Computer Engineering
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ce.et.tudelft.nl/

2010

MSc THESIS

3D Architecture Exploration for Multimedia Applications

Winston Siauw

CE-MS-2010-18

Abstract

Three-Dimensional (3D) silicon integration is an emerging technology that vertically stacks multiple silicon circuit layers. It enables a single chip to be divided over multiple layers, which are stacked on top of each other. A literature study is performed for this thesis, which presents the basic manufacturing techniques for this emerging technology. Furthermore, the constraints and properties of the inter-layer interconnect (Through Silicon Via (TSV)) are investigated, and *the architectural potentials* of 3D silicon integration are explored for memory-on-memory, logic-on-logic, memory-on-logic, and a 3D Network-On-Chip (NOC). *Compared to a planar chip, 3D integration provides five key advantages: (1) wider and denser on-chip interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies, and (5) footprint reduction.* Moreover, practical work is performed for this thesis, which evaluates a novel 3D scheme. *The scheme stacks two (or more) 2D processors on top of each other, where the Functional Unit (FU) boundaries between all processors are removed*. Thus, a processor can execute instructions on all the unutilized FUs of all the (remaining) processors. The free FUs on other processors can be utilized for *fault detection* or for *performance improvement*. Experimental results show that on average 52% of the executed instructions can be protected, or a speedup of on average 7% can be achieved. Both schemes are beneficial because no extra dedicated FUs are needed and fault detection and higher performance are achieved at low cost (only additional control logic and TSVs). This is because the boundaries between the FUs of the CPUs are removed.

TU Delft Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science

3D Architecture Exploration for Multimedia Applications

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Winston Siauw
born in Rotterdam, The Netherlands

Computer Engineering
Department of Electrical Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

3D Architecture Exploration for Multimedia Applications

by Winston Siauw

Abstract

Three-Dimensional (3D) silicon integration is an emerging technology that vertically stacks multiple silicon circuit layers. It enables a single chip to be divided over multiple layers, which are stacked on top of each other. A literature study is performed for this thesis, which presents the basic manufacturing techniques for this emerging technology. Furthermore, the constraints and properties of the inter-layer interconnect (Through Silicon Via (TSV)) are investigated, and *the architectural potentials* of 3D silicon integration are explored for memory-on-memory, logic-on-logic, memory-on-logic, and a 3D Network-On-Chip (NOC). *Compared to a planar chip, 3D integration provides five key advantages: (1) wider and denser on-chip interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies, and (5) footprint reduction.* Moreover, practical work is performed for this thesis, which evaluates a novel 3D scheme. *The scheme stacks two (or more) 2D processors on top of each other, where the Functional Unit (FU) boundaries between all processors are removed*. Thus, a processor can execute instructions on all the unutilized FUs of all the (remaining) processors. The free FUs on other processors can be utilized for *fault detection* or for *performance improvement*. Experimental results show that on average 52% of the executed instructions can be protected, or a speedup of on average 7% can be achieved. Both schemes are beneficial because no extra dedicated FUs are needed and fault detection and higher performance are achieved at low cost (only additional control logic and TSVs). This is because the boundaries between the FUs of the CPUs are removed.

Laboratory : Computer Engineering
Codenumber : CE-MS-2010-18

Committee Members :

Advisor:	Sorin Cotofana, CE, Delft University of Technology
Chairperson:	Koen Bertels, CE, Delft University of Technology
Member:	Kees Goossens, Eindhoven University of Technology
Member:	Demid Borodin, CE, Delft University of Technology

This thesis is dedicated to my beloved family,
who has supported me all the way.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis would not be possible without the contribution from many people.
I would like to thank Prof. Kees Goossens for offering me the opportunity to do my graduation project with him at NXP. In the meetings and discussions he always gave me to the point advice and he directed my research in the right direction. Naturally, I would like the thank Benny Åkesson, who has guided me and has read all the reports in detail. His corrections improved my reports and thesis significantly, thanks! We discussed many topics and he always gave me a new point of view to look at a topic. Furthermore, when he dropped by it was always cozy and it was a nice break of the day, since I was the only student in the student room at NXP. I am proud to be part of their team and would like to thank them both for all their advice and (personal) support they gave me. Even with the long reading days in Eindhoven I enjoyed the time there. Furthermore, I would like to thank Prof. Sorin Cotofana for his support and advice he gave me during my graduation period and for all the nice meetings in Delft. I would like to thank Prof. Kees Goossens, Benny Åkesson and Sorin Cotofana explicitly for their personal approach. They allowed me to continue my graduation period and simultaneously enjoy the warmth of my beloved father and family.

I would like to thank Demid Borodin for his advice and providing me with 3D SimpleScalar. Besides helpful, it was always fun to drop by in his office. His advice speedup my programming work, thanks! I would like to thank Demid Borodin, Saleh Safiruddin, Mottaqiallah Taouil and George Voicu for reading and improving my reports. Furthermore, I would like to thank them all for being part of the $3Dim^3$ research group in Delft and for the nice meetings we had.

Finally I would like to thank my family for supporting me all my life and especially during my master education in Delft and graduation. Moreover, I am grateful to have a great father, who has placed my graduation always first even during his most difficult times. Furthermore, I am thankful for having a great girl friend, Ellen, who has supported me during difficult times, and she was always patient during my solitary writing and study days. Without them it would not be possible to graduate right now, thanks!

Winston Siauw
Delft, The Netherlands
July 4, 2010

1 Introduction

From the emergence of the integrated circuits (IC) technology, higher computation power was predominantly achieved by scaling down the transistor size, as Moore's Law described [21]. To increase the computation power, multiple processor elements on one chip are used, but essentially the IC have remained a Two-Dimensional (2D) planar plane. As Richard Fenyman, physicist and Nobel Lauretate envisioned, in 1985, there is tremendous potential in making chips three-dimensional. However, scaling down has been more cost effective than building in the third dimension [1]. Furthermore, dimensional scaling has consistently improved the device performance, in terms of gate switch delay. However, it has had reverse effects on the global interconnect latency. It is because the width of the wires scaled down, but the length of the global wires did not. The length of the wires remained the same, even though the original chip size shrank. It is because there came more systems on a chip. This resulted in a total chip size, which was similar to the previous chip generation. The global Resistive Capacitive (RC) wire delay has become a circuit limitation factor [1, 22, 23]. The number of interconnects, practically possible, towards a core or (on-chip / off-chip) memory has become a limitation as well. Furthermore, the on-chip global interconnect wires consume a tremendous amount of dynamic power at high speeds, due to charging and discharging. Moreover, signals require multiple clock cycles to travel across the entire chip, due to the wire delay and the high speed. A planar chip is not able to accommodate the rising demands for speed and power reduction without compromising performance, process complexity and cost [1].

In this thesis five key advantages are identified for Three-Dimensional (3D) integration (compared to a planar chip): (1) wider and denser (on-chip) interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies [8], and (5) footprint reduction. The major interconnect bottleneck is solved by the wider / denser (on-chip) interconnects and by the wire length reduction [24]. Wire length reduction is achieved by strategically stacking circuits / cores on top of each other, which results in a simultaneous reduction of wire latency and power consumption. Furthermore, the wire length reduction makes (long) pipelines to cross a core or a chip superfluous, which leads to a lower pipeline (startup) latency [8, p.42]. Moreover, the shorter wire lengths reduce the dynamic power consumption during communication. The heterogeneous device technology enables the uses of different technologies per layer, such as different substrate materials (germanium, silicon or Silicon-On-Insulator (SOI)), or with the same substrate material but with different process technologies (nano meter). Furthermore, 3D integration provides a reduced footprint area, which is especially beneficial to the hand-held market.

Moreover, in this thesis a novel 3D scheme is evaluated. The scheme stacks two (or more) 2D processors on top of each other, where the Functional Unit (FU) boundaries between all processors are removed. Thus, a processor can execute instructions on all the unutilized FUs of all the (remaining) processors. The free FUs on other processors can be utilized for *fault detection* (redundant scheme) or for *performance improvement* (performance scheme). The redundant and

performance schemes are beneficial because no extra dedicated FUs are needed and still fault coverage and higher performance are achieved at low cost (only additional control logic and TSVs), respectively. This is because the boundary between the FUs of the CPUs is removed. The impact on *the performance* is on average 8%, 7%, 4% and 2% for the configurations with one to four Integer ALUs (IALUs), respectively. For the *fault detection*, the average number of instructions executed redundantly are 15%, 23%, 35% and 52% for the configurations with one to four IALUs at CPU 2, respectively.

In this thesis, we explore 3D silicon integration with the main research question, which is stated below. The main research question is divided into multiple sub-research questions and they are bounded by the boundary conditions of Sections 1.1 and 1.2, respectively. The answers on the sub-research questions lead to the answer on the main research question, which are presented in Sections 5.2 and 5.3, respectively.

Main research question:

Compared to a 2D chip, what is the architectural potential and impact of a 3D stacked chip?

1.1 Research questions

This section presents the sub-research questions. The answers on the sub-research questions lead to the answer on the main research question. The answers are bounded by the boundary conditions. The main research question is focused on the architectural impact and potential. However, for a *practical* architectural design it is important to know what type of interconnect is used, and what the constraints and properties are for that interconnect. This knowledge allows a designer to understand the implications of its 3D design and thus if it is a *practical and useful* architectural design. Therefore, this work also looked at the *manufacturing part* for 3D integration and this leads to the research questions one until three. Once the constraints and properties for the interconnect are known then the *architectural* impact and possibilities for 3D integration are investigated. This leads to research question four. Furthermore, this work also investigates a novel 3D scheme, as part of a research project for the Delft University of Technology, where the boundary between the Functional Units (FUs) of two stacked processors is removed. It is removed to allow unutilized FUs to be used by other processors. This resulted in the research question five. The sub-research questions are presented below.

Manufacturing

1. What is the best structure to manufacture 3D chips?
2. Which inter-layer interconnect is the best to use with the best structure?
3. What are the properties of the best inter-layer interconnect?

Architecture

4. What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, memory-on-logic, and 3D Network-On-Chip (NOC)?

Practical work

5. What is the impact on the performance and fault coverage if two stacked processors share their functional units?

1.2 Boundary conditions

This section presents the boundary conditions for main research question and the sub-research questions of Section 1.1. These boundary conditions indicate the criteria a research question is assessed / discussed.

The main research question discusses the architectural potential and impact of 3D integration (compared to a 2D chip) for the topics below. These topics are discussed where possible, since 3D silicon integration is a new field and not all information is known or published.

- Possible 3D design opportunities or strategies
- Temperature impact
- Speed
- Power
- Area

Research question one searches for the best structure to manufacture 3D chips, and it considers the 3D monolithic and the 3D stacked structures, since they are the only two structures that can manufacture a 3D wafer. The best structure from *research question one* is selected according to the following criteria:

- Best (largest) thermal budget
- Greatest number of layers possible to manufacture
- Methods to interconnect various device layers

After the best structure is identified the best interconnect is selected. The best interconnect for *research question two* is selected based on the following criteria:

- Pitch
- Speed
- (Dynamic) power
- Maturity

Subsequently, *research question three* discusses the following topics for the best inter-layer interconnect:

- Latency
- Area
- Power
- Yield
- Reliability

Research question four discusses the architectural potential and impact of 3D integration for the topics below. These topics are discussed where possible, since 3D silicon integration is a new field and not all information is known or published.

- Possible 3D design opportunities or strategies
- Temperature impact
- Speed
- Power
- Area

Research question five investigates the *'impact'* of sharing FUs for the performance and fault detection schemes, and the *'impact'* is defined as follows:

- The total speedup is assessed for the performance mode
- The ratio of redundant executed instructions is assessed for the fault coverage.

1.3 Contributions

This section presents the main contributions of this thesis, which is composed of a literature study and practical work. The practical work evaluates a novel 3D scheme.

- The *methods to manufacture a 3D device* are presented, which are the 3D monolithic structures and the 3D stacked structures. The main difference between the approaches is that the monolithic approach has a start wafer where multiple silicon layers are fabricated upon. Conversely, the 3D stacked approach contains multiple single silicon wafers bonded together to form a multi-silicon wafer (or die). (Chapter 2)
- *Inter-layer interconnects are explained and ranked in a table*, which are the capacitive, inductive, TSV, micro bumps, Package-On-Package (POP) and wire bonded interconnects, and the best interconnect is selected / proposed. The best interconnect is selected according to the following criteria: pitch, speed, power consumption and maturity of the technology. (Chapter 2)

- For *memory-on-memory stacking*, it is indicated that the access latency, the power consumption and the footprint area reduces, compared to a 2D memory with an equal amount of bit storage. Furthermore, it shows the *Non-Uniform Cache Access (NUCA) latency variation* and the *memory wall problem* is diminished. (Chapter 3)
- For *logic-on-logic stacking*, it is indicated that the wire length reduction can be used to *eliminate pipeline stages and / or increase the operating frequency* of a design, which has a positive impact on the performance and power reduction. (Chapter 3)
- For *memory-on-logic stacking*, it is indicated that there are four potentials: *(1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies.* (Chapter 3)
- For *3D Network-On-Chip (NOC)*, it is indicated that it creates a true physical 3D NOC topology, where the router and the network are themselves three-dimensional entities. The advantage between the vertical and the horizontal interconnects is that in the vertical direction, a TSV is just a few tens of μm long as compared to a few thousand μm for a on-chip global wire in the horizontal direction [13]. (Chapter 3)
- In this work a novel scheme is evaluated. Two identical 2D processors are stacked and *the FU boundary between the stacked processors is removed*. Thus, one CPU can execute instructions on all the free FUs. The free FUs on the other CPU can be used for error detection or for performance improvement. (Chapter 4)
- *Overview tables* are made of the articles that are discussed in this thesis, and other interesting articles are also placed in that overview, see Appendix H. Furthermore, an overview of the *simulation tools used by the articles* from Appendix H are shown, see Appendix I. *It indicates that Hspice and Spice are commonly used as circuit simulators. The SimpleScalar is commonly used to evaluate the overall performance in a system. Furthermore, the (only) temperature simulator used by the articles is Hotspot.*

1.4 Outline

This section presents the thesis organization. Chapter 2 presents how 3D stacked devices can be / are manufactured and bonded. Furthermore, various inter-layer interconnects are explained and the best inter-layer interconnect (a Through Silicon Via (TSV)) is selected / proposed. Hence, the answers on *research questions one until three* are presented in Chapter 2.

Chapter 3 presents the five key advantages for 3D stacking, which are: (1) wider and denser (on-chip) interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies [8], and (5) footprint reduction. These advantages are discussed for the topics memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC. Thus, Chapter 3 presents the answer on *research question four*.

Chapter 4 investigate a novel 3D scheme. Two identical stacked 2D processors are simulated, where *the FU boundary between the stacked processors is removed*. Thus, one CPU can execute instructions on all the free FUs. The free FUs on the other CPU can be used for error detection or for performance improvement. The results are presented and analyzed and discussed, and thus the chapter presents the answer on *research question five*.

The final chapter, Chapter 5, presents a summary of the whole thesis and all the research answers. Furthermore, the answer is presented on the main research question, which forms the overall conclusion.

2 Manufacturing

The research questions that are answered in this chapter are:

> Research questions one until three:
>
> 1. *What is the best structure to manufacture 3D chips?*
> 2. *Which inter-layer interconnect is the best to use with the best structure?*
> 3. *What are the properties of the best inter-layer interconnect?*

For an architectural designer it is important to understand the constraints and properties of the inter-layer interconnect. This knowledge allows a designer to understand the implications of its 3D design and thus if it is a *practical and useful* architectural design. That is why in this work we look at the *manufacturing part* for 3D integration, and in specific the inter-layer interconnects. *The answers on the three research questions in this chapter have as goal to select the best inter-layer interconnect, and indicate its properties*.

The best 3D structure should be known before the best interconnect can be selected from the candidate interconnects, since each 3D structure has its own particular interconnects. Hence, research question one. *A 3D silicon device (structure) can only be manufactured via the 3D monolithic approach or via the 3D stacked approach.* Therefore, this chapter starts from research question one and proceeds to research question three.

A 3D monolithic device is sequentially fabricated from the bottom layer up. There is only one main substrate, with one or more transistor layers on top. The transistor layers are divided by an isolation layer. The rest of the basics properties of a 3D monolithic device are presented in Section 2.1. Section 2.1.1 identifies the first main *problem* for the 3D monolithic structure, which is important for *research question one*. Section 2.1.2 discusses the first and the two remaining main problems to manufacture the 3D monolithic structures. The first problem is that unavoidable thermal steps, such as gate oxidation [1], are needed to form a high-quality upper silicon layer and MOSFET layer. The second problem is that it is challenging to build more than three device layers on top of each other with the 3D monolithic approach. The final main problem is that it is difficult to manufacturing interconnects to communicate across the various device layers.

Section 2.2 presents the 3D stacked structure. A basic 3D stacked structure uses two or more individual processed wafers, which are grinded, aligned and then bonded. The wafers are fabricated individually, and therefore it resolves the thermal budget problem (first problem), compared to the 3D monolithic approach. Furthermore, wafer bonding can endlessly be done, and thus the second main problem is solved. Moreover, six different interconnect methods are pos-

sible for communication between the layers, which solves the third problem. Thus, Section 2.2 shows that the 3D stacked structure solves all the problems of the 3D monolithic structure. Hence, the *first of the research question is answered* in Section 2.2, *the 3D stacked structure is the best structure to manufacture 3D chips*.

Subsequently, the six possible interconnects for the 3D stacked structure are presented in Section 2.2.1. The interconnects are ranked for all the criteria, which shows that the Through Silicon Via (TSV) has the best properties and can penetrate two or more layers. Thus, it presents the answer on *research question two*. Thereafter, Section 2.2.5 zooms in on the properties of TSVs, and thereby *answering research question three*.

2.1 3D monolithic structures

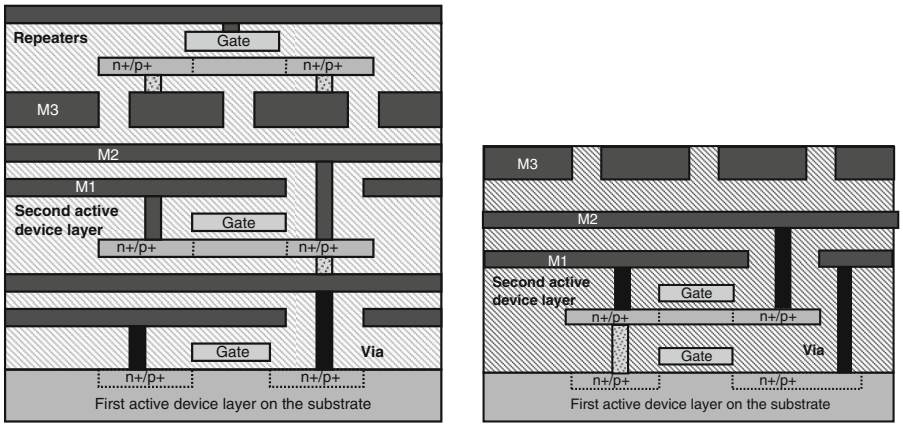
This section indicates that the 3D monolithic structure has three main problems, which are: (1) Maximal three layers can be manufactured, (2) Unavoidable heat steps (thermal budget), and (3) The interlayer communications is difficult to manufacture. The knowledge of these problems is used in Section 2.2, and it shows that the 3D stack structures solves these problems. Hence, *research question one* is answered in Section 2.2.

The 3D monolithic approach, also known as a multi-layer buried structure (MLBS), is sequentially fabricated from the bottom layer up. There is only one main substrate with one or more transistors on top, divided by an isolation layer. A critical step in the manufacturing process is forming a high-quality active silicon layer on top of the isolation layer. The cristallinity / crystal structure of the upper layers is usually imperfect and as result high performance devices cannot be built in the upper layers [1, p.6]. Furthermore, to form the upper layer (incl. the transistor layer), a number of unavoidable thermal steps are needed, such as gate oxidation [1]. Due to the high temperatures, the underlying layer degrades (i.e. unwanted doping diffusion) and therefore a tight thermal budget must be imposed, resulting in a low manufacturing throughput. There are two design approaches possible, based on the topology of the stacked Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) and metal wires (see Figure 2.1). The two options are:

1. Ungrouped metal layers and MOSFET layers
2. Grouped metal layers and MOSFET layers

The ungrouped metal layers and MOSFET layers (the first group) are flexible in vertical and horizontal routing, due to the metal (wire) layer between the active layers in (see Figure 2.1(a)). However, the thermal budget is significantly reduced after the first metal wires are fabricated. This is because the upper silicon layer should be annealed (recrystallized). The temperatures during annealing reaches (almost) the melting point of the metal wires [2, p.41]. Therefore, fabrication becomes impractical with this approach.

With the second approach the MOSFETs are grouped and located at the bottom of the chip (active layer). The metal layers are fabricated on top of the active / MOSFET layer (see Figure 2.1(b)). Thus after annealing, the metal layer is fabricated, which widens the temperature budget. This solves the problem of the first group "The ungrouped metal layers and MOSFET layers ". However, the main challenge lays in the formation of the *high quality silicon* film beyond the first device layer *without harming the MOSFETs* on the lower layer.



(a) Ungrouped MOSFET layers and metal layers

(b) Grouped MOSFET layers and metal layers

Figure 2.1: Approaches to 3D monolithic stacking. [1]

The remaining part of this chapter presents technologies for the "grouped metal layers and MOSFET layers" (second group). Fabrication of the "Ungrouped metal layers and MOSFET layers" (first group) is not practical [2], and therefore not further discussed. Section 2.1.1 presents two methods, used to manufacture the upper silicon layer. Both methods are subsequently used in the following sections to manufacture the upper silicon layer. Afterwards Section 2.1.2 presents two basic approaches to manufacture the MOSFETs (gate, drain and source) onto two silicon layers. Furthermore, Appendix B presents two methods already (previously) in used in the industry, including the FinCMOS method. FinCMOS is considered to be the most promising technology for manufacturing double gates with the 3D monolithic approach [1].

2.1.1 Upper silicon layer fabrication technologies

This section identifies the first main *problem* for the 3D monolithic structure, which is important for *research question one*. The main problem indicated in this section is that *unavoidable thermal steps are needed* to form a high quality upper silicon layer. The next section (Section 2.1.2) uses this knowledge, and shows that this unavoidable heat degrades the conductivity of the lower MOSFET layer.

The first layer is fabricated using the conventional manufacturing process. The upper silicon layer is created with laser recrystallization or with seeds crystallization. Each technique manufactures a layer of silicon, both techniques should not be used together to form a single layer. This section only discusses for the "grouped metal layers and MOSFET layers".

2.1.1.1 Laser crystallization

The first device layer is fabricated with a conventional process and contains either PMOS, NMOS, or a mix of PMOS and NMOS. The lower silicon layer is electrically isolated with an electric isolation layer (SiO_2) [2]. On top of the electric isolation layer, a thick (1μm) planarized heat shield (PHS) is fabricated. This heat shield protects the lower layers from the elevated heat

during the (laser) recrystallization phase. The heat shields between the lower and upper layer, used at fabrication time, has a negative effect on the heat dispersement during normal operation, resulting in high temperatures in the upper planes [2]. The active upper layer is formed by Chemical Vapor Deposition (CVD), which forms polysilicon islands (partial surface coverage), or a thin polysilicon film (total surface coverage). A CVD uses chemical reactions between multiple gaseous molecules (called precursor) and turns it into a solid material, which lands on (are deposed) the substrate. Many chemical reactions form a small island or thin film on the surface of the substrate. These polysilicon islands or films do not have a crystallized structure. Therefore, they are radiated by an Argon-laser, which recrystallizes the island / film into single-grain crystals (see Figure 2.2) [2]. Three silicon layers can be fabricated, at most, by these techniques.

In general, the temperatures at the recrystallization phase reaches up to 950 degrees, but upper layer device formation is also demonstrated with only 600 degrees [2]. These temperatures approaches, or exceeds the melting point of the interconnect metals. Therefore, the interconnect metal layers need to be fabricated after the last silicon layer.

A major drawback is the *quality* of the grown silicon layer of the *upper planes*, and the effect of the high temperatures on the *electrical characteristics of the lower planes*. The channel conductivity of the upper plane is slightly worse, compared to the lower plane [1, 2]. Since the conductivity is degraded at the upper planes, the (slower and larger) PMOS transistors need to be at the bottom planes and the (faster and smaller) NMOS at the upper planes. This equalizes the speed differences. Maximum three device layers, in total, can be fabricated with this technique.

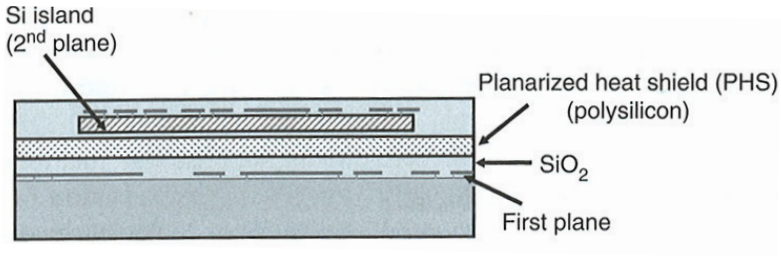


Figure 2.2: Cross section of two planes, used at laser crystallization. [2]

2.1.1.2 Seed crystallization

Another technique to fabricate multiple device layers on top of (bulk) silicon is with seed crystallization. On top of the first oxidized silicon layer an amorphous silicon (a-si) layer, is deposed and crystallized into polysilicon grains (see Figure 2.3.(a)). Amorphous silicon is similar to normal silicon. However, it does not have a regular crystalline structure and therefore it needs to be crystallized. The amorphous silicon layer is deposited with a Low-Pressure Chemical Vapor Deposition (LPCVD) process (see Figure 2.3.(a)). A LPCVD uses a CVD process in a low pressurized chamber, which transforms multiple gaseous molecules into a solid material. A patterned film is deposited, with openings for the seeds to settle in (see Figure 2.3.(b)-(c)). The seeds are made from Nickel (NI) or germanium (Ge). After the seeds are deposed in the openings, the temperature is elevated, which crystallizes the amorphous silicon underneath the seeds (see Figure 2.3.(d)). This thermal annealing process is necessary to alter the structure of the silicon. The temperature reaches up to 900 degrees, depending on the process used. Afterwards, the gates are

manufactured according to the conventional (etching) process (see Figure 2.3.(e)) [2, p.42]

There are two possible methods to form a gate in the upper layer, via dual seeding or via single seeding. With dual seeding, two seeds are used to fabricate the drain and source. With single seeding, a single seed is split up into a drain and source (see Figure 2.3.(D) and 2.3.(E)).

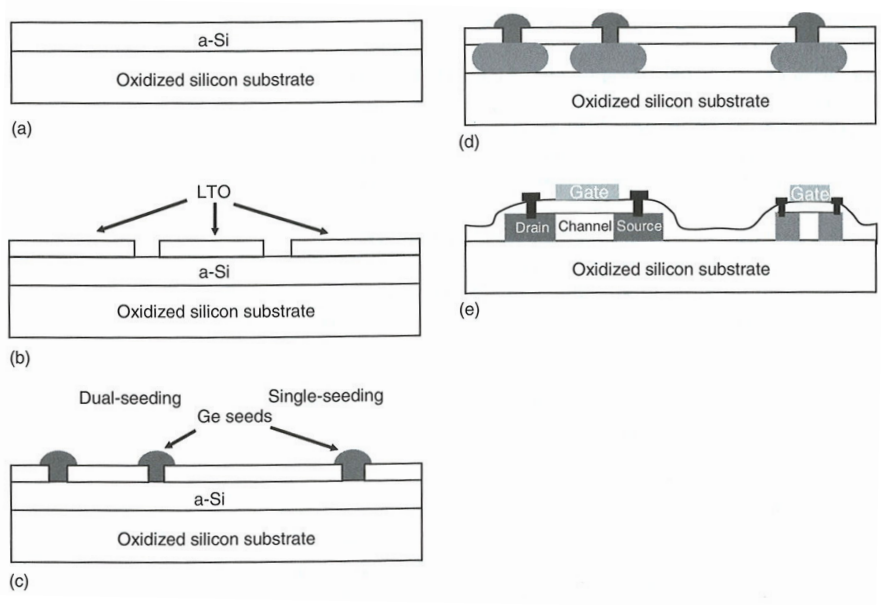


Figure 2.3: Processing steps for seed crystallization. The upper and lower silicon layers are depicted but the isolation layer is not depicted. (a) Deposition of amorphous silicon, (b) A patterned low-temperature oxide (LTO) creates seeding windows, (c) Deposition of seeding materials, (d) Producing silicon islands via thermal annealing (e) Fabrication of gates via the conventional method. [2]

2.1.2 General process challenges

This section identifies the remaining two main problems for the 3D monolithic structure, and shows that 3D monolithic structure is not the best structure, with respect to *research question one*. The first problem is that it is *challenging to build more than three device layers* on top of each other with the 3D monolithic approach, due to the high temperatures and the lower MOSFET gates are hard to reach. The second main problem is that it is *difficult to manufacturing inter-layer interconnects*, which are used for communication across various device layers.

