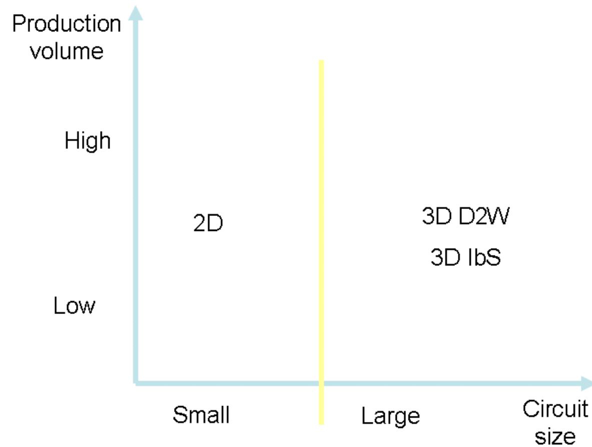


Figure 4.23: Cost-effective approaches in the case of 3D stacking of different dies

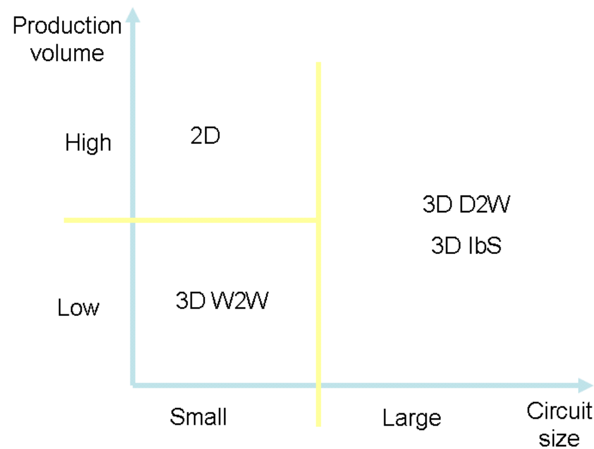


Figure 4.24: Cost-effective approaches in the case of 3D same-die stacking

7 Conclusion

In this chapter, we propose a reconfigurable and stackable circuit for 3GPP LTE (3rd Generation Partnership Project Long Term Evolution) telecom applications. The proposed circuit can meet the computational requirements of the SISO (Single Input Single Output) transmission mode. By stacking several instances of this same circuit, it would be possible to boost overall system performance and address several MIMO (Multiple Input Multiple Output) modes.

In this work, we focus only on the baseband processing of the downlink part (reception chain) and omit all other components of LTE user equipment UE such as radio-frequency and analogue functions, higher layer protocols and multimedia processing. The baseband processor receives the digitized signal as complex samples and performs OFDM demodulation, channel estimation and finally MIMO decoding. Each of these 3 steps has computational requirements that depend on the transmission mode (number of transmission antenna, number of reception antenna) for a given bandwidth.

The proposed reconfigurable and stackable circuit is intended to provide hardware resources that meet these requirements. It is composed 5 elements interconnected thanks to a NoC. The first component is the Smart Memory Engine which is Micro-programmable Memory Controller (MMC) designed to perform data synchronization and distribution in dataflow systems. The second component is the OFDM core devoted to perform direct and inverse fast Fourier transform (FFT and IFFT). The proposed circuit includes also 2 DSPs dedicated to perform complex matrixs computation, useful for channel estimation, advanced MIMO coding/decoding. The global control of the previously described units is performed by a 32-bit MIPS processor, by means of direct addressing and interrupts mechanisms. It is in charge of dynamic reconfigurations, real time scheduling and synchronizations. When multiple basic circuits are stacked (to build a 3D system), global control is distributed between all the MIPS processors of the resulting 3D stack. All the units of the basic circuit are interconnected via a NoC. This NoC is also used to connect all the components of a 3D system resulting from stacking 2 or more basic circuits. The basic circuit is designed as Globally Asynchronous Locally Synchronous (GALS) system. Processing units are synchronous (each one has its own clock frequency), while NoC routers are implemented in Quasi-Delay Insensitive asynchronous logic. The GALS approach allows avoiding issues related to global clock signal distribution to perform multi-plane synchronization.