In Section 2.1.1 we have seen how the upper silicon layers are manufactured. This section uses that knowledge to shows that the lower MOSFETs gates experiences that unwanted heat, due to the manufacturing of the upper silicon layer. There are two general approaches to manufacture two silicon layers with MOSFET gates, the layer-by-layer approach and the simultaneous multi-layer approach. The main difference between the two approaches is that with the layer-by-layer approach, the MOSFETs are fabricated per layer. Conversely, with the simultaneous multi-layer approach, the MOSFETs are fabricated on all layers at once (simultaneously). Two *general* approaches are presented, since there are many possible processes possibilities to

manufacture a layer-by-layer and a simultaneous multi-layer device. This section is intended to give a general idea for all possible 3D monolithic processes. Two existing manufacturing processes, in uses in the industry, are presented in Appendix B. These processes are experiencing the problems described in this section.

2.1.2.1 Layer-by-layer process

With the layer-by-layer approach, the MOSFETs in the active layers are serially (layer-by-layer) fabricated (see Figure 2.4). The process starts with a conventional silicon substrate where N-wells and gates are fabricated, at elevated temperatures, forming the lower active layer (see Figure 2.4.(a)-(b)). The elevated temperature is needed to dope the N-wells. The lower layer, the isolation layer, and the upper layer are fabricated on top of each other. On the upper layer, the gates and P-wells are fabricated at elevated temperatures, harming the lower doped silicon layer.

A problem with the layer-by-layer technique is the duration and amount of heat (thermal budget) that the wafer has to withstand. The lower layer has to withstand more thermal cycles, than the upper MOSFET and metal layers [1, p.33], [23]. This results in serious dopant diffusion in the source, drain and gate regions. That means that the P-wells and N-wells become less concentrated with dopants, and thus it results in a *conductivity degradation*. The thermal budget, limits the dopant activation process of the upper layers, which means that (typically) *maximal three layers* are produced with this layer-by-layer technique [1, p.34]. Nevertheless, it was commonly used to fabricate early stacked SRAM cells.

2.1.2.2 Simultaneous multi-layer process

With the simultaneous multi-layer processing technique, the MOSFETs in the active layers are simultaneously fabricated (see Figure 2.5). Therefore, it solves the thermal budget problem, compared to the layer-by-layer approach. It is resolved, because the gates, drain and source (P-wells and N-wells) are created simultaneously (to dope the N-wells or P-wells). The high temperature is only needed once, which is equal to a regular 2D fabrication process.

The two active silicon layers are also known as double Silicon-On-Insulator (SOI) and could be created with a double Separation by IMplantation of OXygen (SIMOX) process. Alternatively, the techniques described in Section 2.1.1 (laser crystallization or seed crystallization) could be used.

A SIMOX process uses an oxygen ion beam implantation process to implant a layer of oxygen atoms in the silicon. After this the temperature is increased to create a buried silicon dioxide layer (SiO_2), which is an isolation layer. This isolation layer, electrically splits the silicon up into an upper part and a lower part.

A simultaneous multi-layer process begins with a start wafer, which contains two layers of silicon (see Figure 2.5.(a)). Then the shapes of the gates are fabricated, and afterwards the gate, drain and source are doped (N-wells and P-wells) (see Figure 2.5.(b)-(d)). At the doping stage, the temperature is elevated to enable the doping process. Afterwards, the MOSFETs are connected to the metal wired plane via etched vias (see Figure 2.5.(e)-(f)). *However, three main challenges remain:*

1. Gate definition of the lower layers

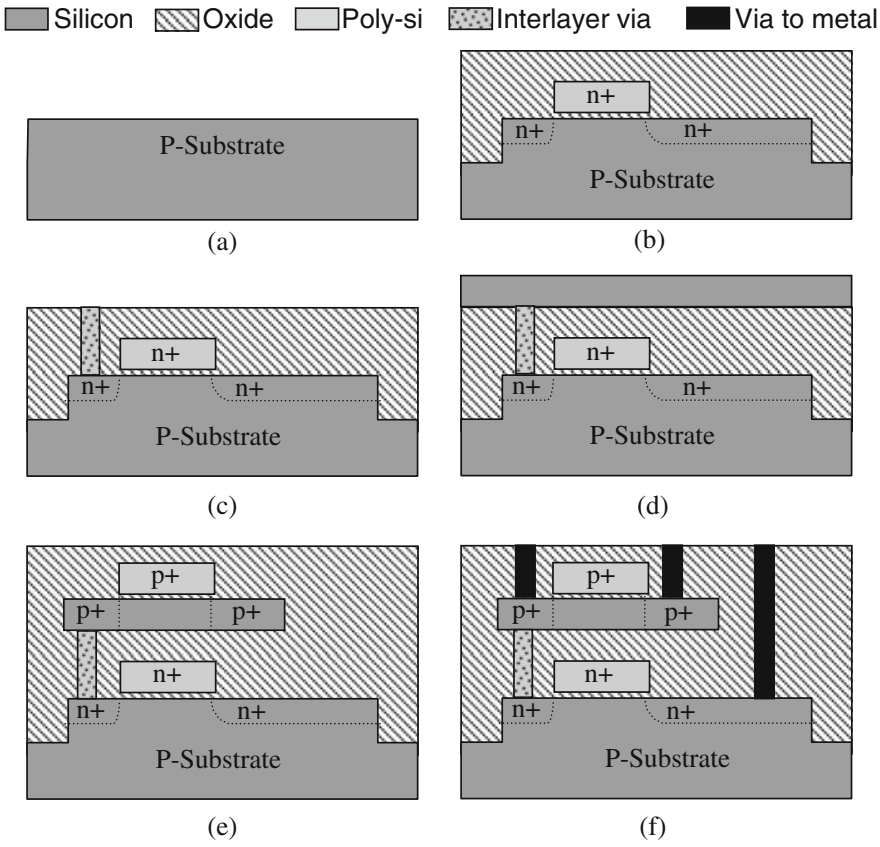


Figure 2.4: An example of a (possible) layer-by-layer stack CMOS process: (a) Starting with a silicon wafer, (b) Following regular NMOS process with shallow trend isolation to form a NMOSFET up to oxide passivation and planarization steps, (c) Opening of via for interlayer connection, (d) Formation of upper-layer silicon film for PMOSFET fabrication, (e) Follows the conventional SOI process to form a PMOSFET, (f) Opening of vias to both the top and bottom layers for metal interconnect. [1]

2. Doping of the bottom devices

3. Interconnecting different layers

The first challenge is the gate definition of the lower layers. The *gate width / length definition is difficult*, because the upper layers prevent lithography of the lower layer and etching underneath the gates. A special process is required to form gates at the bottom layers. A wet-etch process is used, and this process etches the desired silicon regions from the upper and lower silicon to form MOSFETs (see Figure B.1.(a)-(c)). The gates at the lower layers are fabricated via a special tunneling process. However, this process is very difficult to control (see Figure B.1.(f)). At the lower silicon layers over-sized gates are formed, to ensure complete exposure of the contact points of the MOSFET. The over-sized gate causes a large gate-to-drain and gate-to-source capacitance (see Figure 2.5.(e)).

The second challenge is doping the bottom devices. Doping the silicon layers is a crucial

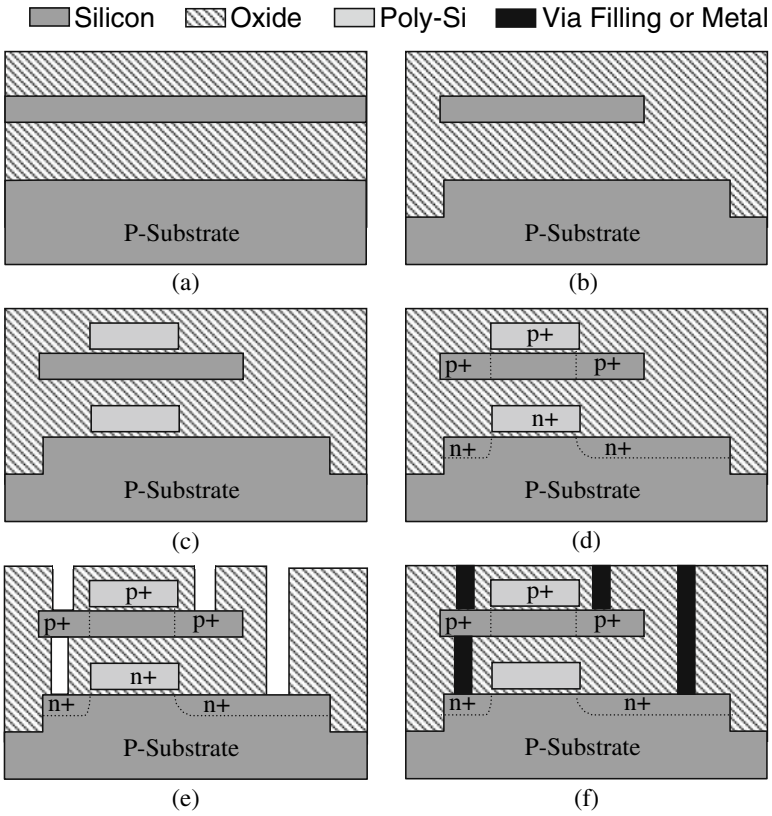
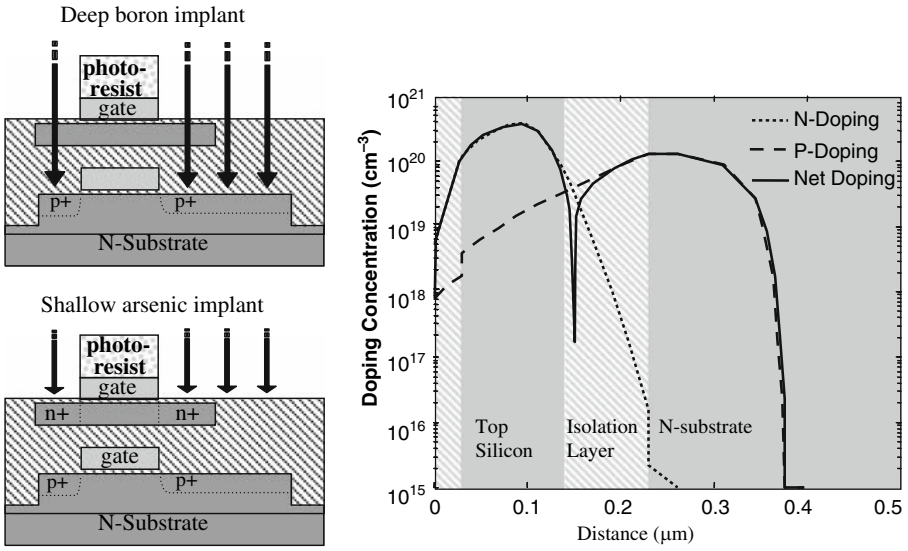


Figure 2.5: An example of a (possible) simultaneous multilayer stacked CMOS process: (a) Starts with a wafer with two or more active silicon layers, (b) Active area formation by etching and refill, (c) Gate formation for devices in all layers, (d) Gate and source/drain doping for all active devices, (e) Contact opening to all active layers, challenging to etch below the top layer , (f) Contact filling for metal interconnect. [1]

step, because a doping process creates P-wells and N-wells in the silicon layers, which is needed for the drain and source of a FET. However, the upper silicon layer is covering the lower silicon layer(s) and therefore it is *difficult to dope the lower silicon layer(s)*. A high-energy ion implant process is the most common technology used to introduce dopants through the upper layers, and it leaves a dense concentrated implant area at the desired depth (see Figure 2.6(b)). Boron has a deeper implant reach/depth compared to phosphorus or arsenic (see Fig 2.6(a)). Therefore, it is preferred to place the PMOSFETs at the lower layers. To relief the thermal budget, all implanted dopants in all layers are activated at the same time.

The final challenge is the *interconnection between the different layers*. Contact between the upper and lower active layers is made by a via, which connects the active layer and a metal layer. As stated previously, the metal layer should be fabricated after the upper active layer is manufactured to avoid thermal budget problems. This metal layer connects the MOSFETs in the upper and lower layers. The *upper MOSFET gate is covering the via path* that needs to be connected to the lower MOSFET (see Figure 2.5.(e)). The contacts of the MOSFET (drain and source) should be at different horizontal places to enable vertical the vias, between the upper



(a) Multi-layer ion doping implantation

(b) Doping concentration per silicon layer

Figure 2.6: (a) Ion implantation to introduce dopants into the top and bottom active layers with different energy. (b) Simulated doping profile for boron and arsenic at the different layers of silicon containing different amount of concentrations of doping. [1]

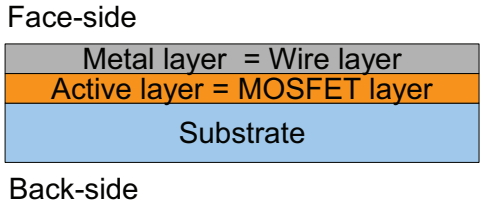
and the lower silicon layers (see Figure 2.5.(f)). Therefore, it reduces the circuit density and consumes more (silicon) area.

2.2 3D stacked structures

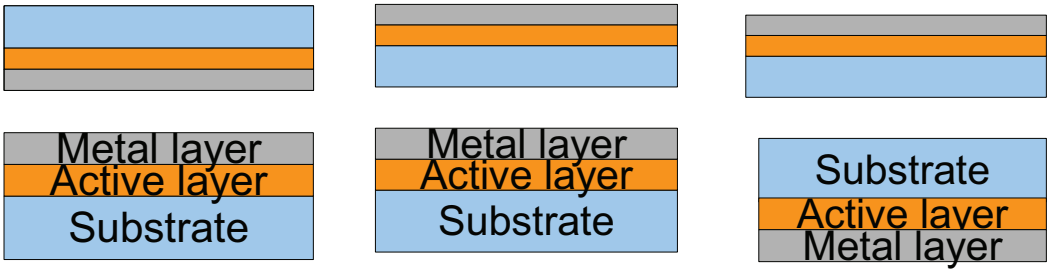
In the previous section, Section 2.1, we have seen that the 3D monolithic structure had three main problems, which are maximal three layers, unavoidable heat steps (thermal budget), and the inter-layer communications is difficult to manufacture. This section will *answer the first research question*, and it indicates that the 3D stacked structure is the best structure to manufacture 3D chips.

The remainder of this chapter discusses the 3D stacked structure. In this section the 3D stacked structure is presented. The main difference between this structure and the previously presented monolithic approach, from Section 2.1, is that the monolithic approach uses only one start wafer where multiple silicon layers are fabricated upon. *Conversely, with the 3D stacked approach there are multiple single silicon wafers bonded together to form a multi-silicon-wafer (or die),* see Figure 2.7(b) until 2.7(d).

Wafer bonding is an emerging fabrication technology and often seen as the key to enable full 3D integration. A basic 3D stacked structure uses two or more individual processed wafers, which are grinded, aligned and then bonded. *The wafers are fabricated individually, and therefore it resolves the thermal budget problem (first problem)*, compared to the 3D monolithic approach. Furthermore, *wafer bonding can endlessly be done, and thus the second main problem is solved.* Moreover, *six different interconnect methods are possible for communication between*



(a) A single wafer / die exist out off three layers, the substrate, active layer, and the Metal layer.



(b) Face-to-face bonding. (c) Face-to-back bonding. (d) Back-to-back bonding.

Figure 2.7: A basic 3D stack of two layers.

the layers, which solves the third problem and they are explained and mutually compared in the next sections. *Thus, 3D stacking solves the three problems from the 3D monolithic approach.*

The substrate of the wafers can be thinned (<100μm) before the bonding process. This reduces the distance between the two dies, which makes the interconnection faster. Furthermore, it reduces the height and weight of the stacked dies [15, p.3, p.5]. Grinding is done in a three-step-process. The coarse-grind process removes the majority of the substrate. The fine-grind process, and a stress-relief step remove the remaining material. Due to the coarse grind process, micro cracks appear on the surface of the substrate (up to a depth of 10-20μm). The fine-grinding process removes the sub-surface damage, up to a depth of 2μm. The stress-relief step removes the last 2μm of the damaged substrate, otherwise cracks can propagate through the whole structure. Stress can be relieved by four different processes: dry or wet polishing and dry or wet etching. Current grinding tools uses wet or dry polishing [1, p.97]. With dry polishing, a pad is used to buff off the damaged layer without any liquids. The wet polishing process uses chemicals, and it is therefore also known as Chemical Mechanical Polishing (CMP). The CMP process uses an abrasive and corrosive chemical slurry. The chemicals in the slurry also weakens and/ or reacts with the material, which should be removed. The abrasive accelerates this weakening process, and the polishing pad helps to wipe the reacted materials from the surface. The dynamic polishing head is rotating with different axes of rotation (i.e. not concentric). This head removes material and makes the wafer planar, which is important because any imperfection decreases the bond surface.

The wafers are thinned (grinded) to thicknesses of less than 100μm and therefore they lose their rigidity and tend to bend and buckle under their own weight. Handling these wafers becomes impossible. However, there are two solutions. The first solution is to bond the wafer

onto the 3D wafer stack to obtain more rigidity, and grind afterwards. The second solution is with the help of a supporting wafer ("handle wafer") [15, p.27]. The handle wafer is temporarily bonded at the front side (the metal layer) of the wafer. This is done before the grinding process and therefore it has more rigidity. After grinding and bonding, the handle wafer should be removed. This requires special properties of the bond between the handle and the wafer. It should be strong enough to withstand grinding, but loose enough to come lose at the desired time. Special techniques, such as thermoplastics (wax), UV curable materials, lamination tapes (double sided tapes) and metal thermo-compression bonding (has a release layer) are used [1, p.107].

3D bonding is done at die or at wafer level, and therefore it leaves three options:

1. Die-to-die (D2D)
2. Die-to-wafer (D2W)
3. Wafer-to-wafer (W2W)

With wafer-to-wafer bonding, the manufacturing throughput is higher than with the other two options. Furthermore, wafer aligning is much more accurate than the other options, due to the fact that a wafer is bigger than a die. Higher accuracy means a higher density of 3D interconnects. However, wafer-to-wafer bonding forces the dies (on the wafer) to have the same size. Moreover, when a die is faulty, it is still bonded, which results in a yield loss of the whole bonded die stack.
Die-to-wafer does not impose a size restriction onto the devices and therefore bigger and smaller dies can be bonded together. Furthermore, the dies can be tested and the known good dies (KGD) are selected and then bonded.
The lowest manufacturing throughput is with die-to-die bonding, due to aligning and pick-and-place of the dies. Similar to die-to-wafer bonding, the dies can be tested, and the KGD are selected and then bonded. The throughput is an important fabrication factor, because the cost of bonding is in general determined by the throughput. Especially, bonding processes with high temperatures need long processing time [1, p.265]. Of course, the yield also plays an important factor in determining the bond level choice.

When considering the bonding sides of a wafer / die, three options are possible:

1. Face-to-face (F2F)
2. Face-to-back (F2B)
3. Back-to-back (B2B)

The side with the metalization is the face-side of the wafer/ die and the substrate side is the back of the wafer/ die, see Figure 2.7(a). Face-to-face bonding is useful for binding two dies together, see Figure 2.7(b). Face-to-back bonding is useful for binding three or more dies together, because then the manufacturing bonding steps are the same, see Figure 2.7(c).

In the first part of this section research question one is answered.

Research question one: What is the best structure to manufacture 3D chips?

The 3D stacked structures is the best, since it can stack unlimited layers, no heat budget restrictions, and there are six different inter-layer interconnects. Conversely, the 3D monolithic structure has three main problems, which are maximal three layers, unavoidable heat steps (thermal budget), and the inter-layer communications are difficult to manufacture.

2.2.1 Interconnects

This section introduces the six interconnects for inter-tier communication (communication between two or more dies / wafers). A tier refers to a single die or wafer layer. In the remaining part of this paper, we refers to a single die or wafer layer, as a tier. The next section, Section 2.2.3, mutually *compares and ranks the interconnects*, and in that section is the comprehension of the presented interconnects from this section mandatory. Together, these two sections provides the answer on research question two.

Research question two:

Which inter-layer interconnect is the best to use at the best structure?

The inter-layer communication can be done via contactless or normal interconnects. The contactless interconnects are first presented, and afterwards are the normal interconnects are presented. A contactless interconnection is either a capacitive or inductive interconnection, presented in Section 2.2.1.1 and Section 2.2.1.2, respectively. Afterwards, the chapter continues with the normal "contact" interconnections, which are: the wire bonded, the Package-On-Package (POP), the micro bumps and the Through Silicon Via (TSV) interconnections.

2.2.1.1 Capacitive inter-plane communication

Contactless inter-plane communication is based on inductive or capacitive coupling. Both methods use alternating current (AC) and this technology is, therefore, also known as AC coupled interconnects. The advantage of the contactless approach is that a minimal amount of substrate thinning is required. This reduces the fabrication costs [25, p.500]. Furthermore, special processing steps are not required, such as through silicon etching or manufacturing bumps. However, the challenges are found in distributing the clock, AC and DC power via these contactless couplings between multiple tiers. In this section, capacitive coupling is presented.

Capacitive inter-plane communication uses a half capacitor on the face-size a tier. Two aligned face-to-face bonded tiers, forms a whole capacitor. The material between the tiers functions as a dielectricum, and is also determining the pitch of the interconnects (the distance to the next capacitor) [25]. Face-to-face bonding is preferred, because the dielectricum between the planes is thinner. Therefore, the size of the capacitor can be reduced, which results in a higher

interconnect density. On each end of the capacitor, a transmit and a receive circuitry is needed. A challenging task is to interconnect three or more tiers with this approach [2,3,25], because then a face-to-back bonding is required. Face-to-back bonding increases the dielectricum between the capacitor planes, which affects the interconnect density.

The transmitter that is connected to the capacitor uses a charge pump and the receiver uses an amplification circuit (see Figure 2.8) [2, p.53,54]. The parasitic capacitance of the capacitor is reduced by the use of Silicon-On-Sapphire (SOS). SOS is a thin silicon layer (<0.6μm), grown on top of a sapphire substrate, and is part of the Silicon-On-Insulator (SOI) family. The main disadvantage of this approach is the size of the capacitors, which affects the interconnect density. The size is influenced by the inter-plane distance and the dielectric constant of the material between the planes. However, it is proven that a 8*8μm electrode can reach communication speeds of 1.23GB/sec [2, p.54].

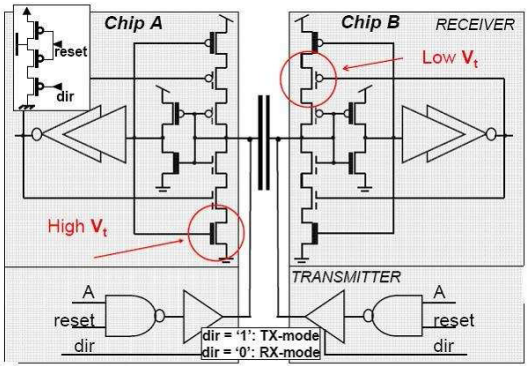


Figure 2.8: Asynchronous capacitive bi-directional circuit, tier "1" (left) and tier "2" (right) communicate through the capacitor. [3]

2.2.1.2 Inductive inter-plane communication

An inductive inter-plane communication uses one spiral coil per tier and two aligned tiers forms an inter-chip transformer (see Figure 2.9). The signal propagation is achieved through current pulses. The magnetic coupling coefficient (k) depends strongly on the separation distance (d) between the primary and secondary coil [4] (see Figure 2.9). The distance, at face-to-back bonding, is in general similar to the thickness of the substrate of a tier, which makes thinning of the substrate advantageous. With a thinner substrate, the coils can be made smaller and therefore the interconnect density is larger. There are two advantages, the manufacture process is less complex and the coils do not occupy area in the active layer, compared to TSVs [26]. The main disadvantage is that the transmit circuitry consumes a lot of power, compared to TSVs [26, p.2195], [2, p.55], and power and clock distribution are realized through galvanic connections, which could induce jitter. Furthermore, interference in horizontal or vertical direction is challenging.

It is proven that communication with 2.8Gb/s is possible with a inductor diameter of 150μm, tolerating 50μm of misalignment and consuming in total 47.6mW [4]. Furthermore, the pitch is determent by the cross talk interference. Two (neighboring horizontal) coils are tested with a spacing of 50μm (1/3 of the coil diameter) and frequencies up to 5GHz and a 40dB of isolation has been demonstrated [4].

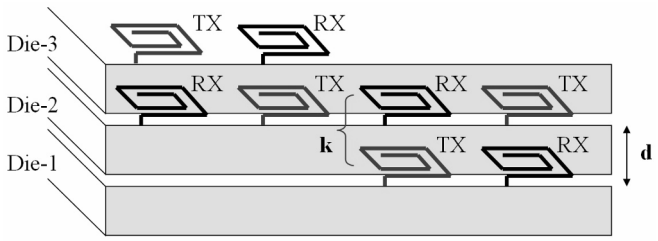


Figure 2.9: Uni-directional inductive coupled interconnects, tiers are face-to-back bonded. [4]

2.2.1.3 Wire bonds

Chip stacking began with mounting smaller thinned dies onto larger ones, which were then glued and wire bonded together. The whole bonded stack was placed in a single package (see Figure 2.10.(a)). Thinning is done to reduce the height and weight of the stack. The substrate is thinned to a thickness of $\approx$50μm-70μm [2, p.25].

Different techniques exist to bond tiers on top each other, such as multi-row wiring, placing a spacer (dummy piece of silicon) between tiers and die-to-die wire bonding. Multi-row wiring is utilized to increase the number of interconnects (see Figure 2.10.a.), and therefore it increased the throughput, compared to off-chip communication. A spacer creates the necessary space (<100μm) between two tiers, for bond wire loops (see Figure 2.10.(b)). Die-to-die bonding reduces the length of the wires (compared to multi-row wiring), and therefore is the parasitic capacitance reduced, which speeds up the interconnect (see Figure 2.10.(c)).

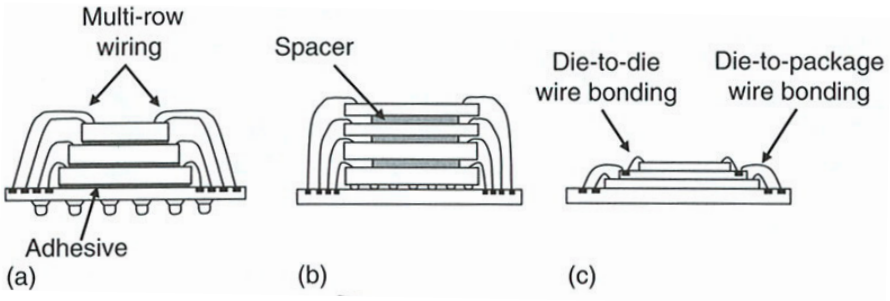


Figure 2.10: a) Smaller dies on top of each other with multi-row bonding, b) Wire bonded tiers, separated by spacers, c) Die-to-die and die to package bonding. [2]

The number of tiers that can be stacked, within a package, depends on the used spacers and tiers. A typical wire bonded chip stack has a height of two tiers. However, a wire bonded chip stack is demonstrated up to five layers [15, p.25]. The packages use a Ball Grid Array (BGA) to connect to a circuit board. The BGA uses solder balls with a diameter of 0.75mm and 0.2mm and a pitch of 1.27mm and 0.35mm, respectively [15, p.5]. The performance and interconnect density is dependent on the bond wires (parasitic capacitance), which is disadvantageous. Therefore, other methods were preferred and used by the industry, such as POP, TSV or bumps [15, p.25].

2.2.1.4 Package-On-Package

Package-On-Package (POP) has a better manufacturing yield per stacked device, compared to the wire bonded interconnection. Therefore, POP became popular. Even though the height of the total package and the bill of materials increased [15, p.5].
In POP stacking, chips are placed on an interposer with or without a package (see Figure 2.11.a). An interposer provides communication to other dies in the stack and has multi-layer wiring [27]. A die can be connected to the interposer in three ways, via: a ball grid array (see Figure 2.12.a), a through hole via at chip level (see Figure 2.12.b), or a via at print circuit board level (see Figure 2.12.c) [2, p.30]. The solder balls do not only provide electrical interconnection between the planes, but also mechanical support. To reinforce the mechanical durability of this System-In-Package (SIP), an epoxy under filling is applied between two planes. An important characteristic is that low temperatures are used to bond and compress the stack. The expected height of a ten plane POP method is 1mm, which is a significant reduction compared to wire bonding [2, p.28]. The through hole electrodes are using small bumps with diameters of 20-30μm to connect the die with the interposer. The larger solder balls, that are connecting the interposers to other interposers, have a diameter of 30-50μm.

The advantage of POP, is that the inter-die throughput and density improves, compared to bond wires or off-chip interconnections. However, the size of the solder ball and the bumps are an obstacle for devices that need a high interconnection density [2, p.30]. Another problem is the rewiring at the interposer, needed by some techniques, as this increases the inter-plane wire length [2, p.30].

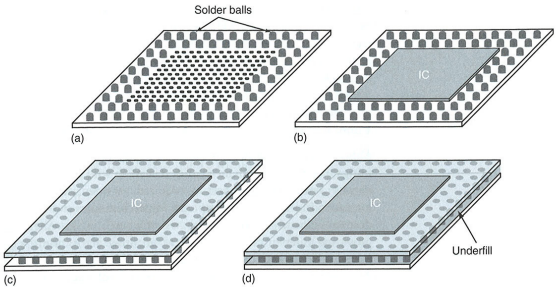


Figure 2.11: Basic manufacturing steps package-on-package bonding. (a) The interposer plane connects the bumps (in the middle) with the solder balls (at the edge) via horizontal wires; (b) Die attached to the interposer; (c) Two planes stacked; (d) Gap between the planes filled with epoxy filling. [2]

2.2.1.5 Micro bumps

The bumps used in the Package-On-Package approach were scaled down to the current micro bumps, and the interposer plane was removed. The micro bumps were directly attached to tier (see Figure 2.13(a)), due to the removal of the interposer. Bonding these bumps is done by eutectic bonding or via flip-chip-based technology. With Eutectic bonding, two tiers with micro bumps are bonded together by a layer of tin (Sn), and therefore this approach is also known as Solid-Liquid InterDiffusion (SLID) bonding [28]. The bumps are joined together by applying

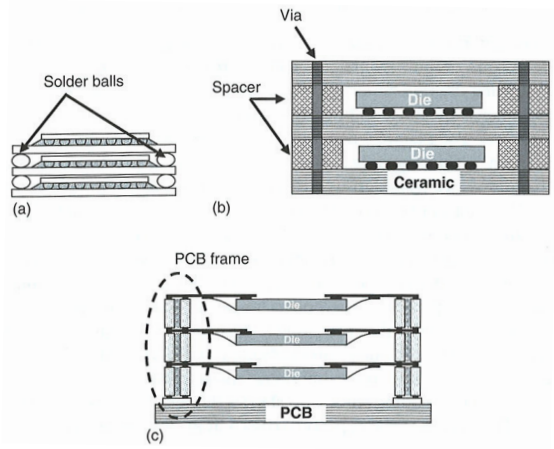


Figure 2.12: Diverse package-on-package bonding techniques. (a) Solder balls, (b) The Ball Grid Array (BGA) of the dies connects to the interposer, which contains horizontal wires and is connected to the vertical through hole vias (at chip level), (c) Through hole via at PCB level. [2]

heat and pressure (see Figure 2.13(b)) [2, p.132]. The thermal heat melts the tin between the two micro bumps, while a downwards pressure is applied. The flip-chip-based technology uses a tier with solder balls. The tier is then flipped and, electrically and mechanically, bonded to another tier.

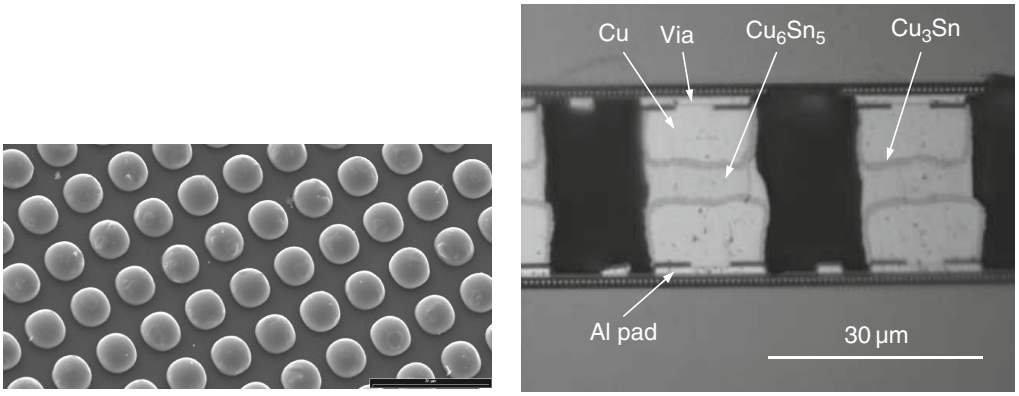
The Injection Molded Solder (IMS) technique is a method, used by IBM, to create the flip-chip-based bumps. The basic idea is that the bumps are created in a mold. The whole mold is aligned and the bumps are transferred onto a wafer [29]. The bumps are created by injecting molten solder, into etched cavities of a glass mold. (The wafer and the glass mold have similar thermal expansion ratios, to prevent miss alignment.) Eventually, the filled mold and the wafer are aligned and brought together in a heated environment where the bumps melt and attach themselves to the wafer. The mold could be reused after the transfer of the bumps, and the wafer can be stored indefinitely, at room temperature, until it is needed.

Optionally, a non-conductive "underfill" (adhesive) is placed before bonding on the substrate of the wafer. This provides extra strength and an airtight seal when the wafer is bonded. Joining two tiers together could be done by bump-up or bump-down presentation.

With eutectic bonding and the flip-chip-based technology, a downward force slightly spread the bumps at a temperature of $\approx$200-400 degrees Celsius [2, p.52]. It is important that the temperature should not to degrade the metal layer of the wafer. Face-to-face bonding is preferred, because the distance between the metal layer and the bumps is smaller, and no TSVs have to be manufactured. However, when three or more tiers are stacked, then TSVs are necessary to connect the micro bumps to the metal layer (which is through the silicon layer). Micro bumps ranges from 10μm-100μm in diameter and havening a pitch of 20μm-200μm (see Figure 2.13(a)) [30, slide31], [31, p.4].

2.2.1.6 Through Silicon Via

Through Silicon Via (TSV) is similar to a normal via, but a TSV penetrates (totally or partially) the silicon substrate, the active layer, and sometimes even the metal layer. TSVs provide the

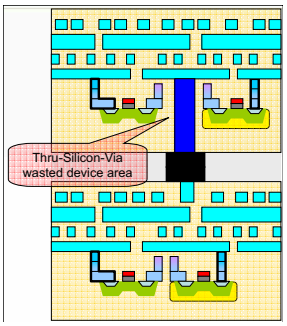


(a) Micro bumps with a diameter of 10μm. [31]

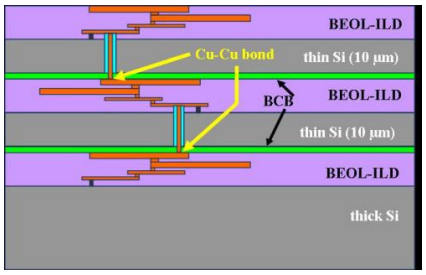
(b) Micro bump SLID [1]

Figure 2.13: Micro bumps.

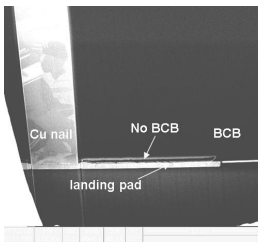
highest interconnect bandwidth (>20Ghz, [32]), compared to wire bonds, bumps and package-to-package interconnects [2, p.56]. The placement of a TSV is not limited to the edge of the chip, but can be placed at any desired position of the chip. Early implementations of TSVs had a low interconnect density of a few hundred TSVs per die, because the TSVs were deep and therefore large (due to the conical shape). By thinning the substrate of the wafer (containing the TSV), the depth became shallower and therefore smaller TSV are possible. The slope of the TSV wall determents the conical shape of the via, which limits the maximum aspect ratio (depth and diameter ratio).



(a) Back to face bonded tiers. TSV with metal-to-metal bonded bumps, occupying area in the active layer. [33]



(b) TSV nail bonded with the landing pad. To the left and right of the bond is a adhesive layer that bonds the tier. [33]



(c) TSV nail bonded with the landing pad and a benzo-cyclobutene (BCB) adhesive layer. [34]

Figure 2.14: TSV with a nail and bump approach.

After manufacturing, the TSVs are used with or without metal micro bumps. When a metal bump is used, both tiers contains a metal bump and these bumps can be bonded together with a metal-to-metal or a eutectic bond (see Figure 2.14(a)). Bonding approaches are explained in Appendix C.2. When no bump is used, the TSV is exposed at the bottom of the substrate. This is fabricated by etching the bottom of the substrate, which makes the TSV stand out like a nail (see Figure 2.14(b) and Figure 2.14(c)). An exposed TSV (nail) is bonded, with a metal-to-metal

(cu-to-cu) or a eutectic bond method, to a landing pad on the other tier. The TSV is etched through the silicon and therefore it also occupies space in the active layer. Methods, such as full laser drilling, drills a TSV hole through the whole tier (including the substrate). This occupies some area in the active layer and the metal (wire) layer of the tier, which is disadvantageous. The diameter of the TSV is thus extra important.

TSV diameters of 1μm is achieved when the wafers are thinned (to 10μm or less), and aligned within a micron [1, p.87]. A typical TSV has a depth of $\approx$10-100μm with a inductance of $\approx$0.5-1nH/mm and a capacitance of $\approx$0.0005pF/μm. The resistance is dependent on the used material. The metals and their resistance are: gold 2.5$\mu\Omega$/cm, copper 1.8$\mu\Omega$/cm, tungsten 5.6$\mu\Omega$/cm and polysilicon $\approx$190$\mu\Omega$/cm. The resistance for tungsten and copper is not large. However, a polysilicon TSV with a diameter of 3μm and a length of 50μm, would have a resistance of $\approx$10Ω. This could form a problem for some devices [1, p.86].

2.2.2 Fabrication of 3D stack structures

Although not mandatory, the author of this thesis would like to invite the reader to read appendix C, since it presents the main manufacturing methods for TSVs and the bonding methods for a 3D tier stack.

The appendix indicates at a manufacturing process the four positions where a TSV can be manufactured, the processes are called Front End Of Line (FEOL), Back End Of Line (BEOL), via-first, and via-last processes. Furthermore, it presents the method to create a TSV, which is via laser drilling or Deep Reactive Ion Etching (DRIE). Moreover, the four methods to bond tiers are presented, which are metal-to-metal, eutectic, oxide-to-oxide, and adhesive bondings. The appendix concludes by presenting the process sequences that are in uses by the industry, since many process sequences can be made with the TSV manufacturing processes and the bonding methods.

2.2.3 Inter-layer interconnects compared

This section compares the presented interconnects from Section 2.2.1 with each other in order to answer *research question two*.

The presented interconnects are the capacitive, inductive, TSV, micro bumps, Package-On-Package (POP) and wire bonded interconnects. The comparison is done in terms of pitch (interconnect density), speed, power and maturity. Table 2.1 presents the collected data, and Figure 2.15 presents the comparison between the interconnects. An estimation is done, supported by references, whenever there is no data available / found. No comparison can be done between the costs of the interconnects, because there is too little data available to make any fair comparison (see Table 2.1). The references, for all the data in Table 2.1, is indicated at the company row. The only exceptions are the data with their own reference, then that data is from that reference.

2.2.3.1 Pitch

The pitch specifies the distance where after repetition can begin, and thus it determines the interconnect density. The diameter of the interconnect is inferior to the pitch. Therefore, only the pitch is ranked. The interconnects from Table 2.1 are ranked in Figure 2.15 by the pitch order. Table 2.1 shows that the POP has a very large pitch, compared to the other interconnects.

The interposers are connected to other interposers via large solder balls, at the POP approach. It is remarkable that even though POP has a lower interconnect density, compared to the wire bond interconnection, it still replaced the denser wire bond interconnection. This is because POP has a better manufacturing yield and the wire bonds have a large parasitic capacitance. This should be taken into account when looking at these pitch ranking positions.

2.2.3.2 Power

The consumed (dynamic) power at the interconnects is data dependent and is related with the parasitic capacitance. If the parasitic capacitance is high, then it needs more power / charge to recharge and discharge the interconnect (capacitor). It is known that TSVs and micro bumps are small, and therefore consume less dynamic power [35], [36]. Furthermore, micro bumps and TSVs are being used together when bonding three or more tiers. Therefore, both are ranked at the first place and no distinction is made between them. Wire bonds, POP and capacitive interconnects have higher parasitic capacitances [2, p.25] than TSVs and micro bumps. Therefore, they both rank under the TSVs and the micro bumps. No distinction can be made between them due to the lack of data. However, they are placed above the inductive coupled interconnects, because it is known that the main problem (of inductive coupled interconnect) is the power consumption [2, p.55]. Therefore, the inductive coupled interconnect is ranked last and the wire bonds, POP and capacitive interconnects are ranked at the second position.

2.2.3.3 Speed

All data in the speed column (of Table 2.1) is converted to a standard unit (bits/seconds), because the original data was defined as baud rate and others as bit rate. The baud rate, by definition, is the number of times the signal in a communication channel changes state, per time unit. The bit rate, can be defined as the number of bits that are conveyed by the signal, per time unit. The main difference is that one signal change can transmit one or multiple bits, depending of the used modulation technique. It is assumed that one signal transition is sending one bit. Therefore, the baud rate is considered to be equally to the bit rate.

A micro bump and a TSV were connected to each other and speed tests were conducted on it, in [32]. The result of the speed test is, therefore, used at both positions in Table 2.1. The diameter of a TSV is smaller, compared to a micro bump. However, a TSV is longer. Hence, it is not possible to make any assumptions who has the most parasitic capacitance and higher speed. Thus, they are both ranked at the same position in Figure 2.15. Wire bonds and POP interconnects are bigger (more parasitic capacitance) than TSVs and micro bumps. Therefore, the wire bonds and POP interconnects are ranked lower. However, it is unknown if the parasitic capacitance is more or less than the inductive and capacitive interconnects. That is why the wire bonds and POP are not ranked.

2.2.3.4 Manufacture maturity

The manufacturing maturity is ranked in Figure 2.15 and it indicates if the techniques are already well known by manufacturers, or if there is still a lot of research and development being done. 3D integration began with wire bonded chips and wire bond techniques are well known by the industry. Therefore, it is place at the first position. However, the wire bond technique has been

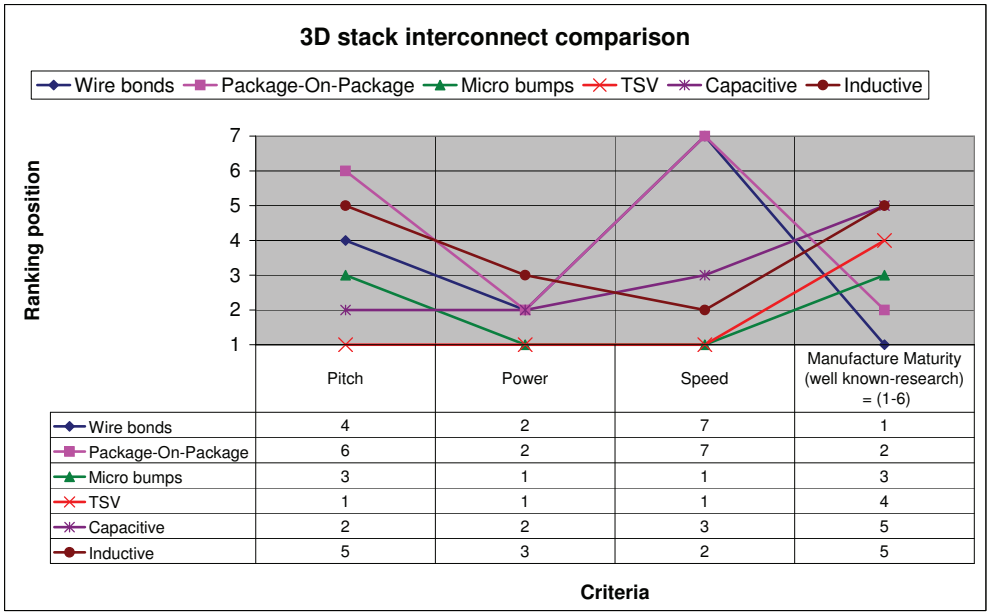
replaced by the Package-On-Package (POP) technique [2, p.25], due to better manufacturing yield of the POP. Furthermore, POP interconnects uses a Ball Grid Array (BGA), which are also well known by the industry. Hence, it is ranked under the wire bond technique. The POP technique is replaced by the micro bump technique by removing the interposer and reducing the size of the BGA into micro bumps. Micro bumps use the same techniques as the flip-chip technique, and is thus considered to be well known. That is why the micro bump technique is ranked under the POP technique. Currently, the industry gives the TSVs a lot of attention [37, p.41]. The consortia are focused on developing affordable manufacturing techniques for the TSV interconnects (see Appendix D). At the same time, the capacitive and inductive interconnects are being developed and some claim ([3,25,38–40, for capacitive], [4,25,26, for inductive]) that the capacitive / inductive interconnects are better. However, the International Technology Roadmap for Semiconductors (ITRS) indicates that the TSV has received a lot of attention by the industry and provides a road map for the TSV interconnects and not for the other interconnects [37, p.44-p.47]. Thus more research and development is done by the industry on TSVs. Therefore, more information is known about TSVs, than about the capacitive and inductive techniques. Thus, the TSVs are ranked above the capacitive and inductive techniques. Between the capacitive and the inductive techniques, there are no hard indications, which of them is better known. Therefore, the capacitive and the inductive interconnects are ranked at the same and last position.

2.2.3.5 Overall ranking

Figure 2.15 indicates that the TSVs and the micro bumps have good properties. However, a micro bump can only provide inter-layer connection up to two layers. Conversely, a TSV can provide *unlimited inter-layer connection*. Moreover, the *TSV is ranked the most times at the first place* (three times), in respect of the other interconnects. The micro bump is ranked on the second best position, since it is ranked twice at the first place. Only at the maturity stage, the micro bump and TSV are ranked at the third and fourth place, respectively. That is because both techniques are still in the developments and research stage. However, there are already devices produced with TSVs, such as a 3D TSV DRAM memory by Elpida [41].

Thus, the answer on research question two is found in this section, the TSV is the best interconnect. This is re-affirmed by the next section, Section 2.2.4, which shows that four consortia are doing research on TSVs and not on other interconnects. The answer on research question two is:

> Research question two: Which inter-layer interconnect is the best to use with the best structure?
>
> *A TSV is the best interconnect to use between multiple tiers, since it can stack an unlimited amount of tiers and it is the best in terms of pitch, (dynamic) power, and speed. Furthermore, it is expect that the maturity of the TSV improves, since many companies and institutions are doing research on it.*



	Pitch	Power	Speed	Manufacture Maturity (well known-research) = (1-6)
Wire bonds	4	2	7	1
Package-On-Package	6	2	7	2
Micro bumps	3	1	1	3
TSV	1	1	1	4
Capacitive	2	2	3	5
Inductive	5	3	2	5

Figure 2.15: Ranking of 3D stacked interconnects from table 2.1. Ranking is done from one to six, which denotes the best and worst ranking position, respectively. Ranking number seven denotes that there is no ranking possible, due to missing of data. Interconnects with the same ranking position are seen as equally good.

2.2.4 Consortia

Appendix D shows that TSVs have a high potential to become the 3D interconnect for stacked devices.

The appendix discusses the five main institutes that do research into 3D integration. These are: ITRS, Sematech, 3D-IC alliance, EMC-3D and 3D-TSV. The ITRS made a 3D TSV road map [46, p.38] and the other four (individual) institutes are all focusing on developing methods for 3D stacking with TSVs as interconnect (sometimes in combination with a micro bump). Hence, it indicates that TSV has high potential to become the 3D interconnect for stacked devices.

2.2.5 Properties of TSVs

Some of the properties of Through Silicon Vias (TSVs) are already presented in the previous sections. *This section zooms in on the latency, area, power, yield, and reliability of the TSVs*, and thus research question three is answered where the 'best inter-layer interconnect' referrers to the TSVs.

> Research question three:
>
> *What are the properties of the best inter-layer interconnect?*

Table 2.1: Properties of 3D stacked interconnects. The references for the data are indicated between the square brackets.

	Wire bonds	Package on package	Micro bumps	TSV	Capacitive	Inductive
Diameter ϕ (μm)[a]	>12.5 [42]	250	10	1-5	8 (8*8)	100
Pitch (μm)	40	400	20	7-10	10	120
(Dynamic) Power (μW)[a]	-	-	-	-	112 (80μW/Gbps)	15000
Speed (Gb/sec)[a]	-	-	>20 (>20Ghz) [32]	>20 (>20Ghz) [32]	1.4 (22Mbps/μm^2)	5 (5Gb/sec)
Cost (USD/wafer)	-	-	-	<150 [43]	-	-
Company	ChipPAC Inc. [44]	ChipPAC Inc. [44]	IMEC [30]	IMEC [45]	University Bologna, Italy [3]	US airforce, North Carolina University [26]

[a]The data is converter to the same time unit, for comparison. The raw data is presented between the parentheses.

2.2.5.1 Latency

This section shows that the delays (per unit length) for TSVs and 2D global wires are similar. However, a TSV is much shorter than a 2D global wire, and thus the TSV latency is negligible.

TSVs can be used in bundles to form a bus, see Figure 2.16(a)) [47] where a 3x3 TSV bundle is presented. However, adjacent TSVs influence the capacitance of each other, due to coupling effects (see Figures 2.16(b) and 2.16(c)). The TSV coupling effects are because a single TSV is similar to a capacitor plane, and two adjacent TSVs (planes) form a whole capacitor. The distance between the two capacitor planes is formed by the pitch and the material in between (SOI or bulk) acts as the dielectricum. Figure 2.16(c) depicts the capacitance of TSVs where the pitch is swept and the diameter (4μm x 4μm) is kept fixed. It shows that the capacitance decreases as the pitch increases, due to the coupling effects. Furthermore, the capacitance is also influenced by the diameter of the TSV, see Figure 2.16(b). The capacitance increases when the TSV diameter grows, it is because the cross-section of the TSV is getting larger and area between the pillars is getting smaller (more coupling effects). However, the *delay (per unit length) is for TSVs and 2D global wires similar*, even for TSVs which are penetrating through 20 layers ($\approx$200-800μm) [6, 47, 48] and [5, p.117]. It is because the capacitance and the resistance of a TSV cancel each other out (see Figure 2.17(a)). The capacitance of dense TSVs is much higher (due to coupling effects), but the resistances is (per unit length) much smaller (50 times) compared to a typical 2D global interconnect [47, 48]. Although not discussed by [47, 48] why the resistance is smaller, the author of this thesis thinks it is because the length of a TSV is much shorter, and the diameter is in general larger, compared to a 2D wire. However, a TSV has a larger capacitance, due to the larger diameter and the coupling effect. Thus, the resistance and capacitance cancel each other out at the RC-delay of a TSV.

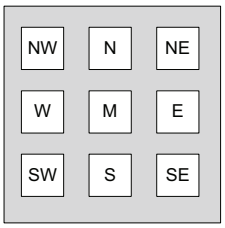
Conclusively, a *TSV has similar delay properties as a 2D (global) wire*, and thus minor (positive or negative) effect on the interconnect latency per unit length, see Figure 2.17(a). However, a *TSV is much shorter* than a global 2D interconnect and thus is a *TSV delay negligible* [5, p.136], compared to a 2D global wire.

For example, a TSV with a diameter of 5μm, a length of 10μm (connecting two layers), and produced in bulk technology has a capacitance of 0.5-0.6 fF per μm (see Figure 2.16(b)). This capacitance gives a delay of less than 0.1ps [5, p.135]. The capacitance of a 2D inter-Functional Unit Block (FUB) interconnect with a length of 1mm gives a latency of 100ps , see Figure 2.17(b). Thus, the 2D (global) wire of 1mm has a delay of 100ps, which is 1000 times larger compared to a TSV of 10μm (0.1ps). This shows that the delay of a TSV is negligible compared to a global (1mm) wire and thus the success of 3D integration lays in *wire length reduction* [5, p.136].

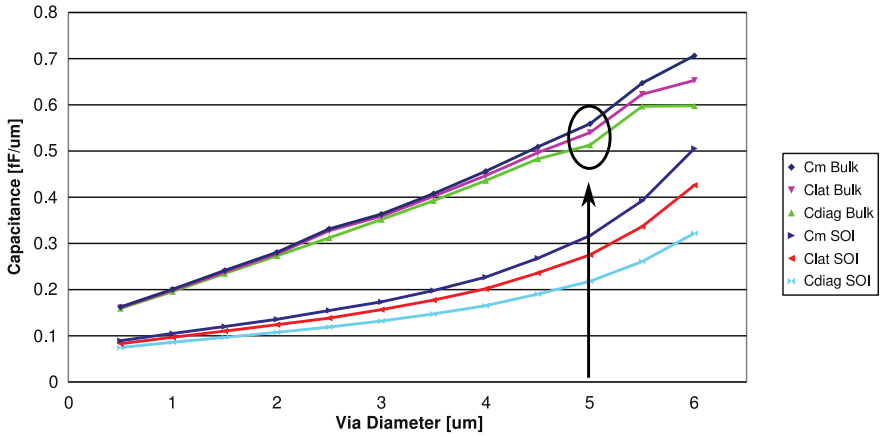
2.2.5.2 Area

This section compares the area of a TSV pitch against the area of CMOS gates. The known TSV pitch sizes were previously presented in Table 2.1, the smallest known TSV pitch size is 7μm.

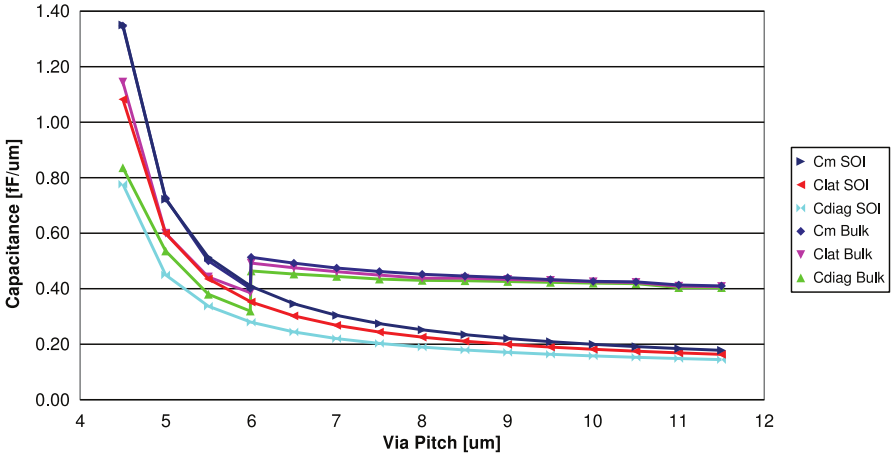
The number of TSVs in a design depends on the area a TSV pitch occupies. Designs where excessive TSVs are used are highly dependent on the area a TSV occupy, such as designs where each gate is connected to another gate via TSVs. The pitch of a TSV consists out of the TSV diameter plus the stay-out area. CMOS gates that are manufactured with 45nm process technology have a gate area of $\approx$1.5μm x 1.5μm [20, p.5]. In table 2.2, the area of one TSV pitch



(a) TSV pillar layout for graphs (b) and (c). The TSVs have a cross-section of 4μm x 4μm.

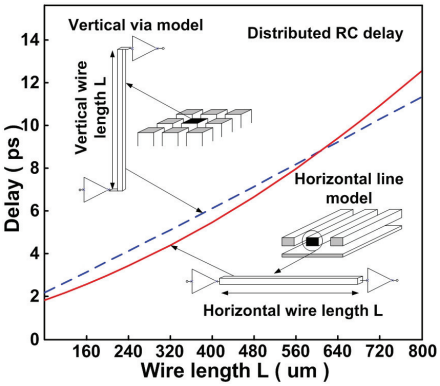


(b) The capacitance of a TSV per unit length (fF/μm), with fixed pitch (8μm) and a variable diameter. The wire delay of a TSV with a diameter of 5μm and a length of 10μm is <0.1ps. [48]

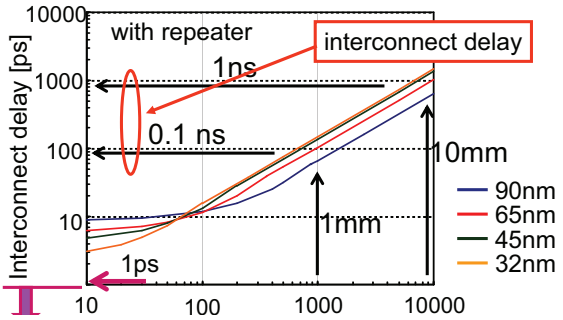


(c) Capacitance of a TSV per unit length (μm), with variable pitch and a fixed diameter (4μm plus 1μm insulator for the bulk device), bundled in a three times three array. [48]

Figure 2.16: The lines in graph (B) and (C) indicates the capacitance of TSVs with bulk and SOI technology bundled in a 3x3 array with the same layout to figure (A). Where Cm=M pillar, Clat=N=S=W=E pillars and Cdiag=SW=NW=NE=SE pillars.



(a) A horizontal global interconnect and vertical bus (TSVs) of an equivalent length are compared. The longest length shown here (800μm) corresponds to a long TSV spanning 20 layers. [47]



(b) The delay and length are plotted for 2D wires with divers 2D process technologies, with on the Y-ax the delay (in ps) and on the X-ax the wire length (in μm). A TSV with a wire length of 10μm has a delay of <0.1 ps. This delay is negligible compared to a 2D wire. [5]

Figure 2.17: The delay and wire length are plotted. (reference for Figure 2.17(b) [5, p.136])

is compared against the area of CMOS gates its occupies[1]. Thus, it indicates if a design has a lot of area overhead and if it is practical. Table 2.2 is calculated via Equation (2.1), where n_CMOS_gates denotes the number of CMOS gates that the TSV pitch occupies. In general, *vertical routing should be restrained* [20, 49]. Naturally, the use of TSVs is a *trade-off between area (cost) and performance*.

$$n\_CMOS\_gates = \frac{area\_TSV\_pitch}{gate\_area} = \frac{(\frac{1}{2} \cdot TSV\_pitch)^2 \cdot \pi}{1.5 \cdot 1.5} \tag{2.1}$$

Table 2.2: The pitch of a TSV occupies n times the area of a CMOS gate. The CMOS gates are manufactured with a 45nm process technology, and they have a gate area of ≈1.5μm x 1.5μm [20].