To assess the performance and the power consumption of the proposed platform, the 4x2 LTE mode is implemented on a 3D system resulting from stacking 2 instances of the basic circuit. A rigorous comparison between a 3D same-die stacked system and a 2D reference architecture confirms that the same-die stacking approach is possible in the case of LTE applications, and that overall system performance and power consumption are not affected compared to the 2D approach.

Besides, based on the cost models presented in chapter 2, the same-die stacking approach for LTE applications provides good results in terms of cost (compared to the 2d approach) in some cases when using the W2W assembly scheme.

The major limitation of the same-die stacking approach is the thermal constraints due to the stacking of several highly active logic layers. High temperatures may limit the operating frequencies of vertically-stacked chip and degrades chip reliability.

Future work will investigate potential solutions for this problem either technological (by inserting thermal vias) or architectural (enhancing power management by means of dynamic task mapping for example).

Conclusions and perspectives

The electronics industry has achieved unusual development over the last decades, mainly due to the rapid progresses in integration technologies. These advances have been led by Moore's law that expects the increase of transistor density. Nevertheless, more scaling according to this empirical law seems to be unsure and very costly, due to several issues such as the exponential increase of leakage and the not-so-exciting gain in performance for aggressive technologies. 3D integration technology is emerging as a promising solution to continue miniaturizing integrated circuits without following Moore's law.

3D integration technology is to stack many circuits vertically and connecting them using Through- Silicon-Vias (TSVs). This results in smaller circuit footprint and shorter vertical interconnections. The potential benefits of 3D integration can vary depending on approach; they include multi-functionality, increased performance, reduced power, small form factor, reduced packaging, increased yield and reliability, mask reuse, flexible heterogeneous integration and reduced overall costs. All these benefits make 3D stacking a promising solution to overcome present limitations of aggressive CMOS technologies, such as limited performance gains, reliability problems, prohibitive fabrication cost...

There are many technological options to fabricate a 3D integrated circuit. A critical issue is to choose the way to assemble chips (die-to-die, wafer-to-wafer or die-to-wafer) and to decide on how active levels are oriented (face-to-face, face-to-back). The fabrication includes 4 major steps: wafer thinning, bonding, alignment and TSVs' formation. The manufacturing yield of a 3D circuit depends on the die yield and the bonding yield.

In spite of its benefits relevant to cost and performance, 3D stacking is facing some major challenges. The first one is the thermal issue caused by the increased power density. Current solutions for this problem consist of thermal driven floorplanning, placement and routing techniques and inserting thermal vias within the 3D stack. Other major issues of 3D integration are the TSV area overhead, the insufficiency of CAD tools, the not-enough comprehension of 3D testing problems and the lack of design-for-testability (DFT) solutions.

When moving from a 2D IC to its 3D version, the partitioning may be performed according to 3 different granularities: macroscopic blocks, functional units and basic operators. All of these 3 types of partitioning allow improving the performance and power with rates that increase when the partitioning granularity decreases. In the same time, issues such as redesign efforts, thermal stress... become more and more noticeable when miniaturizing the size of the stacked elements. Taking into account 3D IC economical and technical limitations, macro-block stacking seems to be the most viable approach. For this reason, this thesis is oriented to focus on

this partitioning granularity, and more specifically with interconnection architecture inside macro-block partitioned 3D systems. In this context, we propose a new 3D NoC topology that allows performance enhancement.

When partitioning a 2D circuit to obtain its 3D version, it is important also to know if this partitioning is cost-worthy. A system-level cost analysis model is proposed to investigate the economical trends of 3D integration. Based on this cost model, 3D stacking turns out to be cost-effective (compared to classical 2D approach) in the case of large-sized circuits when using the D2W or the IbS integration schemes. Another observation is that the optimal number of layers (in terms of cost) depends on how large the design is when using the D2W or the IbS integration approaches. The second contribution of this thesis focuses on cost-aware 3D architectures. Using this same cost model, we investigate the cost-effectiveness of the 3D same-die stacking approach, with a real case study on 4G telecom applications.