TSV pitch (μm)	1	2	3	4	5	6	7	8	9	10
Occupies n CMOS gates	0,3	1,4	3,1	5,6	8,7	12,6	17,1	22,3	28,3	34,9

2.2.5.3 Power

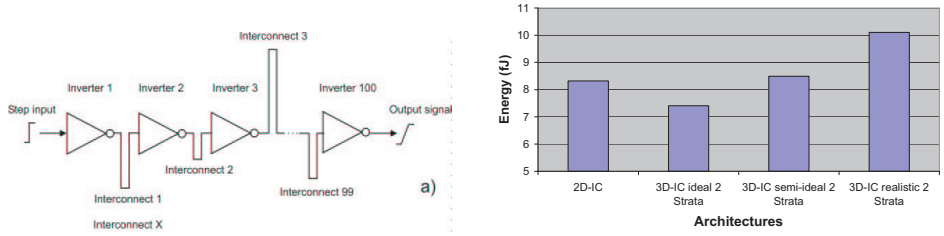
Two types of articles are compared in this section, one article ([6]) claims that TSVs consume more power, and the other articles claim that TSVs reduce power, compared to 2D on-chip wires. Eventually, this section shows that TSVs reduce power when long 2D on-chip wires are replaced by short TSVs [49, sl.9].

The power consumption of 2D wires are compared against the power usages of TSVs by [6]. Three different TSVs with three different pitches are used, and they represent the ideal, semi-ideal, and the realistic pitch cases, and they have a pitch of 1,5μm, 3μm, and 15μm, respectively.

[1]The pitch sizes are hypothetical and it is only presented to show the effect of a certain pitch size.

Even though a TSV with a pitch of 15μm is called realistic by [6], it is superseded. In the current situation (early 2010) is a TSV with a pitch of 7μm a better realistic case.

For the experimental setup, [6] has taken 100 inverters, connected via 100 wires with random wire lengths (see Figure 2.18(a)). The ideal 3D-IC case offers an energy reduction of ≈10% (see Figure 2.18(b)). However, the results of the realistic and semi-ideal case depict that the power consumption is worse, compared to the 2D on-chip wire. Conversely, other Articles [9,10,50–52] indicated that the use of TSVs gave power reductions (in general) between 5% and 20%. The main reason between the dissimilar results is that [6] used random wire lengths (also short wires), while the others used mostly (long) wire dominated circuits, which gave more power reduction than short wires. Thus, in general the *power reduction is achieved when the long 2D wires are replaced by short TSVs* [49, sl.9].



(a) Schematic diagram of a 100-inverter chain, using different wire lengths from the wire-length distribution.

(b) Switching energy comparison for 100-inverters between 2D wires and TSVs. The 3D has circuit has two layers. The TSVs have a pitch of 1,5μm, 3μm, and 15μm and are indicated by the labels 3D-ic ideal, semi-ideal, and realistic, respectively.

Figure 2.18: Energy comparison between 2D and 3D interconnects for a chain of 100 interconnected inverters. [6]

2.2.5.4 Manufacturing yield

Currently (early summer 2010), TSV fabrication processes have a relatively low yield, compared to standard 2D processes [53]. This section presents two possibilities to improve the yield, the yield is improved via redundant TSVs or via Time Division Multiplexing (TDM).

Faulty TSVs can have many causes and occur during fabrication time or even at normal operation. When the fault probability of one TSV during stacking only slightly decreases, especially when a chip contains 10.000 TSVs, then the overall yield dramatically decreases. As shown in Figure 2.19, the yield of a 3D stack with 10.000 TSVs rapidly converges towards zero if the fault probability of a single TSV slightly increases.

During the manufacturing and bonding of a tier defects can occur. These defects comprise a variety of unpredictable physical phenomena's during the thermal compression bonding process at wafer stacking. The phenomena's involve dislocations, oxygen trapped on the surface, void formation, wrongly etched, filled, or bonded. Figure 2.20(a) shows limited yield at bonding two strata with three different process technologies from Honda Research Institute (HRI), IMEC and IBM. The yield is evaluated with the use of the Poisson distribution and only complete or partial open defects are considered. Only complete or partial open defects are considered, due to

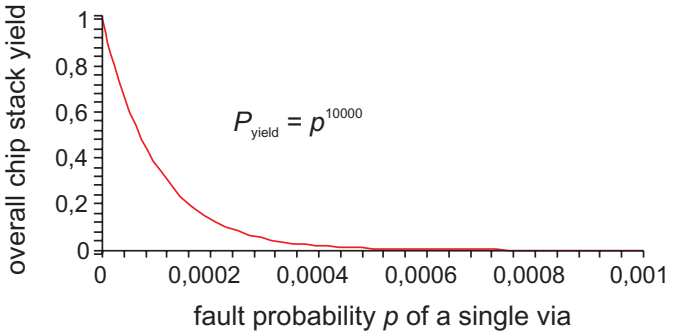


Figure 2.19: Chip stack yield. Ten thousand TSVs (with all a small fault probability) decrease significantly the overall yield when they are combined. [7]

(minor) misalignments [53]. Misalignment is caused by shifts of bonding pads with respect to their nominal positions during the bonding process. Extreme misalignments are highly unlikely in state-of-the-art wafer bonding processes [53]. However, when misalignments occur then it has a minor impact on the delay, and it does not influence the functionality. Figure 2.20(a) shows that at the point of 1.000.000 TSVs (for HRI) per chip the *yield converges to zero due to the immature manufacturing and bonding processes*.

HRI [54] published in 2009 five experiments, shown in Figure 2.20(b). In these experiments a microprocessor layer, a custom analog circuit layer and a 64-Mbit SDRAM layer are bonded. In between the layers there are respectively 655.584 and 180.160 TSVs / wafer (8-inch wafer) and it is manufactured with 0.18μm process technology. The TSVs are manufactured with a traditional DRIE process in FEOL and BEOL processes. Metal-to-metal bonding is used in combination with an adhesive injection and a TSV nail is directly bonded with a micro bump [54]. Unfortunately, the diameter and pitch of the TSVs and micro bumps are not indicated in [54,55]. The best *obtained yield is 68.4%*, which is not desirable for mass production and confirms / indicates that the *current manufacturing technology has yet to mature*.

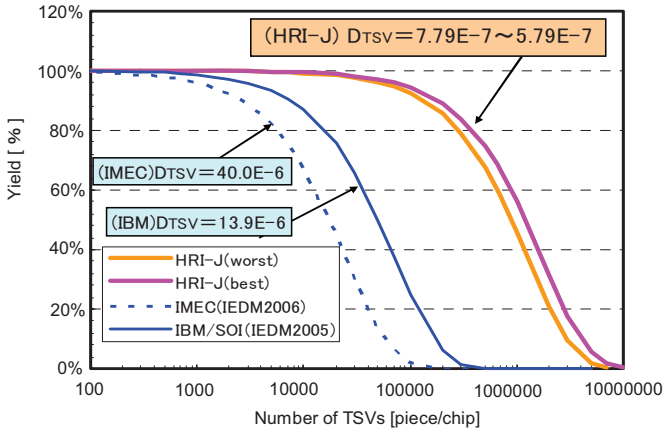
The following two sections show that this low *yield can be improved*, via a *redundant TSV* or via a *Time Division Multiplexing scheme*.

2.2.5.5 Improving yield with redundant TSVs

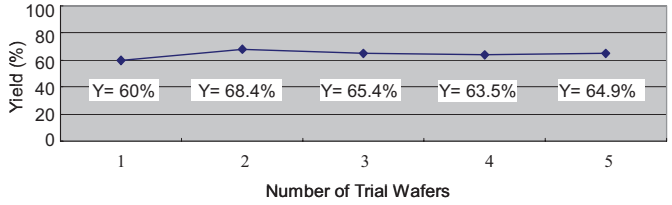
In this section the yield is improved by the use of redundant TSVs. Implementing redundant TSVs can be done with a twin TSV pillar or by implementing an extra (single) TSV.

Twin pillars, shown in Figure 2.21(a), are briefly mentioned by [53, 56]. Unfortunately, no further information is found about the twin pillar. However, it is interesting to discuss what will happen if one leg of the pillar is stuck-at open (not connected), because one leg (50%) of the twin pillar is gone which doubles the resistance while having the same capacitance. That would lead to a speed reduction. Another interesting question is why two narrow pillars are preferred compared to a larger pillar. A larger pillar requires less etching steps and needs a smaller aspect ratio (depth and diameter ratio). A smaller aspect ratio would be easier to build and is thus more reliable. Unfortunately, no information is known about these questions.

Alternatively, it is possible to implement a spare TSV for every cluster of TSVs (see Figure 2.21(b)). Whenever one of the regular TSVs is faulty a new routing path is set via the



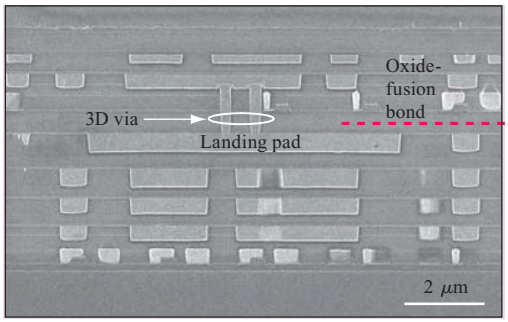
(a) Yield expectation from Honda Research Institute (HRI) 2009 for binding two strata. Two other processes from IBM and IMEC are depicted for comparison. Yield is evaluated using the Poisson distribution [54].



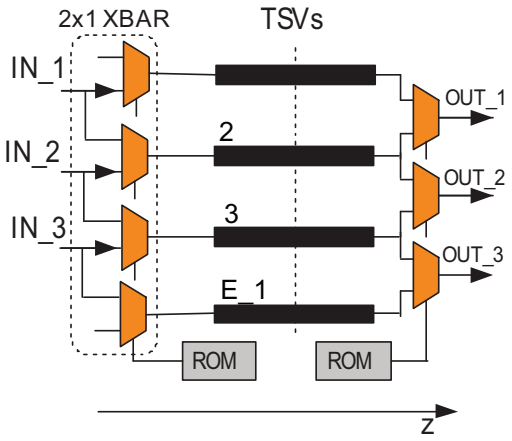
(b) Yield for three layers (a microprocessor layer, custom circuits including analog circuits layer, and a 64 Mbit SDRAM layer) with in between 655.584 and 180.160 TSVs / wafer. Bonded in five different trials. [54]

Figure 2.20: Yield expectations during fabrication time.

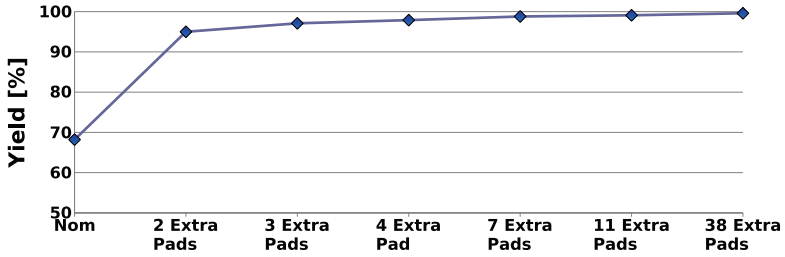
reconfigurable multiplexers. For example, if TSV number 2 is faulty then all the lower signals are rerouted one TSV down (see Figure 2.21(b)), thus signal IN_1 remains routed over TSV 1 and IN_3 is routed over TSV E_1). The output signals do not notice the presence of the diversion path due to the correction at the reconfigurable multiplexers at the output side. As prove of work a Network-On-Chip (NOC) architecture was modified, extra unidirectional TSVs are added in a Xpipes switch. The original Xpipes switch needs 38 TSVs for normal operation, extra redundant TSVs are added (2,3,4,7,11 and 38 extra) for the whole switch. Figure 2.21(c) shows the result from the emulation at [53], where a fixed defect frequency of 9.75 Defects Per Million Opportunities (DPMO) is assumed, and a design with 4.2M TSVs is analyzed. The *original yield (68%) improved to 98%* with only four extra TSVs per 38 signals, and these TSVs only give an area overhead of 2.1% for 130nm technology and 3.8% for 65nm technology (compared to a design with no redundant TSVs). Using more than seven spare TSVs brings only minimal yield benefits and it gives a 10.5% area overhead (compared to a design with no redundant TSVs), which shows that 38 redundant TSVs, a full redundant scheme, is unnecessary. This is the main advantage of this redundant routing scheme. It uses 'n' spare TSVs for every cluster of TSVs resulting in a *small area overhead* and *significant yield improvement*.



(a) Twin pillar. [56]



(b) One extra TSV for every three TSVs. [53]



(c) Yield improvement over seven different hardware configurations: no-redundancy, 2, 3, 4, 7, 11 and 38 extra pads, which respectively results in a total of 38, 40, 41, 42, 45, 49, and 76 TSVs. [53]

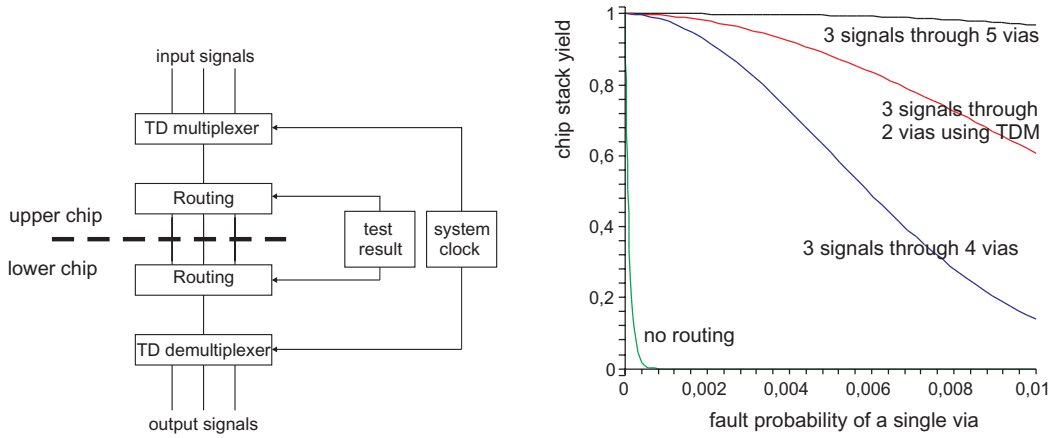
Figure 2.21: Two redundant TSV options to improve the yield.

2.2.5.6 Improving yield with Time Division Multiplexing schemes

In this section the yield is improved by sharing one TSV with multiple signals.

A Time Division Multiplexing (TDM) scheme shares one TSV for two or more signals, and therefore no extra redundant TSVs are needed (see Figure 2.22(a)). The major difference between a redundant TSV scheme and the TDM scheme is that a TDM scheme works on a global clock, and the TDM routing logic needs more area, compared to a redundant TSV scheme [7]. The clock frequency has to be chosen high, compared to the input signals, to minimize the delay caused by TDM. The advantage of TDM is that without the uses of redundant TSVs the yield is improved, while it improves the reliability. However, sharing one TSV with multiple signals increases the delay. Naturally, it remains a tradeoff between delay and area.

Figure 2.22(b) compares the redundant TSV scheme against the TDM scheme. Two yield lines are calculated for one and two redundant TSVs, and one yield line is calculated for the TDM scheme. The yield of a *TDM scheme is better than one redundant TSV*, but worse than having two redundant TSVs (see Figure 2.22(b)). However, the TDM scheme is *much better than without any correction scheme* (see Figure 2.22(b))



(a) TDM routing scheme. [7]

(b) TDM yield calculation of 10.000 TSVs. The TDM scheme performs better than one redundant TSV and worse than two extra TSVs [7]

Figure 2.22: TDM routing scheme and yield result.

2.2.5.7 TSV reliability

This section indicates that the reliability of a TSV is low during normal operation, but it can be improved via the same techniques that improves the yield, that is via redundant TSVs or with the TDM scheme.

Reliability problems can occur during normal operation, even when a 3D stack is correctly manufactured. The main reliability problem of TSVs during normal operation is the difference in Coefficient of Thermal Expansion (CTE). The CTE indicates how the size of an object changes with a change in temperature. Specifically, it measures the fractional change in volume (size) per degree (Celsius) at a constant pressure. The risk of a crack increases whenever two bonded layers have a different CTE due to mismatch of size expansion. To test CTE mismatches in a design / device a Pressure Cooker Test (PCT) and a Temperature Cycling Test (TCT) is applied. The PCT test puts the device in a chamber at 121°C / 100% relative humidity with 30psi for $\approx$168 hours and afterwards the device is activated and tested [57]. The TCT test heats up the device and then cools it down again (-55°C until 125°C), in the next cycle the device is heated up again and so on [57].

We will look at an example from [57] that uses first precondition tests (to eliminate the infant mortalities), and thereafter the TCT test is used to investigate the characteristic life of TSVs. 10 layers are bonded, and electrically connected to each other with TSVs of 100μm (diameter). The TSVs are manufactured with laser ablation, and the layers (TSVs) are bonded via eutectic bonding (soldering) without an under filling [57]. Then precondition tests are conducted to detect open or short TSVs, where the samples are put in a temperature and humidity regulated test chamber at 30°C / 60% relative humidity for 192 hr, followed by three cycles with peak temperatures of 260°C. 30 samples passed the pre-condition tests, and these good samples are tested with the TCT (-55°C-125°C) and lasted 2500 cycles. From the results [57] estimated that TSVs have a *characteristic life of 1216 cycles* and good TSV can even handle 2500 cycles. This is not long for a (general) lifetime expectation of device, other measures can be taken to improve

the reliability and yield.

Whenever a TSV is faulty during or after fabrication time, one could throw the faulty 3D chips away or cope with the faulty TSVs. Throwing faulty chips away, especially with immature fabrication techniques, is costly and therefore unacceptable. The reliability can be improved if the hardware architecture can cope with faulty TSVs, such as the use of redundant TSVs or with TDM schemes from Section 2.2.5.4. No data is found, which impact these schemes have on the reliability of TSVs. However, the author of this thesis expects that this improves the reliability a lot, such as it did for the yield.

This section, Section 2.2.5, discussed the latency, area, power, yield, and reliability of the TSVs. Thus, research question three is answered.

Research question three: What are the properties of the best inter-layer interconnect?

- *The latency of a TSV is similar to a global 2D on-chip wire, but TSVs are much shorter. Thus, the latency of a TSV can be neglected.*
- *The area of a TSV occupies depends on the pitch, the (current) smallest known pitch is 7μm and the area is similar to 17 CMOS gates produced in 45nm technology.*
- *Power reduction is possible when long 2D on-chip wires are replaced by short TSVs. Articles are known that reduced the power consumption by 5%-20%.*
- *The current manufacturing and bonding yield of TSVs are low (68%), but the yield can be improved with redundant TSVs (98%) or with the Time Division Multiplexing scheme*[1].
- *The current characteristic life is low (1216 cycles), but it can be improved with redundant TSVs*[1] *or with the Time Division Multiplexing scheme*[1].

[1] No exact numbers are known.

Architectural potential and impact

3

3.1 Basic architectural considerations

In this chapter research question four is answered. Each topic of the research question (memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC) is answered in a separate section, and all those answers are concatenated at the end of the thesis in Section 5.2, which forms the overall answer on research question four.

> Research question four:
>
> *What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC?*

3D stacking includes five key advantages: (1) wider and denser (on-chip) interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies [8], and (5) footprint reduction. The major interconnect bottleneck is solved by the wider / denser (on-chip) interconnects and by the wire length reduction [24]. Wire length reduction is obtained by strategically stacking circuits / cores on top of each other, and that results in a simultaneous reduction of wire latency and power consumption. Furthermore, the wire length reduction makes (long) pipelines to cross a core or a chip unnecessary, which leads to a lower pipeline (startup) latency [8, p.42]. The heterogeneous device technology enables the uses of different technologies per layer, such as different substrate materials (germanium, silicon or Silicon-On-Insulator (SOI)), or with the same substrate material but with different process technologies (nano meter). Furthermore, 3D integration provides a reduced footprint area, which is especially beneficial to the hand-held market. These key advantages are discussed in this chapter.

The next two sections, Sections 3.1.1 and 3.1.2, present basic architectural knowledge. In particular, they present four main strategies for stacking a 3D design, and it presents a rule of thumb that indicates the wire length reduction per layer. This knowledge is needed to understand the remaining sections of this chapter. In this chapter the term *layer* is used instead of tier, since in (most of) the architectural articles this term 'layer' also is used. The author of this thesis thinks it is because a tier referrers to a wafer or a die, but on architectural level there is no need to refer to a wafer or die. The only goal on architectural level is to discuss the different layers.

Overview tables are made of the articles that are discussed in this chapter, and other interesting articles are also places in that overview. The overview tables are presented in Appendix H where the articles are categorized according to the four stacking strategies. Furthermore, In Appendix I the same articles are presented, but a more detail overview is given, compared to appendix H. In particular, the simulation tools that are used by those articles are shown. In general, a *circuit simulator* is used to determine the new frequency and a second simulator is

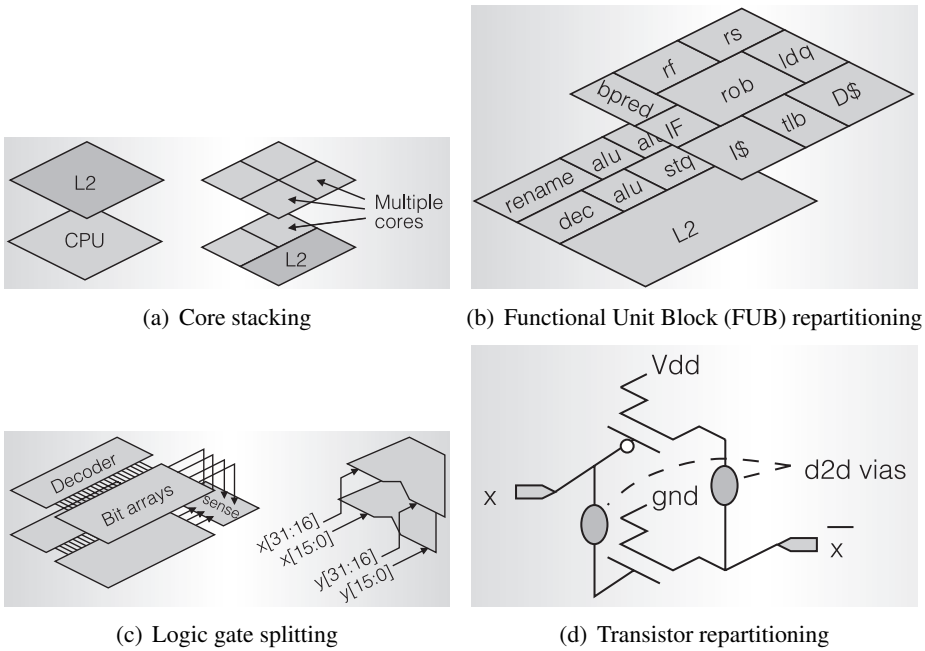
Table 3.1: The stacking strategies are ranked per property. Number '1' indicate the best property and number '4' the worst.

	Inter-FUB wire reduction	Intra-FUB wire reduction	Redesign effort
Core stacking	4	-	1
FUB repartitioning	3	-	2
Logic gate splitting	2	2	3
Transistor repartitioning	1	1	4

used to determine the overall impact on a Chip Multi-Processor (CMP). *Hspice and Spice are commonly used* as circuit simulators. The *SimpleScalar is commonly used* to measure the overall performance in a system. Furthermore, the (only) thermal simulator used by the articles is *Hotspot*.

3.1.1 Strategies

This section presents four 3D stacking strategies. This knowledge is needed in the remainder of this thesis to comprehend the design considerations and potentials. The four main strategies are presented from the coarsest to the finest level: *(1) Core stacking , (2) Functional Unit Block (FUB) repartitioning , (3) Logic gate splitting, and (4) Transistor repartitioning* (see Figure 3.1) [8, p.36]. The benefits of the four strategies are ranked on redesign efforts, inter-FUB and intra-FUB wire reductions, see Table 3.1. These strategies and benefits are discussed in the next paragraphs.



(a) Core stacking

(b) Functional Unit Block (FUB) repartitioning

(c) Logic gate splitting

(d) Transistor repartitioning

Figure 3.1: Four main design approaches ranging from coarse to fine grained. [8]

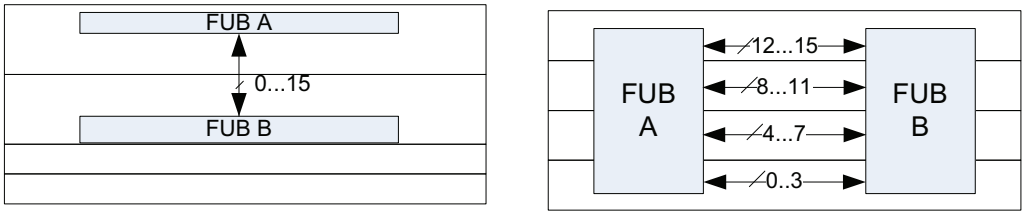
The first strategy, *core stacking*, stacks two or more cores on top of each other. The stacked cores are identical to its 2D version (see Figure 3.1(a)). The advantage is that the redesign efforts for 2D cores are low and thus the reuse factor is high. Furthermore, it *reduces the overall foot print* of the chip. However, the disadvantage is that it gives low wire length reduction which limits the power and performance benefits of 3D stacking [8]. An example of core stacking is a cache on top of a processor (see Figure 3.1(a)).

The second strategy, *FUB repartitioning*, is used to extract more benefits from the 3D integration (see Figure 3.1(b)). Repartitioning the FUBs provides medium wire reductions and it *reduces critical paths between two FUBs* (inter-FUB wire reduction), which leads to *simultaneous power and performance improvements*. It is because the FUBs that frequently communicate with each other can be places above or next to each other (horizontally). This leads to simultaneous power and performance improvements for communications between the FUBs. However, the redesign effort on re-floorplanning and retiming paths increases, compared to core stacking. An example of FUB repartitioning is an Arithmetic Logic Unit (ALU) on ALU. Moreover, the internal structure of a 2D FUB is not affected, and thus no internal power and performance improvements are observed.

FUB repartitioning still uses planar FUBs. Conversely, the third strategy (*logic gate splitting*) divides a single FUB over two or more layers (see Figure 3.1(c)). Naturally, with multiple layers the FUBs can also be placed above and next to each other. This approach provides a high reduction of the wire lengths, since it reduces the inter-FUB and intra-block (in the FUB) wires. That is why logic gate splitting leads to significant power and performance benefits, compared to core stacking or FUB repartitioning [8]. Logic gate splitting is *especially effective for wire dominated blocks*, such as a SRAM memory [9] or a Kogge-stone / Sklansky adders [10, 51]. However, the disadvantage is that logic gate splitting requires more redesign efforts, compared to core stacking or FUB repartitioning.

The fourth strategy is *transistor repartitioning* and it is the finest level of 3D integration (see Fig 3.1(d)). At an extreme case there is one layer devoted for NMOS gates and another layer to only PMOS gates. The TSV pitch size should be smaller than the size of a transistor to make transistor repartitioning practical. Otherwise, if the TSV pitch size is just as large as a transistor, it will take twice as much area. However, it is not likely that the TSV pitch will scale down to such a fine grain interconnect [8, p.37]. Another disadvantage is that the redesign effort is very high, compared to core stacking, FUB repartitioning and logic gate splitting. Furthermore, the wire reduction is not large (compared to a 2D circuit) due to the fact that at a 2D plane most transistors are positioned next to each other, and thus the wire length reduction is small, compared to logic gate splitting.

For all the stacking strategies, the data bus width can be placed horizontally or vertically, and is in this thesis referred to as the *single-layer data approach and the multi-layer data approach*. the single-layer data approach is illustrated by Figure 3.2(a), FUB 'A' communicates (vertically) with FUB 'B', and it can depicts a single 2D FUB that is divided over two FUBs and two layers, or it can depict two individual 2D FUBs that are stacked and communicate. The multi-layer data approach is illustrated in Figure 3.2(b), it depicts two individual FUBs that are internally divided across multiple layers. However, this design approach needs a lot of redesign. Naturally, a hybrid form of the single-layer and the multi-layer data approaches is possible.



(a) Data width is positioned vertically across the layers (single-layer data approach).

(b) Data width is uniformly and horizontally divided across each layer (the multi-layer data approach).

Figure 3.2: Data width positions of a 16 bits bus, with respect to the layers. The four layer thickness are not depicted onto scale, with respect to each other.

3.1.2 Wire length vs. the number of layers

This section shows that (in general) endless stacking does not provide endless (large) wire reduction, after the fourth layer the wire reduction becomes smaller than before the fourth layers. This knowledge is needed in the remainder of this thesis to comprehend the design considerations and potentials.

As indicated previously in this thesis, the wire length reduction is one of the main advantages of 3D integration. The *wire length reduction leads to simultaneous latency and power reductions* [58]. After a 2D plane is partitioned over multiple layers then the following *rule of thumb gives a general indication of the wire reduction length and remaining wire length* at a specific number of layers. *The rule of thumb is: the wire reduction factor is equal to the square root of the number of layers [59, p.3], and the remaining wire length is equal to the original 2D wire length divided by the wire reduction factor.* This rule of thumb is shown in the following equations. Naturally, the real wire reduction depends on the real repartitioning of the cores / FUBs. However, this rule of thumb can be used to illustrate a general trend between the number of layers and the wire reduction.

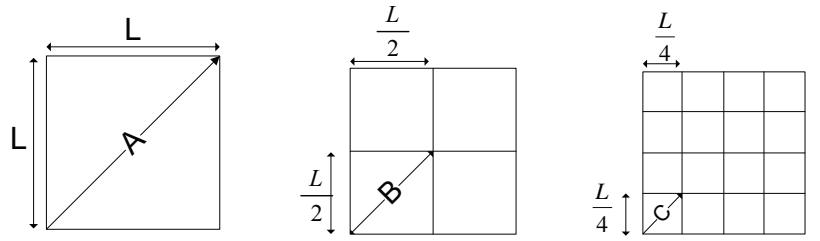
$$Wire\_reduction\_factor = \sqrt{No\_Layers} \tag{3.1}$$

The remaining wire length can be calculated as follows:

$$Remaining\_wire\_length = \frac{2D\_wire\_length}{Wire\_reduction\_factor} \tag{3.2}$$

For example, a sender and a receiver on a 2D plane are connected via wire A (see Figure 3.3(a)). This 2D chip is respectively sliced into four and sixteen parts and then stacked (see Figure 3.3(b) and Figure 3.3(c)). Assume that the sender and receiver remain at the bottom-left and upper-right corner, respectively. The remaining wire length is calculated in Table 3.2 and it shows that the rule of thumb holds.

This rule of thumb holds for wires which are routed diagonally or via the X and Y direction (see Figure 3.4). The rule also holds for diagonal wires, since they can be decomposed into a X and Y vector (this is similar to the Pythagoras theorem). The wire reduction is plotted in Figure 3.5 for all the cases of Figure 3.4, and it is compared to the rule of thumb wire reduction calculation. It is visible in Figure 3.5 that the rule of thumb holds for the 'X and Y' and the



(a) One (2D) layer and can depict a whole 2D chip, a 2D core or a 2D Functional Unit Block (FUB)

(b) The 2D layer divided over four layers

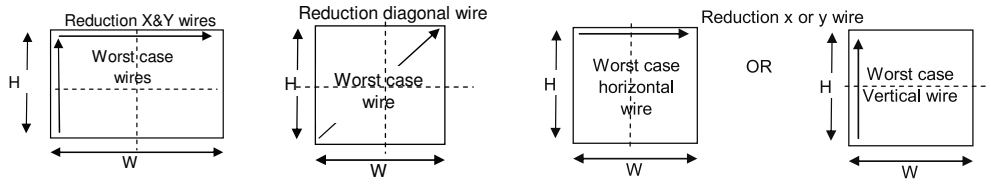
(c) The 2D layer divided over sixteen layers

Figure 3.3: A 2D plane is divided over four and 16 layers and stacked on top of each other. The remaining cross wires are indicated with the letters B and C.

	Wire length A (1 layer)	Wire length B (4 layers)	Wire length C (16 layers)
Pythagoras theorem	$L^2 + L^2 = A^2$	$(\frac{L}{2})^2 + (\frac{L}{2})^2 = B^2$	$(\frac{L}{4})^2 + (\frac{L}{4})^2 = C^2$
	$\sqrt{2 * L^2} = A$	$\sqrt{2 * (\frac{L}{2})^2} = B$	$\sqrt{2 * (\frac{L}{4})^2} = C$
	$L * \sqrt{2} = A$	$(\frac{L}{2}) * \sqrt{2} = B$	$(\frac{L}{4}) * \sqrt{2} = C$
		$\frac{L*\sqrt{2}}{2} = B$	$\frac{L*\sqrt{2}}{4} = C$
Remaining_wire_length		$\frac{Wire\_length\_A}{(\sqrt{4} \Rightarrow 2)} = B$	$\frac{Wire\_length\_A}{(\sqrt{16} \Rightarrow 4)} = C$

Table 3.2: An example of the rule of thumb which shows that the rule of thumb hold (equation 3.1) for wires A, B and C from Figure 3.3

'diagonal wire' length. The diagonal 'X and Y' wire length is equal to the Manhattan wire length, which is a wire that runs also diagonal but only via the horizontal or vertical direction (a stairs pattern). Only for the 'X or Y' line the rule does not hold. In this case the remaining wire length should be multiplied by a factor of two to be similar to the 'X or Y' wire length. This is because for the 'X or Y' case only one X or Y wire is reduced, and thus every cut reduces 50% of the single wire (see Figure 3.4(c)). Conversely, the remaining 'X and Y' and 'diagonal' cases contain two wires (X and Y), and every cut reduces only one of both. Therefore, it only provides 25% wire reduction, and thus explains the difference between the rule of thumb and the 'X or Y' wire reduction case. Conclusively, it is depicted that the reduction after fourth layer is smaller than before the fourth layers. Meaning that *endless stacking does not give endless (large) wire reduction*. Furthermore, *eventually the gate delay will be larger than the wire delay* [10].

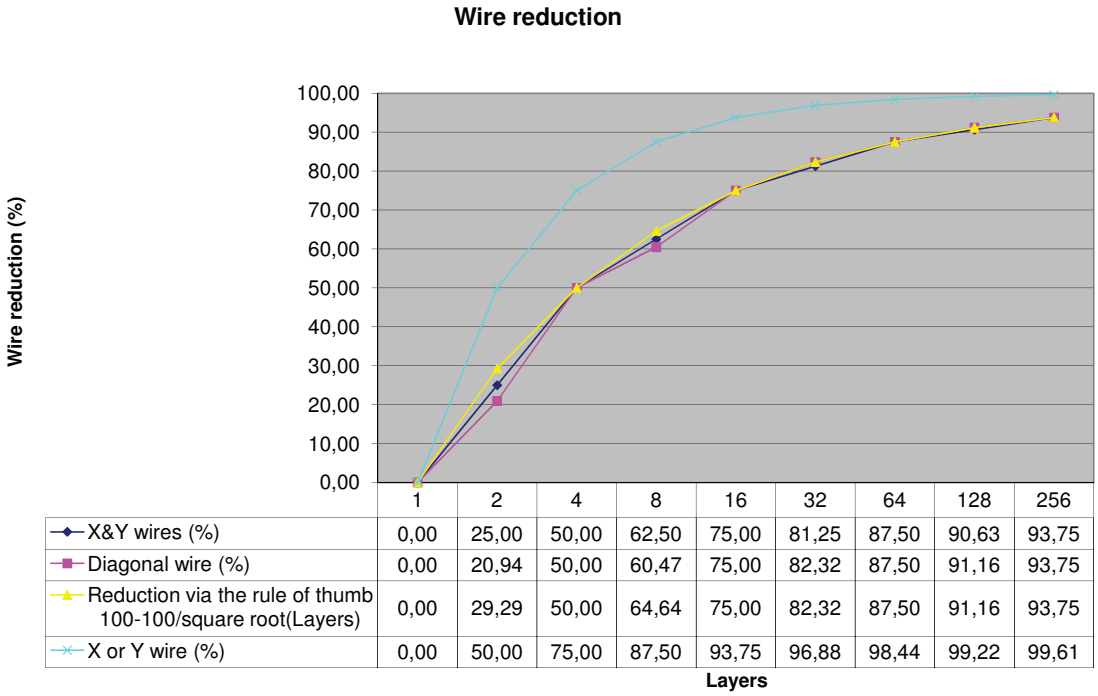


(a) A wire running via the X AND Y direction

(b) A wire running diagonally.

(c) A wire is running via the X OR Y direction

Figure 3.4: Three (worst) cases for which the rule of thumb holds. The outer arrows indicate the size, the inner arrow(s) indicates the worst-case wire length and the dotted lines indicate the cut lines. The width (W) and the height (H) are assumed to be equal.



	1	2	4	8	16	32	64	128	256
X&Y wires (%)	0,00	25,00	50,00	62,50	75,00	81,25	87,50	90,63	93,75
Diagonal wire (%)	0,00	20,94	50,00	60,47	75,00	82,32	87,50	91,16	93,75
Reduction via the rule of thumb 100-100/square root(Layers)	0,00	29,29	50,00	64,64	75,00	82,32	87,50	91,16	93,75
X or Y wire (%)	0,00	50,00	75,00	87,50	93,75	96,88	98,44	99,22	99,61

Figure 3.5: A 2D plane is sliced in to n layers. The wire reductions are in respect to the 2D wire length.

3.2 Memory-on-memory

This section shows that the access latency, the power consumption and the footprint of a 3D memory reduces, compared to a 2D memory with an equal amount of bit storage. Furthermore, it shows that memory-on-memory reduces the Non-Uniform Cache Access (NUCA) latency variation and the memory wall problem.

A 2D memory design has a very regular structure that makes it easy and beneficial to partition across multiple dies. Currently (early 2010), there are institutes and companies that develop and

/ or ship 3D Flash, 3D SRAM and 3D DRAM memory devices [9, 41, 60][1]. 3D memory refers in this thesis to 3D DRAM memory or 3D SRAM cache.

The data in a 3D memory can be stored via the single-layer or the multi-layer data approaches (horizontally or vertically), such as earlier indicated in Section 3.1.1. Both methods are simulated by [16], the horizontal scheme is called Single-Layer Data Access (SLDA) and the vertical scheme is called Multi-Layer Data Access (MLDA). More information can be found in Appendix E.1. [16] shows that the SLDA scheme consumes less energy, compared to the MLDA scheme. It is because the MLDA scheme uses all the layers per data access and the SLDA schemes uses only one layer, which saves energy. It is because the SLDA scheme shuts down the remaining unused layers to save energy. Conversely, the MLDA scheme obtains better memory access times (faster) and a smaller footprint, compared to the SLDA scheme. However, this is because [16] assumed that the SLDA scheme has extra TSVs, which takes area and it has parasitic capacitance (reduces speed). The author of this thesis thinks that without the extra TSVs the footprint and access times are similar, since the differences in the presented results are minor and the only reason presented by [16] for those differences are the extra TSVs. Thus, the *SLDA scheme does not have any benefits above the MLDA scheme*, with respect to the footprint and memory access latency.

No thermal information is known about 3D memory stacking. However, it is known that memory circuits are cooler than logic circuits, and thus logic-on-logic stacking is much more problematic than memory-on-memory. However, in Section 3.3 we will see that logic-on-logic circuits can handle these temperature problems, and thus the author of this thesis *expects that the memory-on-memory circuits can handle the increase in temperatures* as well, since they produce less heat than the logic-on-logic circuits.

Increasing wire delay makes it difficult to provide uniform access latencies to all L2 cache banks in a 2D plane. One known alternative is the Non-Uniform Cache Access (NUCA) architectures [61] (a.k.a. NUCA caches). It allows nearer cache banks to have lower access latencies than the farthest cache banks. However, the NUCA cache proposal / solution does not reduce the wire delay, which is the problem. Conversely, with 3D memory-on-memory these wire lengths are reduced, and the real problem is attacked. It decreases the access latencies to the furthest banks, and thus improving the overall memory access time. Hence, *memory-on-memory diminishes the difference between the closest and furthest bank (NUCA latency variation).* 3D integration enables this by stacking memory banks or via array splitting. Bank stacking reduces the wire delay between the cache edge and the farthest bank (inter-bank delay), and array splitting reduces the intra-bank plus the inter-bank wire delay. Array splitting speeds up the memory access latency of each bank, and thus *memory-on-memory attacks the memory wall problem* [62]. The bank and array approaches are discussed in the next paragraphs, and it shows that the access latency, the power consumption and the footprint of a 3D memory reduces, compared to a 2D memory with an equal amount of bit storage.

3.2.1 Bank stacking

The bank stacking strategy stacks 2D memory banks on top of each other to *reduce the inter-bank delay*, such as shown in Figure 3.6(b). Bank stacking uses the FUB stacking strategy because each bank is seen as a FUB. Bank stacking *reduces the latency variation of a NUCA cache*

[1] no technical details are known.

because the distance between the closest and farthest bank is reduced, seen from the cache edge. The memory banks can be stacked from right-to-left as shown in Figure 3.6(b), or the banks can be stacked from bottom-to-top.

The worst-case path in Figure 3.6(a) contains one Y and three X wire lengths. By applying right-to-left bank stacking the wire length is reduced to only one Y and one X wire length. This is an improvement of 50% (assuming that X=Y). Furthermore, *the footprint is reduced with 50%.* The disadvantage is that the Y wires are not reduced by this stacking approach, which means that (only) the X wire length can be reduced (50% for each layer [9]. However, this reduction leads to *some power and delay improvement.* For example, a 1-Mbyte planar data cache (8-way set associative, 64 banks) provides access latency, and energy per read access improvements of 9.7% and 31.5% [8, p.39], respectively. These improvements are obtained with *minor redesign efforts* because the internal structure of a bank (FUB) remains unchanged.

Conclusively, bank stacking reduces the reduce the inter-bank delay, which reduces the latency variation of a NUCA cache. Furthermore, it reduces the footprint, but provides only some power and speed improvements. However, bank stacking needs only low redesign efforts, which is advantageous.

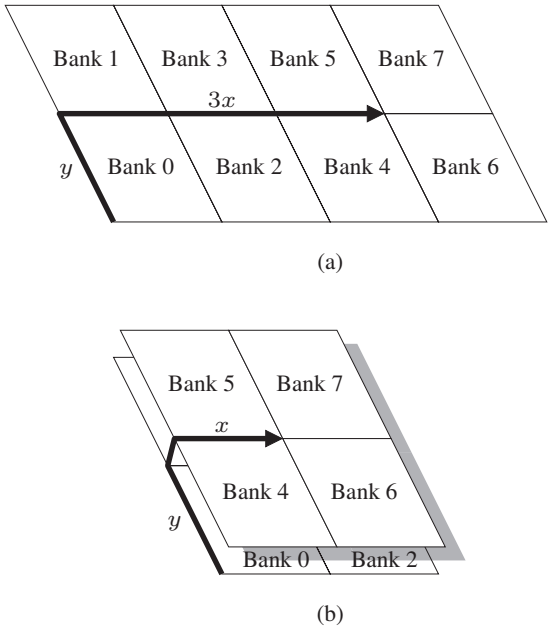


Figure 3.6: SRAM memory bank stacking. The black arrow indicates the worst-case path from the cache edge towards the furthest bank. Assumes is that X=Y. (a) The 2D SRAM bank layout with a worst-case path of four (4=1·Y+3·X). (b) The banks are stacked from right-to-left with a worst-case path of two (2=1·Y+1·X). [9]

3.2.2 Array splitting

This sections shows that array splitting reduces the access latency (compared to a 2D memory), and thereby attacking the NUCA cache and the memory wall problems. Furthermore, the footprint and the power consumption are reduced, compared to a 2D memory with the same bit capacity.

Array splitting uses the logic gate splitting strategy, and it *reduces the inter-bank wire lengths, and the intra-bank wire lengths*. In particular, the bitlines and wordlines within a bank are optimized, and subsequently the wires between the banks are also reduced, see Figure 3.7(c). The NUCA cache problem is attacked, since the *inter-bank wires are reduced*. Furthermore, the memory wall problem is diminished because by the *intra-bank wire reduction speeds up the access latency of a memory bank*. The author of this thesis beliefs that *array splitting is the main advantage 3D integration provides for memory-on-memory devices*. It is because it is able to speed up current 2D memories with medium redesign efforts because memories have a very regular structure, and thus only one memory bank should be redesigned that is used in the whole design. Furthermore, the whole 2D memory plane can be made compact, and thus the long wire lengths of a 2D memory are reduced, which provides significant latency reductions (5%-20% [9]).

A memory bank consists out of one or more memory arrays. On its turn, the memory array consists out of a large 2D memory cell grid (6T SRAM) with some peripheral logic, such as row decoders and sense amplifiers (SA) (see Figure 3.7(a)). When the array is split across the vertical axis it is called column stacking, due to the stacking of the columns / wordlines (see Figure 3.7(b)). Conversely, if the array is split across the horizontal axis then it is called row stacking (see Figure 3.7(d)). Column stacking reduces more latency than row stacking, but row stacking consumes less power, compared to column stacking. Both methods are explained in the next sections.

3.2.2.1 Column stacking

Column stacking reduces inter-bank and the intra-bank wire lengths. With column stacking the wordlines can be folder (see Figure 3.7(b)) or sliced.

With the folded wordlines, one signal driver (inside the 1-to-n decoder) at the middle of the wordline is placed, which folds the wordline into parallel wordlines, see Figure 3.7(b)). A parallel wordline has twice the width of a 2D wordline, leading to a 50% decrease in resistance [9, p.4] and they have the same capacitance, which leads to a speed improvement.

The sliced wordlines are not discussed by the articles [8, 9, 17], but the author of this thesis assumes it is possible to slice the wordline in the middle into two separate wordlines, which needs two signal drivers at the 1-to-n decoder. The resistance of a sliced wordline is less than a 2D wordline, since the length is shorter. Furthermore, the capacitance is of the sliced wordline is less than the original 2D case. Therefore, the sliced wordlines also gives a speed improvement. It is unknown for the author of this thesis how much faster this solution is compared to the first case (folded bitlines). For the folded or sliced wordlines, it is possible to place the sense amplifiers at the bottom layer or divided (split in half) over both layers.

The footprint of the bank reduces because the memory array is divided over two layers, and beside the intra-bank wire length reduction also the inter-bank wire length is reduced. This holds for all the array splitting methods (column and row stacking). The worst-case wire length along

the x direction from Figure 3.6(a) reduces from three 'X' to one and a half 'X' (see Figure 3.7(c)). That is why array splitting reduces the intra and inter-core wire delays.

Column stacking activates two layers at the same time, and reads the same row at both layers. Thus, both layers generate at the same place and time heat[2], which is not advantageous.

3.2.2.2 Row stacking

Row stacking reduces inter-bank and the intra-bank wire lengths. With row stacking, the bitlines and the row decoder are sliced (see Figure 3.7(d)). The resistance and the capacitance of the sliced bitlines are reduces, compared to the 2D memory from Figure 3.7. Thus, it improves the latency and power consumption of a bitline. The row decoder activates only *the top or bottom layer*, in this way thermal stacking of two simultaneous active layers are avoided[3]. There are latency and power reductions at the intra-FUB and inter-FUB level for row stacking, due to the shorter bitlines and the reduction of the banks its footprint.

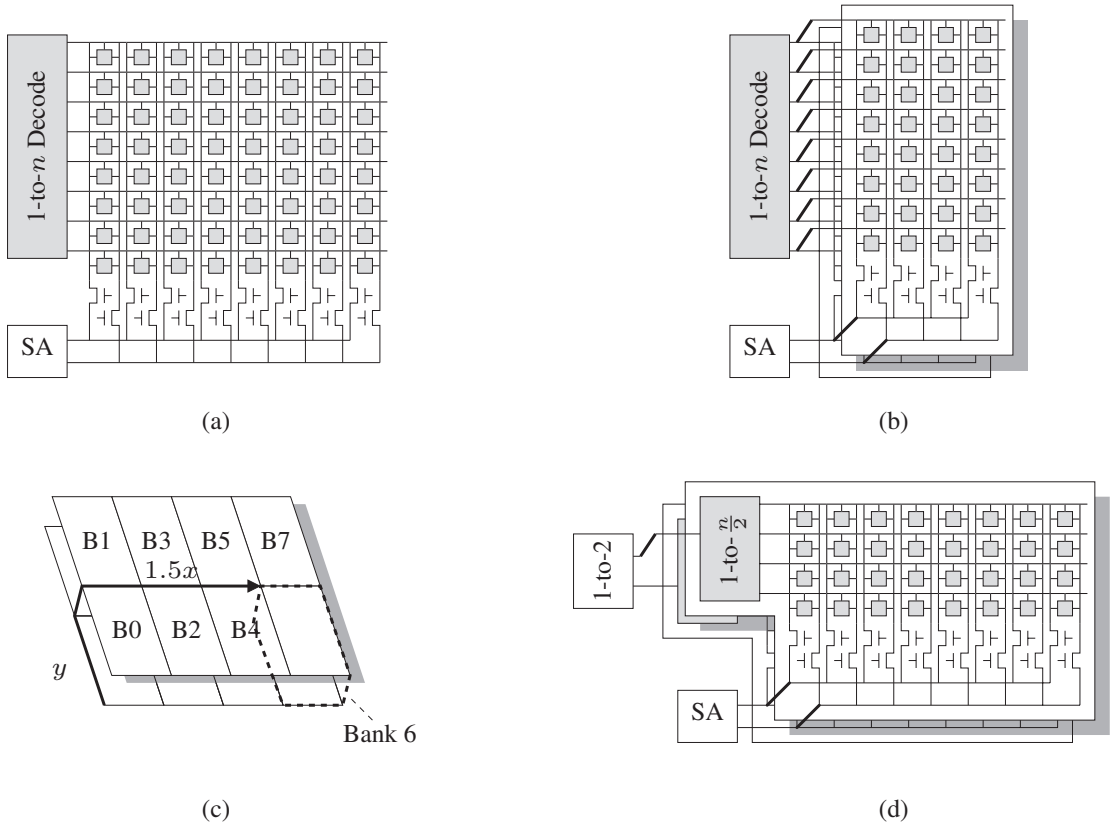
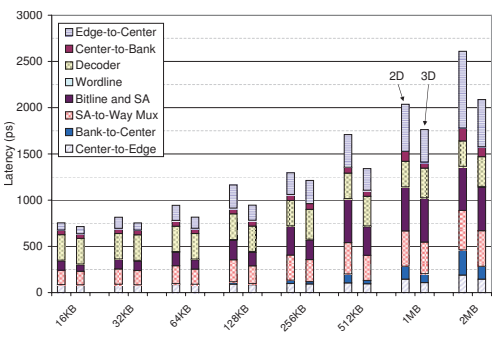
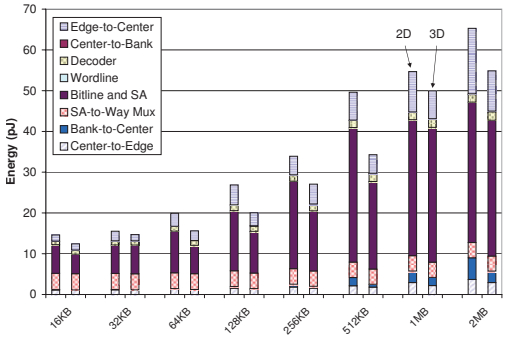


Figure 3.7: (a) A 2D memory array, (b) A 3D column stacked memory array, (c) column stacked banks. The inter-bank wire lengths are also reduced, and (d)3D Row stacked memory array [9].

[2]No numbers are known.
[3]No numbers are known.



(a) Latencies of 2D and 3D caches. [9]



(b) Energy of 2D caches and 3D caches. [9]

Size (KB)	Latency 2D (ns)	Latency 3D (ns)	Savings %	Cycles 2D @ 4GHz	Cycles 3D @ 4GHz
16	0.754	0.714	5.3	4	3
32	0.815	0.754	7.6	4	4
64	0.944	0.816	13.6	4	4
128	1.164	0.945	18.8	5	4
256	1.296	1.213	6.4	6	5
512	1.709	1.341	21.5	7	6
1024	2.037	1.763	13.5	9	8
2048	2.609	2.087	20.0	11	9

(c) Latency results table. [9]

Size (KB)	Energy 2D (pJ)	Energy 3D (pJ)	Savings %
16	14.61	12.43	14.9
32	15.52	14.69	5.4
64	19.92	15.61	21.6
128	26.87	20.05	25.4
256	33.92	27.07	20.2
512	49.62	34.30	30.9
1024	54.68	49.97	8.6
2048	65.28	54.87	16.0

(d) Energy results table

Figure 3.8: Overall simulation results for 2D and 3D (column and row) stacking. [9]

3.2.2.3 Overall results for column and row stacking

The overall circuit and system-level performance for row and column stacking is presented by [9] and [8, p.39]. [9] is written by the same author as [8, p.39], and both articles present column and row stacking. The only difference is that [9] is published in 2005 and [8] in 2007. Unfortunately, the results are not individually specified for the row or column stacking methods in [9], and it is only briefly presented in [8, p.39].

[8] presents latency and power results for column and row stacking. For column stacking, *latency and power improvement are obtained of 13.6% and 32.8%*, respectively [8]. For row stacking, the *latency and power improvements are 21.6% and 30.4%*, respectively [8]. Compared to the bank-stacked approach, the column and row stacking approaches provide greater performance and power benefits, but it also requires more redesign [8]. In the simulations of [8], the word lines accounted for more delay than the bitlines. Thus, reduction of the word line length (column stacking) provides a greater speedup compared to reduction of the bitline length (row stacking) [8]. Thus, column stacking reduces more latency, compared to row stacking. However, row stacking saves more energy compared to column stacking. This is because the power reduction is reduced per access for a large number of bitlines, compared of the power reduction of one or two word line (used at column stacking) [8]. Thus, column stacking is better in latency reduction and row stacking is better in power reduction.

The simulation of [9] used the circuit simulator Spice to obtain the latency and energy of the 2D and 3D circuits. Furthermore, a Mase timing simulator from SimpleScalar 4.0 is used in combination with a SPEC2000 benchmark suite, in order to determine the impact on the performance of a processor. The Mase simulator simulated a 6-bit wide processor with 128-entry Reorder Buffer (ROB), 64-entry scheduler, 32 entry load and store queues, 16KB IL1 and DL1 caches, a 256KB L2 shared cache, and a 1MB L3 shared cache. The processor uses a clock frequency of 4GHz, and it is assumed that the design is implemented in 70nm technology.

The latency and energy benefits are depicted for various cache sizes and it is shown in the Figures 3.8(a) and 3.8(b) [9]. The percentage of latency and energy reductions between the 2D and 3D memory is not linear. This is because different numbers of banks are used for an optimal configuration, and that changes the relative benefits between the 2D and 3D memories (see Figure 3.8(c) and Figure 3.8(d)) [9]. The 3D cache implementation[4] improved the performance of the processor by 3.5% [9], due to the faster 3D cache. However, with a larger 3D cache implementation (the exact size is not reported by [9]) a performance improvement of 10.6% is obtained, due to a lower cache miss ratio. The normal and the larger cache size implementation results show the benefits of the 3D technology [9].

In this section the answer for the memory-on-memory topic of research question four is found.

Research question four: What is the architectural potential and impact of 3D integration for *memory-on-memory*, logic-on-logic, memory-on-logic, and 3D NOC?

The potential for memory-on-memory stacking is that banks can be stacked and internally split, which is called bank stacking and array stacking, respectively. The impact is that bank and array stacking reduces the latency variation of a (NUCA) cache because the distance between the closest and farthest bank (inter-bank wire) is reduced. Furthermore, array stacking provides the main advantage of memory-on-memory, since it also diminishes the memory wall problem by reducing the intra-bank wire lengths, which speeds up the access latency of each memory bank. Bank and array stacking reduce the overall footprint, memory access time, and power consumption. The latency and energy reductions [5]for bank stacking is 9.7% and 31.5%, and for array stacking is this 21.6% and 30.4%, respectively. No thermal information is known about 3D memory stacking. However, the author of this thesis expects that the memory-on-memory circuits can handle the increase in temperatures, since the hotter logic-on-logic circuits can handle these temperatures as well.

[4]It is unknown if row or column stacking is used.

[5]No footprint reduction numbers are known.

3.3 Logic-on-logic

This section shows that the wire length reduction can be used to eliminate pipeline stages and / or to increase the operating frequency of a logic-on-logic stacked device. Furthermore, the temperature impact is discussed.