Regarding interconnection architecture inside macro-block partitioned 3D systems, our contribution is to propose and evaluate a new hierarchical router for NoC-based 3D MPSoCs. This router is fully implemented in asynchronous logic in order to allow low latency transfer. This router is hierarchical as it is composed of 2 totally decoupled modules: one for intra-layer communication and one for inter-layer communication. A comparison of the hierarchical 3D NoC against 3D mesh NoC shows that the hierarchical has almost the same area and power compared to 3D mesh. A fast SystemC- TLM simulator is used to evaluate the performance of the hierarchical router in term of throughput and latency. Simulation results that the proposed hierarchical router can outperform the 3D mesh by more than 25% in terms of throughput and latency using transpose traffic. The proposed NoC architecture will be implemented on a real 3D IC demonstrator. This allows to measure performance and to validate simulation results.

Regarding cost-aware 3D architectures, we propose a reconfigurable and stackable circuit for 3GPP LTE telecom applications. The proposed circuit can meet the computational requirements of the SISO (Single Input Single Output) transmission mode. By stacking several instances of this same circuit, it would be possible to boost overall system performance and address several MIMO (Multiple Input Multiple Output) modes. The proposed reconfigurable and stackable circuit is composed of: 1 Micro-programmable Memory Controller (MMC), 2 Digital Signal Processors (DSP) and 1 OFDM core. The global control of these units is performed by a 32-bit MIPS processor, by means of direct addressing and interrupts mechanisms. It is in charge of dynamic reconfigurations, real time scheduling and synchronizations. When multiple basic circuits are stacked (to build a 3D system), global control is distributed between all the MIPS processors of the resulting 3D stack. These elements are interconnected thanks to an asynchronous NoC. This NoC is also used to connect all the components of a 3D system resulting from stacking 2 or more basic circuits. To assess the performance and the power consumption of the proposed platform, the 4x2 LTE mode is implemented on a 3D system resulting from stacking 2 instances of the basic circuit. A rigorous comparison between a 3D same-die stacked system and a 2D reference architecture confirms that the same-die stacking approach is possible in the case of LTE applications, and that overall system performance and power consumption are not affected compared to the 2D approach. Besides, based on the cost models presented in chapter 2, the same-die stacking

approach for LTE applications provides good results in terms of cost (compared to the 2D approach) in some cases when using the W2W assembly scheme.

A first perspective of this work is to deal with the thermal constraints within 3D same-die stacking architectures due to the stacking of several highly active logic layers. 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 technological (by inserting thermal vias) or architectural (enhancing power management by means of dynamic task mapping for example).

Another perspective is to investigate the application of the same-die stacking approach on other application fields that are more likely to provide better results in term of cost. These applications should require MPSoC architectures with larger sizes and better modularity properties. Examples of these architectures may include general purpose server processors, such as the microprocessor chip for the IBM zEnterprise 196 (z196) system ($512mm^2$) [69], and the Poulson Intel Itanium processor ($544mm^2$) [70].

Bibliography

[1] G. Poupon. *Packaging avancé sur silicium.* Lavoisier, 11, rue Lavoisier, Paris, 2008.

[2] Patrick Dorsey. Xilinx stacked silicon interconnect technology delivers breakthrough fpga capacity, bandwidth, and power efficiency. *Xilinx white paper*, October 2010.

[3] W.P. Maly and D. Yangdong. 2.5-dimensional vlsi system integration. *IEEE Transactions on Very Large Scale Integration (VLSI) System*, 13:668– 677, June 2005.

[4] C.H. Yu. The 3rd dimension-more life for moore's law. *Microsystems, Packaging, Assembly Conference*, 13:1–6, Oct. 2006.