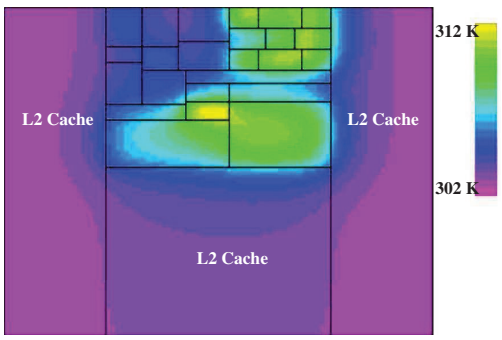
There are two basic strategies to stack an device. The first strategy *reduces the inter-FUB (FUB-to-FUB) wire lengths*, and the second strategy *reduces the intra-FUB wire lengths*. Naturally, a hybrid form of the inter-FUB and the intra-FUB strategy can be used. *Inter*-FUB wire reduction can be achieved via FUB repartitioning or via logic gate splitting, and *intra*-FUB wire reduction can only be achieved via logic gate splitting. *The inter-FUB and intra-FUB wire reductions can be used with a logic-on-logic device for two methods: (1) Eliminate pipeline stages and (2) Increase the operating frequency*. The pipeline stages can be eliminated, since 3D integration compacts a 2D planar design. Therefore, the wires are shorter and thus some pipeline stages become superfluous. For example, [11] simulated a 3D Intel Pentium 4 processor (two layers) and archived pipeline and performance improvements of 25% and 15%, respectively. Conversely, the shorter wire lengths can be utilized to increase the operating frequency, which speeds up the overall performance. Furthermore, for both methods power reductions are achieved because the inter-FUB and intra-FUB wires are shorter, and thus they have less parasitic capacitance, which automatically reduces the power consumption. Conclusively, *the reduction of pipeline stages and the increase of the operating frequency are the main potentials for logic-on-logic stacking.* In Section

The temperature impact for logic-on-logic stacking is important to consider, since multiple power dense and thus hot planes are stacked on top of each other. There are three differences between the 2D and 3D thermal situation. The first difference is that the *heat sink area reduces* equally to the footprint reduction. Thus, more heat should be dissipated by a smaller area. Secondly, the *distance between the heat sink and the lower layer is larger*, compared to a planar chip. Thirdly, the power density increases, and thus is the maximum temperature higher, compared to the same circuit on a planar chip. IBM proposes to solve the cooling challenge via *water cooling structures*. Water is pumped through the chip, and it is conducted by channels of 50μm between the layers [63]. Another solution is to use *dummy TSVs to dissipate heat towards the upper layers*, since metal TSVs can conduct heat well [23, 58]. [58] shows that the regular TSV, used for communication, also conduct the heat well. It is shown for a stacked Alpha 21364 processor of four logic-on-logic layers that the temperature difference between the closest and the farthest layer from the heat sink is not large (<2.5°C). The TSVs are the only connection points between the layers, the remaining space between the layers is not filled (air gap). Thus, this indicates that *TSVs can conduct heat well*. However, the maximum temperature for the 4-layer stack increased with 32.2°C (compared to the 2D processor) to 68.9°C, see Figure 3.9. Figure 3.9 also depicts the a 2-layer stack that reaches a maximum temperature of 54.9°C, which is an increase of 16.4°C, compared to the 2D chip. Observe that in Figure 3.9 the logic structures are cooler than the cache structures. the *temperature increase significantly*, but not four or two times hotter at a 4-layer and 2-layer stack, respectively. The author of this thesis belief that it is because the *wire reduction diminishes the heat generation*.

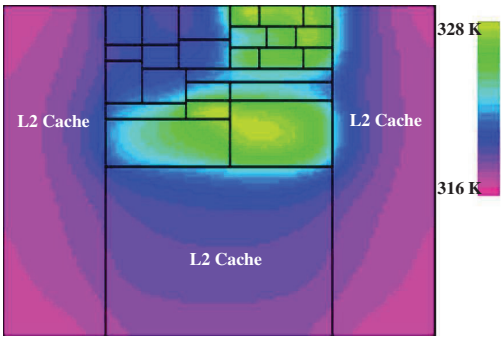
Without extra cooling structures it is also possible to achieve a temperature neutral[6] stacked chip. [11] simulated a stacked Intel Pentium 4 processor (2 layers), which resulted in speedups.

[6]Temperature neutral is with respect to the original temperature of the 2D chip.

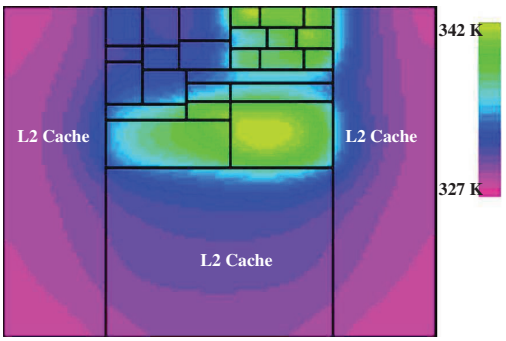
However, some of the performance gain is trade-in to achieve a lower temperature. By *scaling the supply voltage and the clock frequency* the power consumption, performance and temperature of a chip are controlled, see Table 3.3. [11] obtained a *temperature neutral 3D processor*, while achieving power and performance improvements of 34% and 8%, respectively.



(a) Thermal Profile of the planar Processor. [58]



(b) Thermal Profile of the 2-layer 3D Processor. [58]



(c) Thermal Profile of the 4-layer 3D Processor. [58]

Figure 3.9: Thermal profiles of an 2D and 3D Alpha 21364 processor. The figures are not shown in perspective with respect to each other. Purple indicates cool areas and the yellow and green hotter areas.

The next three sections present examples that use the wire length reduction for frequency improvements and pipeline reductions. Furthermore, also thermal information is presented. The first two sections are presented in the appendix, which is interesting but not mandatory to read. In the last section, Section 3.3.3, a Kogge Stone (KS) adder and a log shifter are stacked, and where the wire reductions can be used for frequency increase or for power consumption reduction.

3.3.1 Pipeline reduction and thermal herding

Although not mandatory, the reader is invited to read Appendix F.1.

The appendix shows that the pipeline stages of a 2-layer Intel Pentium 4 can be removed, and that increases the performance with 15% but also the temperature with 14°C (compared to the 2D case). Furthermore, the appendix shows that it is possible to achieve a temperature neutral 3D stacked Intel Pentium 4 processor. The supply voltage and the clock frequency are varied to optimize the design in power, performance or temperature, as can be seen in Table 3.3. Subse-

quently, a temperature neutral 3D processor is achieved, while achieving power and performance improvements of 34% and 8%, respectively. Conclusively, the pipeline reduction improved the performance, while having a lower power consumption or a similar temperature, depending of the optimization preference.

Table 3.3: The 3D results are in respect to the 2D situation and the bold numbers are indicating similar 2D values. [11]

	Power (%)	Temperature (%)	Performance (%)	Vcc (%)	Frequency (%)
2D baseline	100	99	100	1	1
Same power	**100**	127	129	1	1.18
Same frequency	85	113	115	1	**1**
Same temperature	66	**99**	108	0.92	0.92
Same performance	46	77	**100**	0.82	0.82

3.3.2 Frequency speedup

Although not mandatory, the reader is invited to read Appendix F.2, some information is already briefly presented in Section 3.3. In the appendix an Alpha 21364 processor is stacked in two and four layers. The wire reductions are used to increase the frequency of the processor and larger caches are used. The appendix is an summary of an article [58] and a journal [17]. They discuss the same proposal and are published in the same year. Moreover, they have one author in common, and thus it is assumed that they describe the same proposal. The summary is made because some information is only presented in the article and some only in the journal.

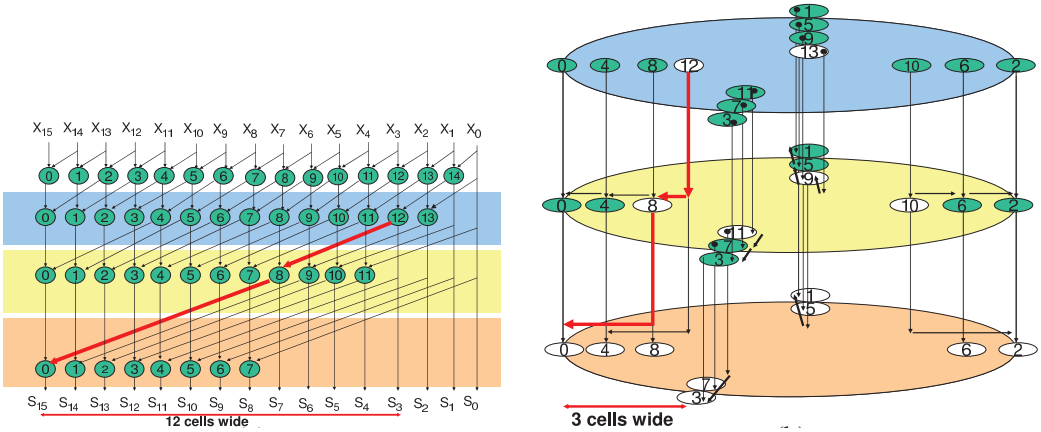
The maximal obtained speedups are 9.73% for the 2-layer and 16.61% for the 4-layer design. The overall temperature difference between the closest layer and the farthest layer from the heat sink is not large (<2.5°C), and it is because TSVs conduct heat well. The temperatures on the 2-layer and 4-layer increased with 16.4°C and 32.2°C (compared to the 2D processor), respectively.

3.3.3 Impact on arithmetic units

In this section a Kogge Stone adder and a log shifters are stacked according to the logic gate splitting strategy. The wire reductions can be used to increase the frequency or to reduce the power consumption, depending of the optimization preference. Furthermore, performance of the log shifters improved only for two layers, beyond two layers marginal benefits was reported. Thus, it shows that endless stacking does not provide endless wire reductions, since the circuits become eventually gate dominated instead of wire dominated [51]. The average peak-temperature (average of multiple runs) of the 16-bits KS adder and log shifter increased with 5°C and 20°C, respectively. The next two sections present the basics of the Kogge Stone adder and a log shifters, they are followed by a section where the results for both designs are discussed.

3.3.3.1 Kogge Stone adder

The Kogge Stone (KS) adder is one of the fastest adders in CMOS design [10]. However, wire delay dominates its performance [10]. The critical path for a 2D 16-bit KS adder is indicated in Figure 3.10(a), and it is expressed in the number of cells that the wire crosses. Thus, in Figure 3.10(a) the critical path spans 12 cells, from the top-right corner to the bottom-left corner. The real circuit layout is assumed to be partitioned as the 2D (and 3D) *logical* drawing of Figure 3.10(a). However, in reality this is not the case. Usually, to reduce the critical path, the adder cells from the bottom layer are shifted towards the middle in the *circuit* layout. Nevertheless, [10] claims that the worst-case is 12 cells and thus we assume this worst-case path in this section. The 2D design is divided over four different layers, each having their own color. For clarity, only the bottom three layers are shown in the 3D layout (see Figure 3.10(b)). The critical path in the 3D layout is reduced to three cells wide and two TSVs. The simulation results are presented in Section 3.3.3.3.



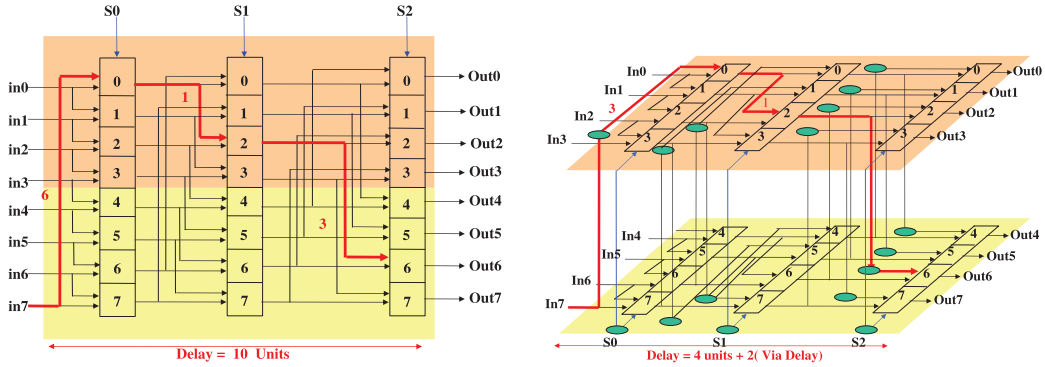
(a) Logic layout of a 2D 16-bit Kogge stone adder, where the critical path spans 12 cells.

(b) The three bottom layers of a 3D Kogge stone adder. The critical path spans three cells and two TSVs.

Figure 3.10: The critical path is indicated for the Kogge Stone adder. The critical path is expressed in the number of cells the wire crosses before it reaches the destination. [10]

3.3.3.2 Logarithmic shifter

The performance of a 2D logarithmic shifter is wire dominated [10]. The 2D layout of an 8-bit log shifter is depicted in Figure 3.11(a). The log shifter is constructed from multiple multiplexer rows and they are controlled by the signals S0, S1 and S2. The 2D log shifter is divided over two layers and each layer has their own color. The wire length is expressed in the number of cells that the wire crosses. The 2D critical path spans 10 cells, while the corresponding 3D path spans only four cells and two TSVs (see Figure 3.11(b)). The simulation results are presented in Section 3.3.3.3.



(a) An 8-bit log shifter and it is constructed from three multiplexer rows. The critical path spans 10 cells.

(b) 3D Log shifter layout and the critical path span four cells and two TSVs.

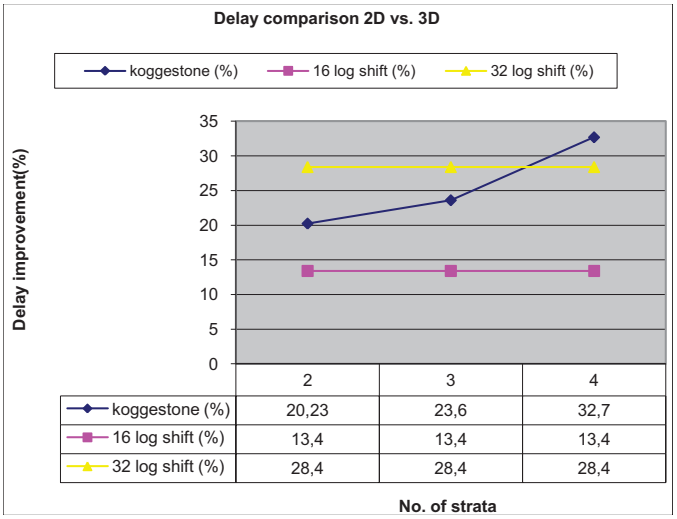
Figure 3.11: The critical path is indicated for the log shifter. The critical path is expressed in the number of cells the wire crosses before it reaches the destination. [10]

3.3.3.3 Simulation results Kogge Stone adder and logic shifter

The circuit layout is designed with the 3D magic tool of the Massachusetts Institute of Technology (MIT). The latency and power is simulated with the Hspice simulator. Furthermore, the thermal impact on the designs is simulated by 3D Hotspot with an ambient temperature of 45°C.

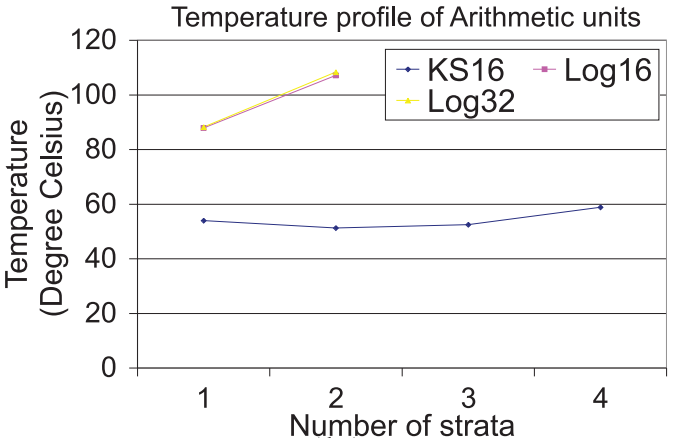
The delay and power improvements for a 16-bit Kogge Stone (KS) adder, and 16 and 32-bit log shifters are depicted in Figure 3.12(a) and Figure 3.12(b), respectively. The results in the graphs are normalized to the 2D situation. The delay and power of the KS adder has improved for all the simulated layers (2, 3 and 4 layers). However, the delay and power for the 16 and the 32-bit log shifters improved only for two layers, and beyond these layers only marginal benefits were reported. This shows that it is *not always beneficial to split a circuit over multiple planes*. It is because *eventually the circuits become gate dominated* instead of wire dominated [51]. Furthermore, it is remarkable that other articles [58, p.3] and [51] indicate that logic structures (such as the Brent-Kung, Sklansky and the Kogge-Stone adders) are predominantly gate delay dominated, and thus no significant speedups and power savings were achieved. Thus, before logic gate splitting is applied *it should be determined if the FUB is wire or gate delay dominated*.

The average peak-temperature (average of multiple runs) of the 16-bits KS adder increased with 5°C as depicted in Figure 3.12(c). It is interesting to see that the temperature of the KS adder decreased at the 'two-layer' point, compared to a single layer. Unfortunately, this is not discussed in [10]. However, the author of this thesis thinks that it is because the wires are shorter, and thus they produces less heat. The average peak-temperature increased with 20°C for both (16-bit and 32-bit) log shifter designs. This significant increase of temperature is due to its compact 3D design and minimal power reduction, compared to the KS adder [10].



(a) Delay improvement of the 3D designs.

(b) Power improvement of the 3D designs.



(c) Thermal results of the Kogge stone and Log shifter. [10]

Figure 3.12: All the results are normalized to the 2D situation.

In this section the answer for the logic-on-logic topic of research question four is found.

Research question four: What is the architectural potential and impact of 3D integration for memory-on-memory, *logic-on-logic*, memory-on-logic, and 3D NOC?

The potential for logic-on-logic stacking is that the wire length reduction can be used to eliminate pipeline stages and / or increase the operating frequency of a design, which has an positive impact on the performance and power reduction. For a 2-layers 3D Intel Pentium 4 processor this resulted in a pipeline stage reduction and performance improvements of 25% and 15%, respectively. However, the log shifters showed that it is not always beneficial to split a circuit over endless layers because eventually the circuits become gate dominated instead of wire dominated. Only two layers are beneficial for the log shifters, providing delay and power improvements up to 28% and 8%, respectively. No numbers are known about the footprint reduction, but the author of the thesis expects that the footprint also reduces for the stacked devices.

The temperature impact for logic-on-logic stacking is important to consider. There are three differences between the 2D and 3D thermal situation: (1) the heat sink area reduces, (2) the distance between the heat sink and the lower layer is larger, and (3) the power density increases. Possible solutions to solve the cooling challenge are: the use of water cooling, dummy TSVs, and scaling the supply voltage and the clock frequency. TSVs conduct heat well, since for a stacked Alpha 21364 processor of four logic-on-logic layers the temperature difference between the closest and the farthest layer from the heat sink is only <2.5°C. By scaling the supply voltage and the clock frequency it is even possible to achieve a temperature neutral 3D processor, while achieving power and performance improvements of 34% and 8%, respectively. However, it is shown for a stacked Alpha 21364 processor of four logic-on-logic layers, connected with TSVs, the maximum temperature increased with 32.2°C to 68.9°C [58], compared to the 2D processor. The author of this thesis thinks that the temperature increase is significant, but it is not four times hotter for a 4-layer stack because the wire reduction diminishes the heat generation.

3.4 Memory-on-logic

There are four main approaches for Memory-on-logic stacking: (1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies. These approaches are discussed in this section.

An intuitive core stacking approach for memory-on-logic integration is core stacking, where

the cache is stacked on top of the processor plane, such as illustrated in Figure 3.13(b). The benefit of such an organization is the 50% *footprint reduction*, which is achieved with *low redesign efforts*. Furthermore, this configuration allows the use of *multiple memory ports at various positions in the design*, which provides parallel access, and that increases the total memory bandwidth [64]. Moreover, each memory port can have a wider data bus width, compared to the 2D situation [7].

The worst-case path of a 2D cache is depicted in Figure 3.13(a) and it is similar to the 3D core stacked worst-case path, see Figure 3.13(b). Only the wire length between the logic and memory planes is reduced, but not the critical path of the core stacked cache. The worst case path of a cache reduces by the use of a 3D cache, such as banked stacked or array split caches, on top of a processor plane. Furthermore, a *larger cache provides a higher cache hit ratio*, which improves the overall performance of the processor [11]. Moreover, 3D caches have a lower power consumption, compared to 2D caches with the same amount of bits. Naturally, the worst case path of the processor can also be reduced by the use of a 3D processor. This increases the performance of the processor, such as indicated previously at the logic-on-logic section in this thesis.

Another strategy is the use the *heterogeneous* property of 3D integration, which allows different process technologies (nm) or substrates (SOI or bulk) to be used with the same chip. The cache misses are reduced by replacing the original 2D *SRAM cache* for a 2D or 3D *DRAM memory* (and stacking it on top of a processor plane), since 2D *DRAM contains eight times more bits* than a 2D SRAM plane with the same footprint [11]. Furthermore, DRAM cache reduces also the power consumption, compared to a SRAM cache with the same footprint [11]. Even though the main (DRAM) memory can be placed on-chip, the author of this thesis beliefs that an off-chip (DRAM) memory will remain between the processor and the hard disk for speed optimizations. It is because a 3D memory is limited by the number of layers, due to practical manufacturing reasons. For example, 50 layers can give a low yield and is not practical.

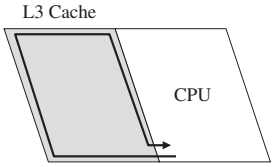
Instead of stacking the memory layer on top of the processor plane another approach can be used, the multi-layer data approach. The basics of this approach is previously presented in Section 3.1.1. The memory plane and processor plane are via logic gate splitting divided over multiple planes, as shown in Figure 3.13(c)[8]. However, this approach requires a lot of redesign of the 2D circuits. Furthermore, this strategy does not allow the use of on-chip DRAM memory, since manufacturing DRAM and logic on a single 2D plane is difficult. This is a drawback because DRAMs are more power efficient, and have a lower temperature, and larger bit capacity, compared to a 2D SRAM cache of the same footprint size.

[11] presents power density and heat maps, as shown in Figure 3.15, and they reaffirm that the generated heat within 2D logic / cores is much higher (29°C), compared to the heat of a 2D cache. Furthermore, a SRAM cache plane is hotter than a DRAM plane [11], since SRAM circuits are power denser than DRAM circuits. Thus, it is better to stack DRAM on top of a processor than a SRAM cache. However, even when a SRAM cache is stacked on a processor the temperature increase is less than 5°C, and it reached a maximal temperature of 92.9°C. [11]. The increase is minor because a heat sink with convectioned air cools the chip. Nevertheless, the *thermal impact of the stacked memory on to logic is not significant*, while there are significant performance and power improvements [11].

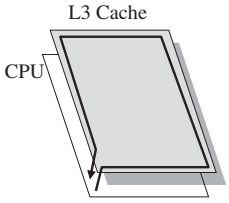
[7] No articles are known that presents these memory-on-processor (specific) improvements.

[8] No articles are known that presents these memory-on-processor (specific) improvements.

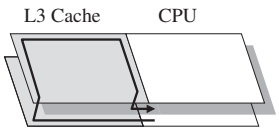
In the next section we will look at an example that makes use of a larger cache on top of a processor and the heterogeneous property of 3D integration. Larger DRAM and SRAM caches are stacked on top of a processor to reduce the cache misses.



(a) 2D logic core and memory cores on one plane.



(b) The L3 cache from Figure 3.13(a) is stacked on top of the processor plane (core stacking).



(c) The L3 cache and processor plane form Figure 3.13(a) are divided over multiple layers (via FUB repartitioning or logic gate splitting).

Figure 3.13: Two different memory-on-logic approaches are depicted and the worst-case path is indicated by the bold arrow. [9]

3.4.1 Larger 2D cache stacked on a processor

In this section larger DRAM and SRAM caches are stacked on top of a Intel core 2 duo processor to reduce the cache misses, and it improves performance.

It is beneficial to stack a *larger* 2D cache on top of the 2D processor, compared to the original cache. It is because the increased bit capacity of the largest (2D) on-chip cache improves the performance and the power consumption. The performance is improved by capturing larger working sets / blocks and by the higher cache hit ratio, due to the larger cache. The higher cache hit ratio reduces the off-chip bandwidth demands towards the main memory, see Figure 3.14(a). This results in a reduced power consumption of the off-chip communication [11]. Furthermore, 2D DRAM uses less power than the original 2D SRAM cache [11]. These advantages are shown in the next example, where various caches are stack on a Intel core 2 duo processor.

The cores of the baseline Intel core 2 duo processor have separated (L1) instruction and data caches, and share a 4MB L2 cache (see Figure 3.14(a)). The 4MB L2 cache is connected via an off-chip bus to the main memory and occupies ≈50% of the 2D floor plan. On top of the planar Intel core 2 duo processor, three different (SRAM or DRAM) caches are stacked. The cache access time increases with the cache size for all these simulations and the used microarchitecture parameters are shown in table 3.4. At the first design an extra 8MB SRAM cache is stacked to increase the total L2 cache to 12MB. The area of the extra cache is just as large as the baseline

processor (see Figure 3.14(c)), since 4MB SRAM cache is 50% of the total 2D processor area, and thus twice that area results in a cache size of 8MB. At the second design a 32MB DRAM cache is stacked on top of the cores (see Figure 3.14(d)). The original 4MB SRAM cache is removed and this results in a 50% footprint reduction. As stated previously, a DRAM cache is about 8 times denser than a SRAM cache with the same footprint [11]. Thus, the 32MB DRAM uses the same area as the 4MB SRAM cache, but it is 8 times larger (bit capacitance). The 2MB tags (not depicted in Figure 3.14(d)) of the 8MB DRAM are placed on the bottom layer. The used tag cache technology depends on the individual design (for example, SRAM or DRAM) [11]. At the final design a 64MB DRAM cache is stacked on top of the baseline processor and the tags are stored in the original 4MB SRAM cache. Therefore, the total cache size is 64MB and not 68MB.

Table 3.4: The used parameters for the 2D and 3D simulation. [11]

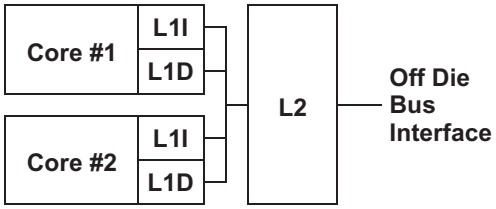
Parameter	Value
Core Parameters	Same as Intel Core 2 Duo
L1D Cache	32KB, 64B line, 8-way, 4 cyc
Shared L2	4 MB, 64B line, 16-way, 16 cyc
Stacked L2	SRAM: 12 MB, 24 cyc
	DRAM (DDR3): 4-64MB, 512B page,
	16 address interleaved banks, 64B sectors
Main Memory	DRAM (DDR3), 16 banks, 4KB page, 192 cyc
Bank delays	Page open 50 cyc
(stacked L2 & DDR memory)	Precharge 54 cyc
	Read 50 cyc
Off die Bus BW	16 GB/s

3.4.1.1 Cache hits and memory access simulation results

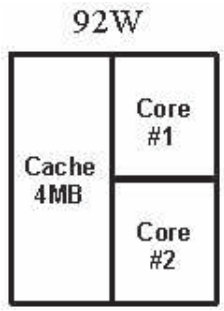
The cache hits and memory access times of the three designs from Section 3.4.1 are simulated with two different simulators. A multi-threaded benchmark is executed on a multi-processor simulator[9]. The multi-processor simulator generates an address trace which is the input to a (trace driven) multi-processor memory hierarchy simulator and it runs billions of memory references to simulate large caches. This memory simulator is developed within Intel and it models all aspects of the memory hierarchy, including DRAM caches with banks, RAS, CAS, page sizes, etc. [11].

The simulation results are depicted in Figure 3.14(f) and it shows the Cycles Per Memory Access (CPMA) and the off-chip bandwidth. The CPMA indicates the average number of cycles per memory (cache) request. At each group on the x-axis the number '4' indicates the baseline processor (see Figure 3.14(f)) and the rest of the numbers indicate the other three presented architectures. The CPMA bars in Figure 3.14(f) show that for several benchmarks (gauss, pcg, sMVM, sTrans, sUS, and svm) the CPMA improves, and it is because of the larger cache size and the subsequent higher cache hit ratio. Thus, the 3D designs have a faster (average) memory

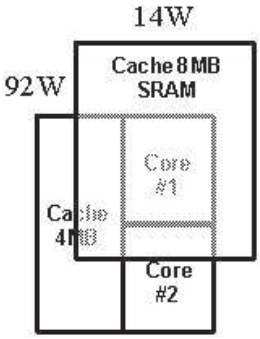
[9]Unfortunately, no simulator name is provided by [11].



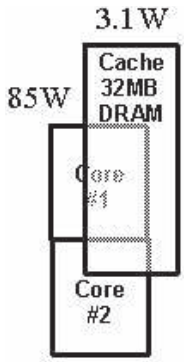
(a) The logical layout of an Intel core 2 Duo baseline processor. The off-die bus interface is connected to the main memory.



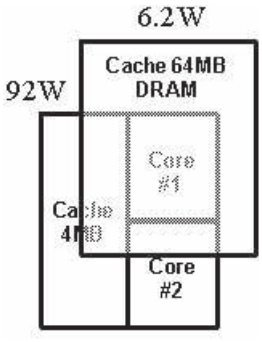
(b) The basic floor plan of an Intel core 2 Duo baseline processor with the original 4MB SRAM L2 cache.



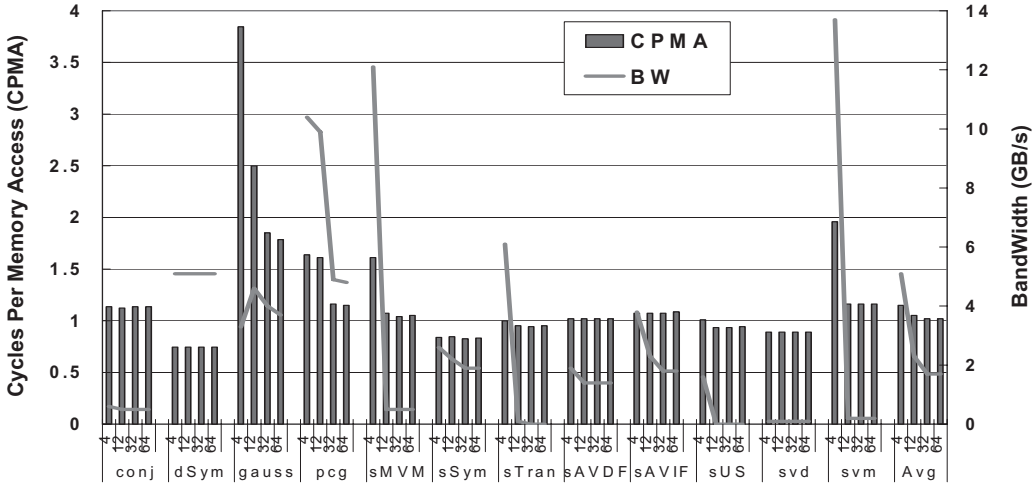
(c) An Intel core 2 Duo processor with an additional 8MB of SRAM cache on top. The total cache size is 12MB.



(d) An Intel core 2 Duo processor with an additional 32MB of DRAM cache on top.



(e) An Intel core 2 Duo processor with an additional 64MB of DRAM cache on top. The total cache size is 64MB because the original 4MB SRAM is used for the tags.



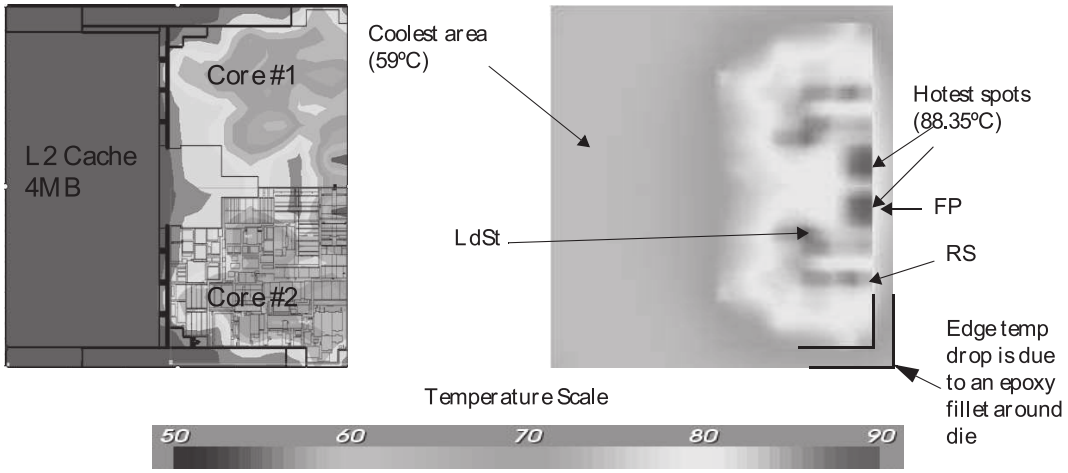
(f) The bandwidth and cycles per memory access (CPMA) results of an Intel core 2 duo.

Figure 3.14: Three different memories stacked on top of an Intel core 2 duo with the bandwidth and CPMA simulation results. The power consumption is indicated per floor plan. [11]

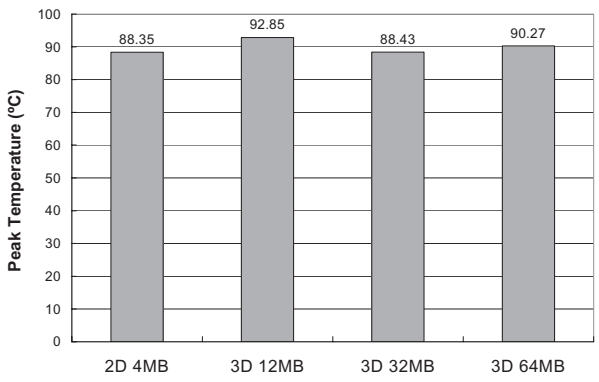
access time than the 2D design. The data of the remaining benchmarks that do not show CPMA reductions is smaller than 4MB. Thus, the data fits inside of the original 4MB cache, and hence more capacity does not always improve the CPMA [11].

The off-chip bandwidth of Figure 3.14(f) indicates the amount of cache hits or misses A high bandwidth demand indicates many misses and a low bandwidth demand many cache hits. The bandwidth trend lines in Figure 3.14(f) show significant reductions as the cache capacities increases [11]. Thus, the *larger cache improved the cache hit ratio* of the Intel core 2 duo processor. It decreased the off-chip communication needs and it speeded up the CPMA. In its turn, the CPMA speeds up the overall performance.

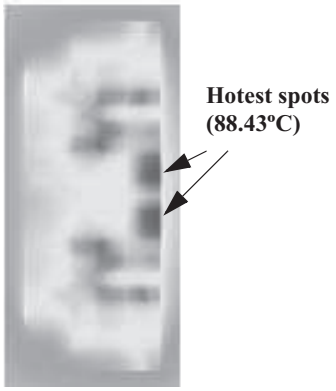
In a particular case, the stacked 32MB DRAM architecture reduces (on average) the bandwidth demand by 3x and the CPMA by 13%. There is also a 66% (average) *power reduction due to the reduced off-chip communication*. A bus power consumption rate is assumed of 20mW/Gb/s [11]. Unfortunately, there is no information presented by [11] regarding the overall performance on the Intel core 2 duo.



(a) The power map (left side) and the thermal map (right side) of an Intel core 2 duo processor.



(b) Temperature results of the 2D and 3D Intel core 2 duo. The L2 4MB SRAM cache occupies ≈50% of the total surface.



(c) A thermal map of an Intel core 2 Duo with 32MB of DRAM on top.

Figure 3.15: Thermal results and maps. [11]

3.4.1.2 Thermal simulation

An Intel thermal modeling tool produced a power and thermal map for the Intel core 2 duo, and it is depicted in Figure 3.15(a). The thermal modeling tool considers the heat sink, integrated heat spreader, die, bonding layer, metal wires, package, socket, motherboard, and an ambient temperature of 40°C [11]. It is assumed that all designs have a standard cooler (a heat sink on top with forced convectioned air), such as used by a standard personal computer. The power and heat maps illustrate the relation between the consumed power and the generated heat. Furthermore, it shows that the heat generated within logic / cores is much higher compared to the heat of the cache [11]. The greatest concentration of power (and thus heat) is located at the FP units, reservation stations, and the load/store unit (see thermal map at Figure 3.15(a)). The planar processor has two hottest spots which are both at 88.4°C, and the coldest spot is 59°C.

The power consumption for the 3D case is indicated per floor plan in Figure 3.14. The caches consume 7W, 14W, 3.1W, and 6.2W, which stands for the 4MB SRAM, 12MB SRAM, 32MB DRAM, and 64MB DRAM cache, respectively. The stacked (DDR3) DRAM consumes less power than an off-chip (DDR3) DRAM because TSVs consume a lot less power than the traditional off-chip interconnects [11]. Furthermore, notice the high power consumptions of the SRAM cache, compared to the DRAM caches at Figures 3.14(c) till 3.14(e). It shows that *SRAMs are more power dense, and thus hotter than DRAMs* [11]. However, the increase between the 2D and 3D (SRAM and DRAM) architectures is <5°C, compared to the original 2D chip. It is because a heat sink with convectioned air cools the chip, and thus is the *thermal impact of the stacked memory onto logic is not significant* improvements [11].

In particular, the thermal map of the 32MB architecture is depicted in Figure 3.15(c). *Cache has uniform power and heat distribution, and thus an (low) uniform temperature is added over the whole plane* [11]. Therefore, the thermal signature of the 3D thermal map is similar to the 2D thermal map. *Conclusively, the thermal impact of the stacked memory onto logic is not significant*, while there are significant performance and power improvements [11].

In this section the answer for the memory-on-logic topic of research question four is found.

Research question four: What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, *memory-on-logic*, and 3D NOC?

There are four potentials for memory-on-logic stacking: (1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies. The footprint reduction depends on the size of the memory and logic, for core stacking it provides 50% footprint reduction, given that the cache and logic are both 50% of the total die size. Power consumption reduces when more memory ports and larger 2D or 3D caches are used, since they reduce the communication distance and cache misses, respectively. Furthermore, the use of heterogeneous technology allows energy efficient planes to be stacked, such as DRAM memory. Performance improvements can be achieved by stacking 3D memory and 3D logic on top of each other. Furthermore, by the use of larger on-chip caches the cache hit ratio improves, and thus also the overall performance. Moreover, the overall performance also improves by using multiple memory ports at various locations and that communicate simultaneously.

SRAM has a higher power density than DRAM, and is thus hotter. However, even when a SRAM cache is stacked on a processor the temperature increase is less than 5°C (compared to the original 2D situation), and it reaches a maximal temperature of 92.9°C [11]. However, the increase is less than 5°C because a heat sink with convectioned air cools the chip. Nevertheless, the thermal impact of the stacked memory on to logic is not significant, while there are significant performance improvements.

3.5 3D Network-On-Chip

With a 3D Network-On-Chip (NOC) there are three aspects important: (1) the total number of 3D routers in a network, (2) the inter-layer interconnect topology (i.e. bus, or point-to-point), and (3) a full crossbar scales inefficiently. Furthermore, 3D integration enables the uses of heterogeneous technology, which allows the separation of photonic and electrical planes.

Although not mandatory, the next section, photonic Network-On-Chip section, invites the reader to read Appendix G.2. The subsequent sections discuss the three important aspects of the NOC (number of 3D routers in a NOC, inter-layer interconnect topology, and the 3D crossbar).

3.5.1 Photonic Network-On-Chip

Although not mandatory, the author of this thesis would like to invite the reader to read Appendix G.2. The appendix explains the basics of a photonic NOC, and shows that the *heterogeneous property of 3D integration allows the separation of photonic and electrical planes*, as shown in Figure 3.16. This separation is desirable, since sharp bends in optical wave guides should be avoided.

Moreover, the appendix discussed that photonic interconnects offers (theoretically) ultra-high communications bandwidths in the terabits per second range [18]. Photonic signaling has low power consumptions, since the power consumption of an optical signal at the chip level is independent of the distance. There are five main components in a photonic network:(1) a laser source, (2) an opto-electric modulator (transmitter), (3) a photonic waveguide, (4) a Photonic Switching Element (PSE), and (5) a photonic detector (receiver). The necessity of a separated 3D photonic plane is because it is preferred to limit (sharp) angles in the waveguide, and the use of PSE [18].

A '3D hybrid photonic and electric NOC' is compared to a '2D all-electrical mesh NOC'. This is done for 64 cores (8x8) and 100 cores (10x10). The waveguide ring architecture provides up to *13x power reduction, a throughput improvement of 1.9x, and a latency reduction of 1.55x*, compared to an all-electrical mesh NOC. The results indicate that this architecture provides a high performance per watt, as well as a moderate throughput and latency improvement. However, [18] did not take into account the off-chip laser source, and this result might thus give a false picture. The results with the laser source are unknown for the author of this thesis.

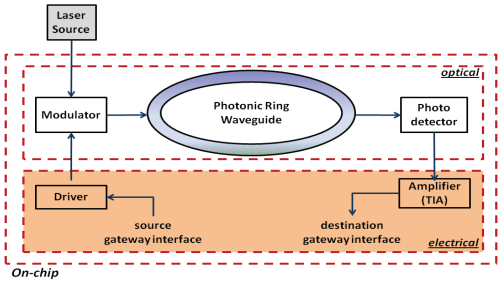


Figure 3.16: High level overview of 3D photonic transmission architecture. Note, the laser source is off-chip.

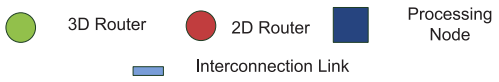
3.5.2 Number of 3D routers in a NOC

This section shows that it is not always necessary to have a full 3D connected NOC, which saves area and power. With a full 3D connected NOC *all* the routers in the network can send packages to different layers.

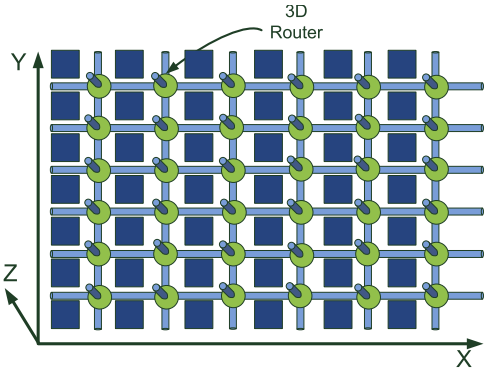
3D chip integration enables the creation of a true physical 3D topology, where the *router* and the *network* are themselves three-dimensional entities. The advantage between the vertical and the horizontal interconnects is the *wire length difference*. In the vertical direction, a TSV is just a few tens of μm long as compared to a few thousand μm for the horizontal direction [13]. In a 3D stacked device, the *NOC architecture* should to enable communication between FUBs and

cores at different layers (inter-layer communication). This can be done with two strategies: (1) a full 3D connected NOC or (2) *non*-full 3D connected NOC.

It is possible to implement a full 3D connected NOC, where all the routers have connections to other layers, as illustrated in Figure 3.17(b). However, [12] shows that *a full 3d connected network is not (always) necessary*. The main advantage of a non-full 3D NOC, is a reduction in energy and area. The key factor is to use a *healthy mix between 2D and 3D routers*. However, the drawback is that the latency increases. Multiple router mixes are simulated by [12], and more information can be found in Appendix G.1. The simulation results of [12] show that not a particular mix of 2D and 3D routers is the best for reducing area, energy, or latency for all the simulated traffic types (uniform, hotspot, and transpose traffic). In general, the *non-full 3D schemes obtained energy and area reductions of 5% and 8%*, respectively. However, they induced a minor latency increase of 8% (on average). However, the non-full 3D schemes can only be used at low and medium traffic loads. Otherwise, it results in an increase of the energy consumption and latency. Thus, the best 2D and 3D router mix depends on the traffic type and the desired topology used in the particular system. Otherwise it could be disadvantageous.



(a) Legend



(b) A full 3D interconnected NOC, where all routers are able to route a package to all the layers.

Figure 3.17: Legend and a full 3D interconnected network. [12]

3.5.3 The inter-layer interconnect topology

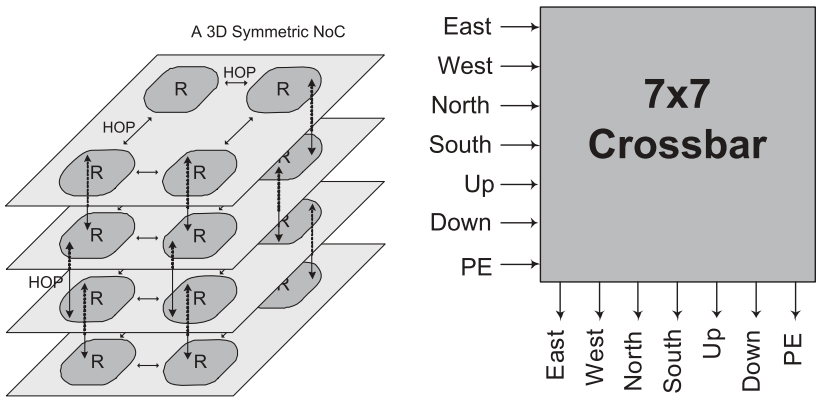
This section show that single-hop communication (via a bus) is the best strategy to use for inter-layer communication because with multi-hop communication the logic of the router dominates the latency.

An important aspect of the *NOC architecture* is the *inter-layer communication method between 3D routers*. This can be achieved via two strategies: *(1) multi-hop communication (point-to-point) or (2) via single-hop communication (bus).*

With the multi-hop communication, a *package travels multiple hops before it reaches the*

upper plane, assuming that it started for the lowest plane, see Figure 3.18(a). On each layer a router is connected to the router above and beneath it, such as depicted in the figure. This multi-hop approach has two main drawbacks: (1) it *does not utilize the wire reduction*, and (2) the *7x7 cross bar needs more area*. With the first drawback, the benefits of the shorter (negligible) inter-wafer distance between the layers are underutilized, since traveling in the vertical dimension takes multiple hops. Furthermore, the *inter-layer communication latency is dominated (per hop) by the router*, compared to the TSV latency. It is because the buffering and arbitration of each package within each router. Naturally, stacking reduces the average number of hops between a source and a destination. However, the inter-layer and intra-layer hops are indistinguishable [13].

Secondly, a baseline 2D 5x5 crossbar is turned into a 7x7 crossbar by adding an up and down port, as shown in Figure 3.18(b). A 7x7 crossbar needs a *larger area*, compared to the planar (5x5) crossbar. It is because *crossbars scale upward very inefficiently*, since a 5x5 crossbar has 25 intersections and a 7x7 crossbar has 49. Conclusively, with the multi-hop strategy there are too many hops between the upper layer and lower layers, and a crossbar scales very inefficiently. Thus, this multi-hop architecture is not optimal to use in a 3D NOC architecture [13].



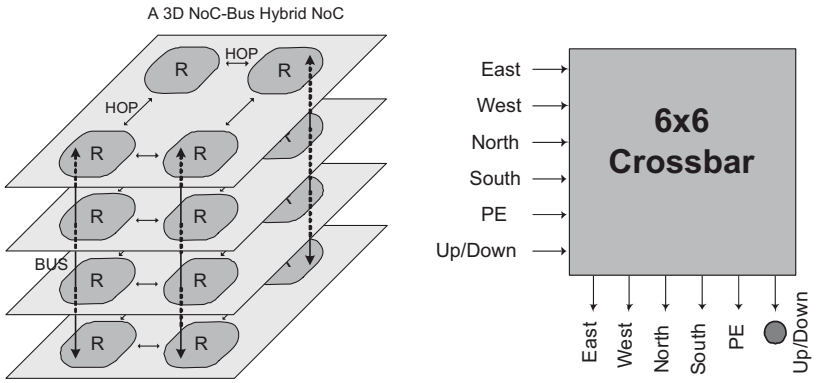
(a) Hop-by-hop communication. (b) 7x7 Crossbar Configuration.

Figure 3.18: A 3D NOC with a hop-by-hop approach. [13]

A hybrid 3D NOC router solves the drawbacks from the multi-hop communication strategy because the router provides a *single-hop (bus) for inter-layer communication*, and multi-hop for intra-layer communication, as can be seen in Figure3.19(a). Inter-layer single-hop communication is beneficial, given the very small inter-strata distance, such as illustrated in Figure 3.19(a). The area of the crossbar reduces because it requires a 6x6 crossbar, compared to the 7x7 crossbar. It is because the bus adds only a single port to the baseline 2D 5x5 crossbar (see Figure 3.19(b)). Thus, flits from different layers that wish to move up/down should arbitrate for access to the shared medium. A drawback is that a single inter-layer bus does not allow concurrent communication.

3.5.4 3D crossbar architecture

This section presents a 3D crossbar that reduces area, power and latency with 48%,36% and 20%, respectively, compared to the previous discussed 6x6 and 7x7 crossbars, see Figures 3.19(b) and



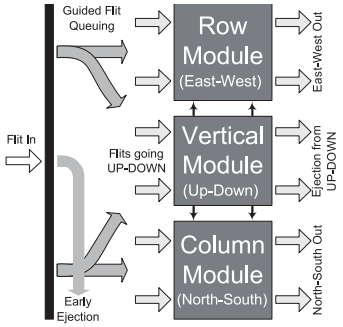
(a) Bus hybrid architecture. (b) 6x6 crossbar.

Figure 3.19: A 3D NOC with an single-hop approach. [13]

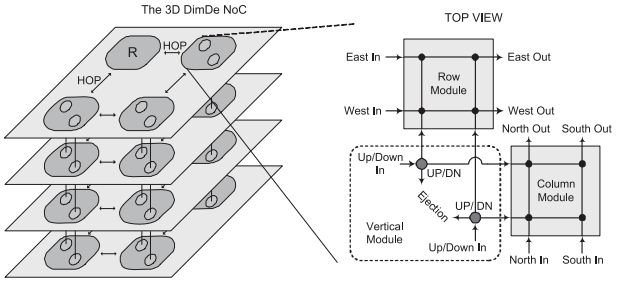
3.18(b)). The improved crossbar scheme reduces the area and power with at least 18% and 17%, respectively, compared to the 6x6 and the 7x7 crossbars. An efficient structure is needed, since *crossbars scale inefficiently* [13]. For example, the smallest crossbar (a 6x6 crossbar) contains 36 interconnection points per layer. Conversely, the 7x7 crossbar need 49 interconnection points per layer, which is much more, compared to the 36 of the 6x6 crossbar.

The improved decomposed crossbar is called 3D Dimensionally-Decomposed (DimDe), and uses *guided flit queuing, which decomposes the incoming traffic into three independent streams* before it reaches the crossbar, as shown in Figure 3.20(a). The east-west traffic stream is the first stream, and it routes packages only along the X direction of a router. The second stream is the north-south traffic stream, which routes the packages only along the Y direction of a router. The third independent stream requires a third vertical module, such as shown in Figure3.20(a). The vertical module contains two inter-layer bidirectional busses, and it is connected to the row and column modules. This segregation of traffic allows the use of smaller crossbars / modules, and they are called the row, column, and vertical modules. The *column and row modules have each a 4x2 crossbar*, and the *vertical module contains only two interconnection points*, as can be seen in Figure 3.20(b). These smaller modules optimize the area and power consumption, compared to the 6x6 and 7x7 crossbars [13]. Note that the communication links from the vertical module towards the row and column modules are unidirectional, see Figure 3.20(a). This is because guided flit queuing already segregated the packages at the entrance of the router. Thus, no packages need to flow from the row and the column modules towards the vertical module, given that the packages are correctly separated. For example, a package that needs to travel to another layer is first routed via the vertical module upwards or downwards. When arrived at the correct layer, the package is routed to the exit of the row or column module, and then it continues its path towards its destination (in that layer). Conversely, if the package already arrived at the proper layer and destination (router) then it takes the ejection exit of the vertical module, which is connected to the PE.

The high-level architectural overview of the 3D DimDe router is shown in Figure 3.20(b). The arbitration complexity per router is reduced by splitting up the arbiter into an intra-layer and a global arbiter. The intra-layer arbiter handles the local requests within a crossbar per layer, and a single global arbiter (per vertical link / bus) handles the requests from the intra-layer arbiters.



(a) The 3D DimDe Router Architecture, the column and row modules have each a 4x2 crossbar, and the vertical module contains only two interconnection points. The exits indicated by 'early ejection' and ' ejection from up-down' are connected to the Processor Element (PE) of that router.



(b) Overview of the 3D DimDe NOC Architecture. 'Ejection' at the vertical module indicates the connection towards the Processor Element (PE).

Figure 3.20: 3D DimDe NOC Architecture. [13]

The decoupling of arbitration policies helps parallelize tasks; while flits arbitrate locally in each layer, the global arbitration is controlling the vertical paths.

The 3D DimDe crossbar is at least 48% and 36% more area and power efficient, respectively, compared to the smallest crossbar (6x6) [13], as can be seen in Table 3.5. It is because the 3D DimDe uses smaller decomposed crossbars. Furthermore, simulation results with (real) commercial and scientific traffic patterns / workloads show that the 3D DimDe design offer *average latency and throughput improvements of more than 20%* [13] because the 3D DimDe contains two bi-directional inter-layer busses, and the 6x6 and the 7x7 only have one bidirectional and two unidirectional busses, respectively. Conclusively, the crossbar of a 3D DimDe router is at least 48% and 36% more area and power efficient, respectively. Furthermore, it offers better latency and throughput of more than 20% compared to the 6x6 crossbars.

Table 3.5: Area and power comparisons of the crossbar switches. The power is measured with 50% of the switching activity at 500MHz, the used TSV diameter is not indicated. [13]

	Area (μm^2)	Power (mW)
4x2 Crossbar(for 3D DimDe)	*3039.32*	*1.63*
5x5 Crossbar(Conventional 2D Router)	8523.65	4.21
6x6 Crossbar(3D NOC-Bus Hybrid)	11579.10	5.06
7x7 Crossbar(3D Symmetric NOC Router)	17289.22	9.41

In this section the answer for the NOC topic of research question four is found and presented below. All partial answers on research question four, which were presented in this chapter are concatenated and again presented in Section 5.2.

Research question four: What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, memory-on-logic, and *NOC*?

The potential of 3D Network-On-Chip (NOC) is that it enables the creation of a true physical 3D NOC topology, where the router and the network are themselves three-dimensional entities. The advantage between the vertical and the horizontal interconnects is that in the vertical direction, a TSV is just a few tens of μm long as compared to a few thousand μm for the horizontal direction [13]. Hence, the latency of a TSV can be neglected, compared to a global 2D wire. Three aspects are important. Firstly the total number of 3D routers in a network are important to reduce the energy and area. [12] shows that a full 3d connected network is not (always) necessary, and obtained energy and area reductions of 5% and 8%, respectively. However, it induced a minor latency increase of an average of 8%. Secondly, the use of an inter-layer communication bus is important because it can bridge multiple layers in a single-hop. Thirdly, crossbars scale very inefficiently, and with the uses of guided flit queuing the area, power and latency of a crossbar reduces with 48%, 36% and 20% (compared to a 6x6 crossbar), respectively. Another potential of 3D integration is the uses of heterogeneous technology. It enables, for example, the separation of photonic and electrical planes, and it is beneficial to limit (sharp) angles in the waveguide and the use of Photonic Switching Element (PSE).

4 Practical work

This chapter describes the practical work done for this thesis, where a novel 3D scheme is evaluated. The scheme stacks two (or more) 2D processors on top of each other, where the Functional Unit (FU) boundaries between all processors are removed. Thus, a processor can execute instructions on all the unutilized FUs of all the (remaining) processors. The unutilized (free) FUs on other processors can be utilized for fault coverage or for performance improvement. This chapter investigates the impact of this scheme.

In Section 4.1 the problem statement is presented. Subsequently, the solution and implementation are presented in Section 4.2. Thereafter, the experiments and the result analyze are presented in Section 4.3. The chapter concludes with a discussion of related work in the Sections 4.4.

4.1 Problem statement

Higher demands for computation power led to an increase of Processor Elements (PEs) on a chip. It is desirable to be able to use every unutilized Functional Unit (FU) on-chip, even if the FU belongs to another processor. By stacking two processors on top of each other instructions can be issued from one processor to another (see Figure 4.1(b)). In this work a novel 3D scheme is evaluated. The scheme stacks two (or more) 2D processors on top of each other, where *the FU boundaries between all processors are removed*. Thus, *a processor can execute instructions on all the unutilized FUs of all the (remaining) processors*. The free FUs on the other processors can be utilized for *fault coverage* or for *performance improvement*. FUs are free whenever an instruction stalls (i.e. by cache misses or register dependencies [65]) or when there are no instructions to be executed.

This work follows the core stacking strategy and it investigates the impact on two 2D stacked processors on top of each other. It is preferred to have two processors on top of each other (3D integration) instead of using two CPUs alongside each other. This is because with 2D technology the wire lengths between the scheduler on CPU 2 and the FUs on CPU 1 are too long, as shown in Figure4.1(a). 3D integration makes the long wires superfluous by the use of (short) TSVs. Furthermore, by the use of TSVs the length of a wire towards a FU on a different CPU can be kept similar, compared to the wire length towards its own FUs. Naturally, in a 2D plane the FUs can be repositioned near the edge towards the other CPU, this results in shorter (2D) wire lengths. However, this repositioning method will not work for three or more CPUs (not scalable). Moreover, it requires a high redesign effort for existing processors and IP-cores. Conversely, with the core stacking strategy this large redesign effort is not needed, this shows the advantages of this strategy.

The research question that is answered in this chapter is stated below.

> Research question five:
>
> *What is the impact on the performance and fault coverage if two stacked processors share their functional units?*

The 'impact' on the performance from research question five is defined as the speedup a program achieves when it is executed with the improved architecture, compared to the normal execution cycle time. Furthermore, the 'impact' for the fault coverage from research question five is defined as the ratio of instructions executed redundantly (for partial error detection), compared to all the instructions that need a FU.

This work targets only uni-directional instruction issuing for two stacked processors due to time limitations and the current structure of the 3D SimpleScalar (3D SS) simulator, which is an adapted version of the (2D) SimpleScalar (revision 3.0d). Thus, only one of the two CPUs can issue an instruction onto the other CPU. Bi-directional instruction scheduling and stacking of more than two processors is left for future work. In this chapter only CPU 2 can issue instructions to CPU 1, unless it is explicitly stated different.

Issuing an instruction to a free FU of a different CPU is done for two purposes. The first strategy aims to *improve the performance (throughput)*, where free FUs of CPU 1 are utilized for instructions that cannot find a free FU on CPU 2. This strategy has the advantage that an instruction can be executed on any free FU on both CPUs (uni-directional). Without the use of extra (dedicated) FUs performance improvement is achieved. Hence, this strategy is called the *performance mode*. The second strategy is called the *redundancy mode*, it uses unutilized (free) FUs of CPU 1 to redundantly execute instructions running on CPU2 and the results are compared for fault detection. Hence, this scheme *provides partial error detection without the need of extra dedicated FUs*. The instructions that are executed on CPU 2 are concurrently executed on CPU 1 if and only if an appropriate FU of CPU 1 is free. Otherwise, the instruction is only executed on CPU 2 and not on CPU 1, offering no redundancy and hence no error detection. Fault coverage is needed because the susceptibility of microprocessors to transient faults increased, due to of technology scaling. Transient faults (a.k.a soft-errors) are errors that arise by strikes of cosmic particles and radiation from packaging materials, and result in degraded reliability. Thus, this scheme is beneficial for devices that need a higher reliability factor than regular devices.