[5] L. Labrak and I. O'Connor. Heterogeneous system design platform and perspectives for 3d integration. *In Proceeding of 2009 International Conference on Microelectronics*, pages 161 – 164, 2009.

[6] P. Leduc et al. Enabling technologies for 3d chip stacking. *International Symposium on VLSI Technology, Systems and Applications*, 13:76–78, April 2008.

[7] Xiangyu Dong and Yuan Xie. System-level cost analysis and design exploration for three-dimensional integrated circuits (3d ics). *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 234 – 241, 2009.

[8] V. F. Pavlidis and E. G. Friedman. *Three-dimentional integrated circuit design.* Morgan Kaufmann, Burlington, USA, 2009.

[9] J. Cong, J.Wei, and Y.Zhang. A thermal-driven floorplanning algorithm for 3d ics. *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 306 – 313, 2004.

[10] E. Wong and S. K. Lim. 3d floorplanning with thermal vias. *Proceedings of the 2006 conference on Design, automation and test in Europe*, pages 1–6, March 2006.

[11] S. Das, A. Chandrakasan, and R. Reif. Design tools for 3d integrated circuits. *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 53 – 56, 2003.

[12] H. Lee and K. Chakrabarty. Test challenges for 3d integrated circuits. *IEEE Design and Test of Computers*, pages 26–34, 2009.

[13] P.R. Morrow et al. Three-dimensional wafer stacking via cu-cu bonding integrated with 65-nm strained-si/low-k cmos technology. *IEEE Electron Device Letters*, 27:335–337, May 2006.

[14] T. Watanabe et al. Novel retinal prosthesis system with three dimensionally stacked lsi chip. *Proceeding of the 36th European Solid-State Device Research Conference*, 27:327–330, Sept. 2006.

[15] E. Culurciello and P. Weerakoon. Vertically-integrated three-dimensional soi photodetectors. *IEEE International Symposium on Circuits and Systems*, 27:2498–2501, May. 2007.

[16] J. Van Olmen et al. 3d stacked ic demonstrator using hybrid collective die-to-wafer bonding with copper through silicon vias (tsv). *IEEE International Conference on 3D System Integration, 2009*, 27:1–5, Sept. 2009.

[17] G.H. Loh, Y. Xie, and B. Black. Processor design in 3d die-stacking technologies. *Micro IEEE*, 27(3):31–48, May-June 2007.

[18] B. Black a et al. Die stacking (3d) microarchitecture. *IEEE International Symposium on Microarchitecture*, pages 469–479, 2006.

[19] G.L. Loi al. A thermally-aware performance analysis of vertically integrated (3-d) processor-memory hierarchy. *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 991– 996, July 2006.

[20] T. Kgil et al. Picoserver : Using 3d stacking technology to enable a compact energy efficient chip multiprocessor. *SIGOPS Oper. Syst. Rev.*, 40(5):117–128, December 2006.

[21] B. Black, D. W. Nelson, C. W., and N. Samra. 3d processing technology and its impact on ia32 microprocessors. *IEEE International Conference on Computer Design No22*, pages 316–318, 2004.

[22] K. Puttaswamy and G. H Loh. Dynamic instruction schedulers in a 3-dimensional integration technology. *In Proceedings of the ACM/IEEE Great Lakes Symposium on VLSI*, pages 153–158, 2006.

[23] V. Balaji et al. Architecting microprocessor components in 3d design space. *Proceedings of the 20th International Conference on VLSI Design held jointly with 6th International Conference: Embedded Systems*, pages 103–108, January 2007.

[24] K. Puttaswamy and G. H Loh. Implementing register files for high-performance microprocessors in a die-stacked (3d) technology. *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, 2006.*, 00(6-), March 2006.

[25] Roshan Weerasekera, Li-Rong Zheng, Dinesh Pamunuwa, and Hannu Tenhunen. Extending systems-on-chip to the third dimension: Performance, cost and technological tradeoffs. *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design San Jose, California*, pages 212–219, 2007.

[26] P. Mercier, S. Singh, K. Iniewski, B. Moore, and P. O'Shea. Yield and cost modeling for 3d chip stack technologies. *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pages 357 – 360, September 2006.

[27] P. Marchal, B. Bougard, G. Katti, M. Stucchi, W. Dehaene, A. Papanikolaou, D. Verkest, B. Swinnen, and E. Beyne. 3-d technology assessment: Path-finding the technology/design sweet-spot. *Proceedings of the IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, 97(1):234 – 241, January 2009.

[28] P.K. Nag, A. Gattiker, Wei Sichao, R.D. Blanton, and W. Maly. Modeling the economics of testing: a dft perspective. *IEEE Design and test of Computers*, 19(1):29 – 41, 2002.

[29] John H. Lau. Tsv manufacturing yield and hidden costs for 3d ic integration. *Proceedings of the Electronic Components and Technology Conference (ECTC)*, pages 1031 – 1042, 2010.

[30] W. J. Dally and B. Towles. Route packets, not wires: on-chip inteconnection networks. *Proceedings of the 38th annual Design Automation Conference*, pages 684 – 689, mai 2001.

[31] P. Guerrier and A. Greiner. A generic architecture for on-chip packet-switched interconnections. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition 2000*, pages 250 – 256, 2000.

[32] C.A. Zeferino, M.E. Kreutz, L. Carro, and A.A. Susin. A study on communication issues for systems-on-chip. *Proceedings of the 15th Symposium on Integrated Circuits and Systems Design*, pages 121 – 126, 2002.

[33] P. Wielage and K. Goossens. Networks on silicon: blessing or nightmare ? *Proceedings of the Euromicro Symposium on Digital System Design*, pages 196 – 200, 2002.

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

[35] L. Benini and G. De Micheli. Networks on chips: a new soc paradigm. *Computer*, 35:70 – 78, 2002.

[36] E. Rijpkema, K. Goossens, and J. Dielissen A. Radulescu, J. van Meerbergen, P. Wielage, and E. Waterlander. Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip. *IEEE Proceedings of Computers and Digital Techniques*, 5:294–302, 2003.

[37] V.F Pavlidis and E.G Friedman. 3-d topologies for networks-on-chip. *Proceeding of 2006 IEEE International SOC Conference*, pages 285 – 288, 2006.

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

[39] S. Murali, C. Seiculescu, L. Benini, and G. De Micheli. Synthesis of networks on chips for 3d systems on chips. *In proceeding of Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific*, 58:242 – 247, 2009.

[40] J. Kim, C. Nicopoulos, D. Park, R. Das, X. Yuan, N. Vijaykrishnan, M.S. Yousif, and R. Das Chita. A novel dimensionally-decomposed router for on-chip communication in 3d architectures. *ACM SIGARCH Computer Architecture News archive*, 35(2):138–149, May 2007.

[41] Li Feihui, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. Design and management of 3d chip multiprocessors using network-in-memory. *In proceeding of 33rd International Symposium on Computer Architecture*, pages 130 – 141, 2006.

[42] D. Park, S. Eachempati, R. Das, A.K. Mishra, Y. Xie, N. Vijaykrishnan, and C.R. Das. Mira: A multi-layered on-chip interconnect router architecture. *In proceeding of 35th International Symposium on Computer Architecture*, pages 251 – 261, 2008.

[43] Y. Chen, J. Hu, and Ling Xiang. De bruijn graph based 3d network on chip architecture design. *In proceeding of International Conference on Communications, Circuits and Systems*, pages 986 – 990, 2009.

[44] Yi Xu, Yu Du, Bo Zhao, Xiuyi Zhou, Youtao Zhang, and Jun Yang. A low-radix and low-diameter 3d interconnection network design. *In proceeding of IEEE 15th International Symposium on High Performance Computer Architecture*, pages 30 – 42, 2009.

[45] Shan Yan and Bill Lin. Design of application-specific 3d networks-on-chip architectures. *In proceeding of IEEE International Conference on Computer Design*, pages 142–149, 2008.

[46] V.F Pavlidis, I. Savidis, and E.G. Friedman. Clock distribution networks for 3-d integrated circuits. *IEEE Custom Integrated Circuits Conference, 2008 (CICC 2008)*, pages 651– 654., 2008.

[47] T. Ragheb, A. Ricketts, M. Mondal, S. Kirolos, G. Link, V. Narayanan, , and Y. Massoud. Design of thermally robust clock trees using dynamically adaptive clock buffers. *IEEE Transactions on Circuits and Systems*, 56(2):374 – 383, Feb. 2009.

[48] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin. An asynchronous noc architecture providing low latency service and its multi-level design framework. *In proceeding of 11th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 54–63, 2005.

[49] Jens Sparsø and Steve Furber. *Principles of Asynchronous Circuit Design*. Springer, thirteenth edition, 2002.

[50] Y. Thonnart, P. Vivet, and F. Clermidy. A fully-asynchronous low-power framework for gals noc integration. *In proceeding of Conference and Exhibition Design, Automation and Test in Europe, 2010 (DATE 2010)*, pages 33–38, 2010.

[51] A.M. Rahmani, A. Afzali-Kusha, and M. NED Pedram. A novel synthetic traffic pattern for power/performance analy sis of network-on-chips using negative exponential distribution. *Journal of Low Power Electronics*, 5(3):396–405, 2009.

[52] P. Pande, C. Grecu, M. Jones, A. Ivanov, and R . Saleh. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. *IEEE Transactions on Computers*, 54(8):1025– 1040, 2005.

[53] Motorola. Long term evolution (lte): A technical overview. *Technical white paper*, 2007.

[54] Ericsson. Lte - an introduction. *Technical white paper*, June 2009.

[55] Rohed and Schwarz. Umts long term evolution (lte) technology introduction. *Application Note 1MA111*, 2007.

[56] J. Berkmann, C. Carbonelli, F. Dietrich, C. Drewes, and Wen Xu. On 3g lte terminal implementation - standard, algorithms, complexities and challenges. *International Wireless Communications and Mobile Computing Conference, 2008 (IWCMC '08)*, pages 970 – 975, 2008.

[57] C. Jalier, D. Lattard, A. Jerraya, G. Sassatelli, P. Benoit, and L. Torres. Heterogeneous vs homogeneous mpsoc approaches for a mobile lte modem. *In Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE), 2010*, pages 184 – 189, 2010.

[58] M.M. Rana. Channel estimation techniques and lte terminal implementation challenges. *2010 13th International Conference on Computer and Information Technology (ICCIT)*, pages 545 – 549, 2010.

[59] Laurent Boher, Rodolphe Legouable, and Rodrigue Rabineau. Performance analysis of iterative receiver in 3gpp/lte dl mimo ofdma system. *IEEE 10th International Symposium on Spread Spectrum Techniques and Applications, 2008 (ISSSTA '08)*, pages 103 – 108, 2008.

[60] J. Ketonen M. Juntti J.R. Cavallaro. Performance-complexity comparison of receivers for a lte mimo-ofdm system. *IEEE Transactions on Signal Processing*, 58(6):3360 – 3372, 2010.

[61] Zhan Guo and P. Nilsson. A low complexity soft-output mimo decoding algorithm. *2005 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, pages 90 – 93, 2005.

[62] J. Martin, C. Bernard, F. Clermidy, and Y. Durand. A microprogrammable memory controller for high-performance dataflow applications. *Proceedings of ESSCIRC, 2009 (ESSCIRC '09)*, pages 348 – 351, 2009.

[63] F. Clermidy and C. Bernard. A low-power vliw processor for 3gpp-lte complex numbers processing. *Conference and Exhibition Design, Automation and Test in Europe, 2010 (DATE 2010)*, 2011.

[64] C. Jalier, D. Lattard, G. Sassatelli, P. Benoit, and L. Torres. Flexible and distributed real-time control on a 4g telecom mpsoc. *In Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, pages 3961 – 3964, 2010.

[65] Fabien Clermidy, Romain Lemaire, Xavier Popon, Dimitri Ktenas, and Yvain Thonnart. An open and reconfigurable platform for 4g telecommunication: Concepts and application. *Proceedings of the 2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools (DSD '09)*, 2009.

[66] Lionel Cadix et al. Integration and frequency dependent parametric modeling of through silicon via involved in high density 3d chip stacking. *The Electrochemical Society transactions*, 33(12):1–21, October 2010.

[67] 3GPP TSG-RAN. 3gpp ts36.211, physical channels and modulation (release 8). *Technical white paper*, 2007.

[68] F. Clermidy, C. Bernard, R. Lemaire, J. Martin, I. Miro-Panades, P. Vivet Y. Thonnart1, and N. Wehn. A 477mw noc-based digital baseband for mimo 4g sdr. *In Proceeding of IEEE International Solid-State Circuits Conference (ISSCC) 2010*, 53:278–279, February 2010.

[69] J. Warnock et al. A 5.2ghz microprocessor chip for the ibm zenterprisetm system. *In Proceeding of IEEE International Solid-State Circuits Conference (ISSCC) 2011*, 2011.

[70] Shankar Sawant et al. A 32nm westmere-ex xeon enterprise processor. *In Proceeding of IEEE International Solid-State Circuits Conference (ISSCC) 2011*, 2011.

Résumé

Les travaux de cette thèse s'intéressent aux problèmes de performance et de coût des architectures MPSoC à base de NoC, en tirant parti des possibilités offertes par les technologies d'intégration 3D. Plusieurs contributions originales sont proposées. Tout d'abord, une étude approfondie à propos des différentes granularités de partitionnement au sein des circuits 3D est réalisée. En se basant sur cette analyse, ce travail de thèse est orienté aux architectures 3D partitionnées au niveau des blocs macroscopiques. Ainsi, la contribution de l'intégration 3D est limitée aux interconnexions verticales inter-blocs. Afin d'améliorer les performances de ces interconnexions, une topologie hiérarchique de NoC est proposée pour diminuer la latence et augmenter le débit des communications au sein des architectures 3D partitionnées au niveau des macro-blocs. D'autre part, un modèle au niveau du système est présenté pour évaluer et comparer les coûts des différentes options technologiques de l'intégration 3D. Partant de cette évaluation, nous proposons une architecture multiprocesseur reconfigurable empilable pour les applications de télécommunication 4G, en tenant compte des problèmes de coût.

Mot-clefs

Architectures MPSoC, Technologies d'intégration 3D, Réseaux-sur-puce, Analyse de coût des circuits intégrés

---

Title

Architectures multiprocesseurs pour applications de télécommunication basées sur les technologies d'intégration 3D

Abstract

This PhD research is intended to deal with cost and performance issues of NoC-based MPSoC architectures by taking advantage of the opportunities offered by 3D integration technologies. Several original contributions are proposed. First, a deep investigation of the different partitioning granularities within 3D circuits is performed. Based on this analysis, this PhD work is oriented to focus on core-level partitioned 3D architectures, and then to restrict the contribution of 3D stacking to the global inter-block vertical interconnections. To enhance the performance of global interconnect architectures, a hierarchical NoC topology is proposed to improve communication latency and throughput within core-partitioned 3D architectures. On the other hand, a system-level cost analysis model is presented to assess and compare several 3D integration technology options. Based on this evaluation, we propose a cost-aware stackable reconfigurable multiprocessor NoC-based architecture to address the requirement of 4G telecom applications.

Keywords

MPSoC architectures, 3D integration technologies, Network-on-Chip, Integrated circuit cost analysis

---

Adrr : CEA-Leti, MINATEC - 17 rue des Martyrs, 38054 Grenoble cedex 9, France
ISBN : ␣␣␣␣␣␣␣␣␣␣␣␣␣