4.2 Implementation

The simulation environment used at this research is presented in Section 4.2.1. Thereafter, the basics of the chosen simulator (SimpleScalar) is presented in Section 4.2.2, and these are necessary to understand the adaptations of the simulator. Subsequently, the adaptations to the 3D SS simulator are presented.

(a) Two CPUs on a 2D plane, the instructions are issued from the left CPU (CPU 2) to the right CPU (CPU 1) via long wires.

(b) Two stacked CPUs, the instructions are issued from the bottom CPU (CPU 2) to the top CPU (CPU 1) via (short) TSVs.

Figure 4.1: Logical overview of two connected CPUs where an instruction is issued from one CPU to FUs on another CPU.

4.2.1 The simulation environment

The effectiveness and the impact of the performance and the redundancy mode is evaluated with a cycle accurate simulator '3D SimpleScalar (3D SS)'. 3D SS is an adapted version of the *sim-outorder* simulator from the basic (2D) SimpleScalar (SS) (revision 3.0d) [66]. The *sim-outorder* simulator is the most accurate simulator of the total five simulators supported by SS [14] [67, sl.13].

SS is chosen for this project since there are no cycle accurate 3D simulation tools publicly available. Many other articles also have chosen to adapt SS, as can be seen in Appendix I. It shows that SS has more benefits than other simulators, in terms of adaptability of the source code. Moreover, thorough knowledge about SS was already present within the Delft University of Technology and the 3D SS will be used as basic platform for future research work.

4.2.2 Sim-outorder basics

The pipeline used in sim-outorder is depicted in Figure 4.2, as (briefly) presented by the designers of SS in [14]. However, the whole pipeline is only used for timing evaluation, the functional simulation is only performed at the fetch and dispatch stages.

The main simulator loop is implemented in the function *sim_main()* whose main structure is outlined in Figure 4.3. The functions use the pipeline stages, and they are executed in reverse order with respect to Figure 4.2. This is done to eliminate pipeline relaxation problems, and thus no extra temporary variable is needed to transfer the output data towards the next block. 3D SS simulates multiple CPUs with private memories. The *sim_main()* function contains the main simulator loop, which simulates the activities of every CPU at every clock cycles in a serial fashion. Thus, each CPU executes one instruction (machine cycle) and thereafter another CPU is allowed to run a single instruction and so on.

4.2.3 Implementation

Our simulator implementation supports the redundant and the performance modes. This is implemented in 3D SS, and the desired mode can be selected via a command line option. Both modes use a common function that searches for a free FU on a different CPU. This section presents these implementations in more detail.

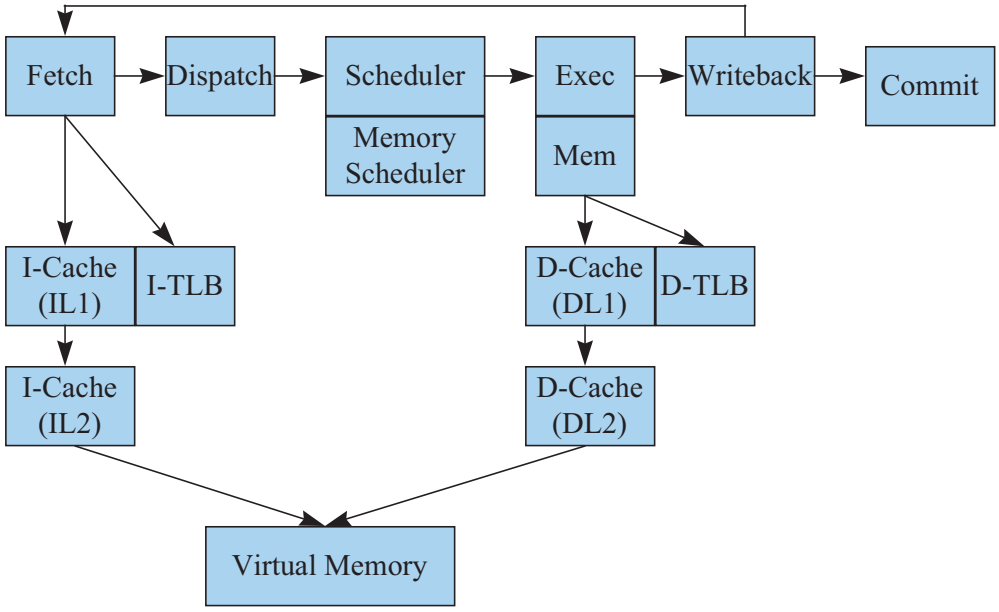


Figure 4.2: Schematic view of the pipeline used by the *sim-outorder* simulator. The FUs use data from the execution queue. [14]

```
sim_main()
{
        ruu_init()          /*ruu = register update unit*/
        for(;;)
        {
                ruu_commit();
                ruu_writeback();
                lsq_refresh();
                ruu_issue();
                ruu_dispatch();
                ruu_fetch();
        }
}
```

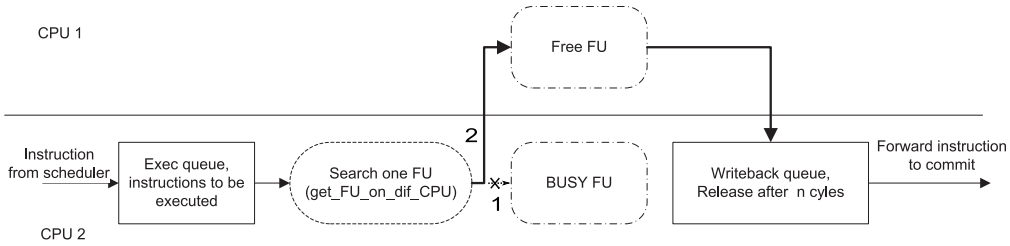
Figure 4.3: The basic structure of the main simulator function (*sim_main*) of the *sim-outorder* in SS.

4.2.3.1 Redundant and performance modes

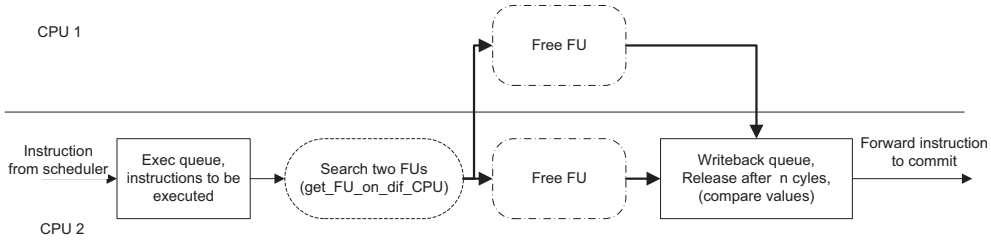
In this paragraph, the basic idea is presented for the redundant and performance modes. In the *performance mode*, if an instruction needs a FU and the desired FUs on the current CPU (CPU 2) are busy then a free FU (of the appropriate type) is searched on the other CPU (CPU 1), see Figure 4.4(a). Whenever CPU 1 has a free FU, the instruction is issued to it. Otherwise, the instruction is re-scheduled in the execution queue. Instructions looking for a free FU on

a different CPU is hereafter denoted by *'candidate instructions'*, since these instructions are a candidates to be executed on another CPU. This holds for both modes, the redundant and performance modes.

In *redundant mode*, all the instructions are executed on the current CPU (CPU 2),and in addition they try to execute redundantly on CPU 1, when possible, as shown in Figure 4.4(b). Thus, whenever a candidate instruction finds a free FU on the other CPU (CPU 1) then its copy is redundantly executed on that FU. Otherwise, when a candidate instruction does *not* find a free FU on CPU 1, it only executes on CPU 2, without error detection.



(a) An instruction path in performance mode if the FU on CPU 2 is busy.



(b) An instruction path in redundancy mode when the FU on CPU 1 is free.

Figure 4.4: The path from the exec queue to the writeback queue is indicated for the performance and the redundant mode. The bold arrows indicated the different paths between the two modes and the function name is placed between parenthesis.

4.2.3.2 Command line

At the *sim_outorder* simulator an extra command line option is added, which enables the user to choose between three modes: (1) normal, (2) redundant and (3) the performance mode. The command '-3d:FUmode *mode*' enables one of the three modes, and thus the command '-3d:FUmode *performance*' enables the performance mode. However, if the FU mode command option is not given, the normal mode is enabled by default. A full 3D SS command line example is given below, it executes 3D SS with two CPUs and two programs (*program1* and *program2*) in *performance mode*. The command '-3d:cpus 2 program1 ' selects two stacked CPUs, and it indicates that the first CPU executes should 'program1'. The next command '-3d:NextCPUInput program2' determines that 'program2' should be executed on the second CPU.

./sim-outorder *-3d:FUmode performance* -3d:cpus 2 program1 -3d:NextCPUInput program2

4.2.3.3 The search FU function

To search a free FU on a different CPU a single function '*get_FU_on_dif_CPU*' is implemented. This function is used for the performance and redundant modes. The basic flow chart of this function is given in Figure 4.5. This function searches a free FU of the required type. As stated previously, the search for free FUs is uni-directional, and thus CPU 2 can issue instructions on CPU 1, but CPU 1 cannot issue instructions on CPU 2. This is due to time limitations and the current structure of the basic 3D SS program. However, when desired and if the 3D SS is adapted the implemented function of this project can perform bidirectional searches, by changing the first if statement from Figure 4.5. The adaptation of the basic 3D SS is out of the scope of this project, and left for future work.

Whenever the function '*get_FU_on_dif_CPU*' finds a free FU on a different processor it returns a pointer to that FU, but if no free FUs are found then a NULL pointer is returned. When a candidate instruction finds a free FU on a different CPU then the function *ruu_issue* makes that FU busy for n issue latency cycles. Subsequently, that instruction is passed to the writeback queue of the current CPU, which releases the result after x operation latency cycles.

4.3 Experiments

In Section 4.3.1 the experimental methodology is presented. Subsequently, the results are presented and analyzed in Sections 4.3.2 and 4.3.3.

4.3.1 Experimental methodology

The simulator models two identical processors on top of each other, and each processor runs their own kernels / benchmarks independently. However, only CPU 2 can use the FUs of CPU 1 and 2. The configuration of the stacked processors is shown in Table 4.1. Each experiment runs two benchmarks, one per CPU, as indicated in Table 4.2. In this chapter it is assumed that 'benchmark 1' and 'benchmark 2' execute on the simulated CPU 1 and CPU 2, respectively. The same experiment is executed multiple times, with different numbers of Integer ALUs (IALUs). This is done to evaluate the impact on the performance and fault coverage, as defined by the research question. The number of simulated IALUs per CPU is varied because the selected experiments contain more than 90% of integer instructions and it is interesting to see the effects on both schemes. The benchmarks are predominantly from the MediaBench consortium [68] because the project (of the Delft University of Technology) that is supporting this work focuses on multimedia applications.

The following benchmarks are compiled to SS executables and used for the simulation: cjpeg, djpeg, mpeg2decode, mpeg2encode, fibonacci, mult_matrix_200×100, mult_matrix_20×10, SAD (sum of absolute differences) and add_images. Djpeg decodes a JPEG file to various file formats, the output format bitmap (BMP) is chosen for this experiment. Cjpeg, performs the inverse operation of djpeg, a BMP file format is encoded to a JPEG file format. Mpeg2encode converts multiple graphical frames (PPM format) into MPEG 2 video.

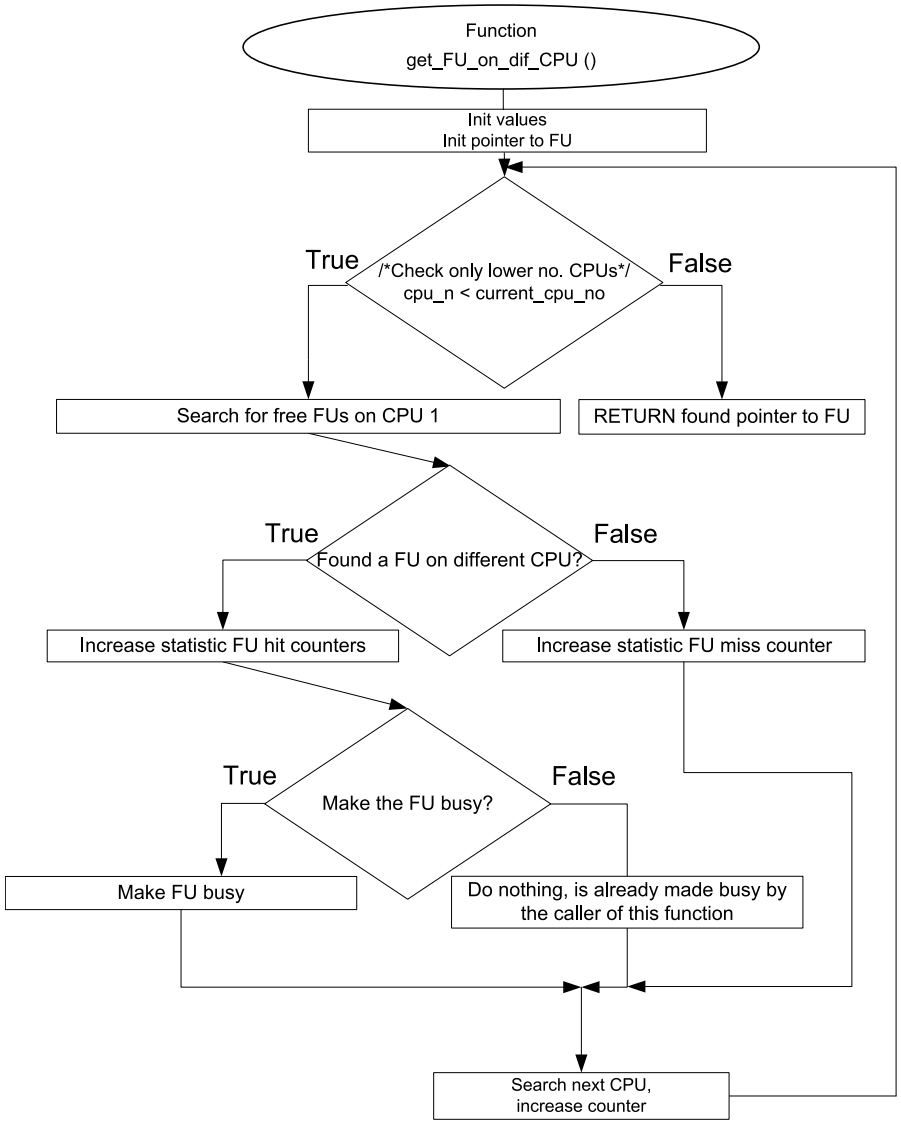


Figure 4.5: Global flowchart of the function get_FU_on_dif_CPU, which checks for free FUs on a different CPU and returns a pointer towards a free FU if available.

Mpeg2decode performs the inverse operation of the mpeg2encode. The fibonacci produces the fibonacci numbers sequence, starting with the numbers 0 and 1, and each remaining number is the sum of the previous two. Mult_matrix_200×100 and mult_matrix_20×10 multiply two integer matrices of the size 200×100 and 20×10 with each other, respectively. SAD (Sum of Absolute Differences) takes the absolute value of the difference between the current pixel and the same pixel in the next frames, and is used for motion estimation in video encoders.

For all the experiments, on CPU 1 the longest benchmark is executed, in respect to the benchmark on CPU 2. This is necessary, otherwise in the performance mode the FUs of CPU 1 would continuously be free. This boundary condition is added due to the uni-directional instruction issuing. However, for demonstration purposes experiment 11 (SAD and fibonacci) is added to

Table 4.1: Processor configuration.

Fetch/Decode/Issue/Commit Width	4
Branch Predictor	Combined, 8K meta-table
BTB Size	1 K, 2-way associative
Return address stack size	32
Branch Misprediction Latency	7
RUU Size	128
LSQ Size	64
# of int. ALUs	1 to 4
# of int. multiplier / dividers	1
# of FP ALUs	2
# of FP multiplier / dividers	1
Memory Latency	112 (first chunk) 2 (subsequent chunks)
L1 Data Cache	32 KB, 2-way set associative
L1 Instruction Cache	32 KB, 2-way set associative
L2 Unified Cache	512 KB, 2-way set associative
3D FU mode	performance and redundant
# of stacked CPUs	2

the results, where CPU 1 terminates before CPU 2 terminates. This experiment shows that a very high speedup / redundancy appears when the boundary condition is not obliged. It is because CPU 1 is idle while CPU 2 is still running, thus all the FUs of CPU 1 are available to CPU 2.

4.3.2 Experimental results

In this section the results are presented obtained for both the performance and the redundancy modes.

4.3.2.1 Performance mode

Figure 4.6 shows the speedup, candidate instruction ratio, and the hit rate of the candidate instructions for the benchmarks on CPU 2 in performance mode. The performance results of CPU 1 are not shown, since CPU 1 does not issue instructions to CPU 2 (the experiments are unidirectional). Furthermore, the slowdown of CPU 1, due to the instructions issued from CPU 2, is on average 0.15% with a maximum of 0.37%, which is considered not significant and thus neglected. We explain the little slowdown of CPU 1 by the fact that 90% of the executed instructions need the IALU unit, whose issue latency is one cycle and thus it does not cause any stalls. Each experiment has a group of four bars. Each bar depicts the result of the same experiment, but with different number of IALUs. For some experiments three or less bars are visible because the value of the non-visible bars is nearly zero.

Figure 4.6(a) shows the speedup of benchmark 2 (executed at CPU 2), in respect to the 'normal' execution time. The speedup is calculated with Equation (4.1), where CPU_n_{nor} and CPU_n_{perf} denote the number of total cycles that CPU_n needs to execute a benchmark in

Table 4.2: Abbreviations of the experiments that are presented in this chapter.

Experiment name	benchmark 1 at CPU 1	benchmark 2 at CPU 2
Expr. 1	cjpeg	cjpeg
Expr. 2	cjpeg	djpeg
Expr. 3	djpeg	cjpeg
Expr. 4	djpeg	djpeg
Expr. 5	mpeg2decode	mpeg2encode
Expr. 6	mpeg2decode	mult_matrix_200x100
Expr. 7	mpeg2encode	add_images
Expr. 8	fibonacci	SAD
Expr. 9	mult_matrix_20x10	fibonacci
Expr. 10	add_images	cjpeg
Expr. 11	SAD	fibonacci

normal mode and in performance mode, respectively.

$$Speedup_{cpu\_n}(\%) = \frac{CPU\_n_{nor} - CPU\_n_{perf}}{CPU\_n_{nor}} \cdot 100 \tag{4.1}$$

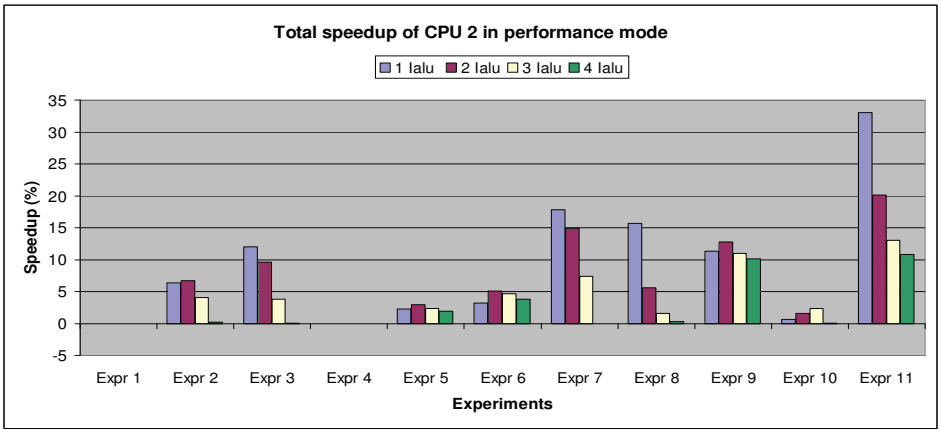
The ratio of candidate instructions is shown in Figure 4.6(b), it denotes the ratio of the total instructions from benchmark 2 that cannot execute on its own CPU (CPU 2) and try to execute on the FUs of CPU 1. The hit rate of all the candidate instructions is depicted in Figure 4.6(c), it shows the percentage of the candidate instructions that succeeded to execute on CPU 1.

4.3.2.2 Redundant mode

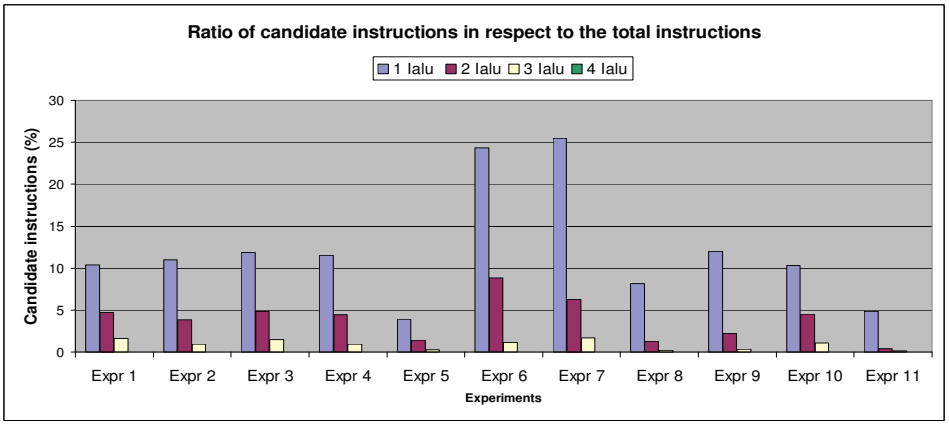
The ratio of instructions, from CPU 2, that succeeded to execute redundantly on CPU 1 is indicated in Figure 4.7. The results of Figure 4.7 are grouped per experiment and each group has four bars. Each bar runs the same experiment with a different number of IALUs. Figure 4.7 illustrates only the redundant results of the benchmarks at CPU 2, since CPU 1 does not issue instructions to CPU 2 (the experiments are uni-directional). As expected, zero percent of speedups has been observed for CPU 2 because CPU 2 executes all the instructions as in the normal mode, and thus no speedup graph is given. Furthermore, the performance reduction at CPU 1, due to the instructions issued from CPU 2, is negligible and is on average 0.09% with a maximum of 0.24%. The reason is the same as in the performance mode case: 90% of the executed instructions need the IALU unit, whose issue latency is one cycle and thus it do not cause any stalls. Therefore, it is unnecessary to depict the decrease in speed of CPU 1.

The candidate instructions constitute on average 82% out of the total instructions, which are executed on CPU 2. The remaining executed instructions are memory access instructions (load, store), they do not need FUs and thus they are not executed redundantly. At a FU miss in the redundant mode, no candidate instructions are re-scheduled but they are discarded, since the instruction is executed by default on CPU 2.

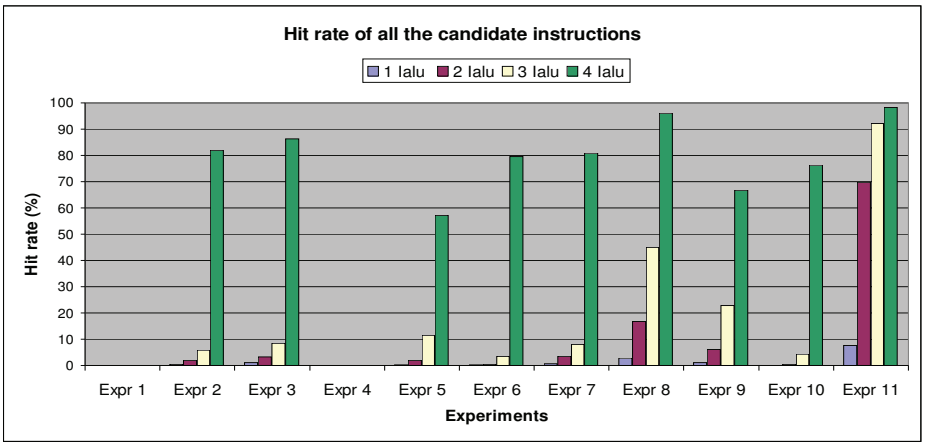
The hit rate of all the candidate instructions show how many percent of the total candidate instructions are issued (find available FUs) to CPU 1. The hit rate ratio has exactly the same percentage as the redundancy ratio, since no instructions are re-scheduled and all instructions



(a) Speedup of CPU 2 when it is possible to issue instructions to CPU 1.



(b) The percentage of candidate instructions from CPU 2, in respect to the total instructions that need a FU. The bars with four IALUs are not visible because for all the experiments the value is nearly zero.



(c) Hit rate of the candidate instructions. The bars with one IALU are not always visible because the values are approximately 1%.

Figure 4.6: The performance mode results. The experiment abbreviations are introduced in Table 4.2. Note, for all the experiments, except Expr 11, CPU 1 runs longer benchmarks than CPU 2.

that need a FU are candidate instructions. Therefore, it is superfluous to depict a hit rate ratio graph.

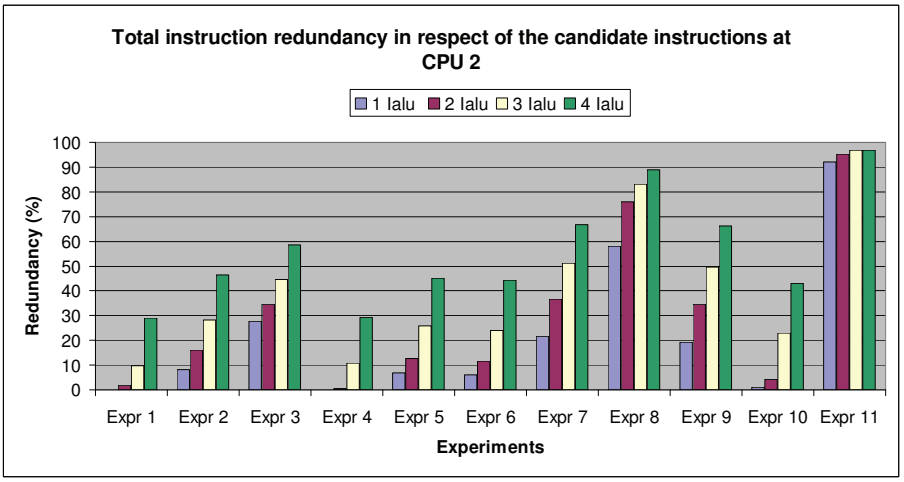


Figure 4.7: The percentage of redundantly executed (candidate) instructions of CPU 2 issued onto CPU 1.

4.3.3 Analysis

In this section the results are discussed for the performance and redundant modes to understand the impact of sharing FUs at both modes. For both modes, the factors are discussed that influence performance and reliability. Subsequently, it is explained why the performance of CPU 1 does not suffer much when executing extra instructions from CPU 2. The discussion continues by analyzing the highest, lowest, and the average speedups / fault coverage. As previously stated, it is assumed that 'benchmark 1' and 'benchmark 2' execute on the simulated CPU 1 and CPU 2, respectively.

4.3.3.1 Performance mode

CPU 1 does not suffer significantly from the extra instructions executed on it, on average its performance decreases 0.15% with a maximum of 0.37%. This is because approximately 90% of the instructions are integer computational instructions to be executed on IALU, whose issue latency is one cycle. Therefore, there are no extra stalls introduced, since in the next cycle the FU is released. However, when an instruction is issued to CPU 1, whose issue latency is longer than one cycle, such as floating point and integer divisions (12 and 19 issue latency cycles, respectively) then instructions might stall the normal benchmark execution on CPU 1.

Conclusively, the performance of CPU 1 does not suffer from any drawbacks because the mediabench benchmarks use many IALU instructions. The drawback of the performance scheme is that the speedup is data dependent, and no guarantee can be given that CPU 1 does not suffer for other workloads due to the instructions from CPU 2.

In the performance mode the speedup depends on the hit rate and the number of candidates. The hit rate indicates the percentage of the candidate instructions that finds free FUs on CPU 1. Furthermore, the hit rate between two different experiments can vary because it depends on the benchmark running at CPU 1. If CPU 1 runs a benchmark with many stalls, due cache misses or dependencies, then the FUs are often free and thus the hit rate is high. The candidate instruction ratio denotes the ratio of the total instructions from benchmark 2 that cannot execute on its own CPU (CPU 2) and try to execute on CPU 1. The number of candidate instruction depends on the Instruction-Level Parallelism (ILP) of benchmark 2, if the ILP is high then there are more candidate instructions. The candidate instruction *ratio* increases if the hit rate is low because the candidate instructions that have a FU miss are *re-scheduled* into the execution queue of CPU 2 and they might become a candidate instruction again. Thus, the number of candidates naturally increases. A possible reason for having many FU misses on its own CPU is when benchmark 2 has a high ILP, but a limited number of FUs on CPU 2 to issue the instructions simultaneously. For example, the hit rate is low in experiment six for the configuration with one IALU is, but it has a high candidate instruction ratio (see Figure 4.6(b) and Figure 4.6(c)), and the inverse holds for the configuration with four IALUs (high hit rate and little candidate instructions).

The achieved speedups per experiment and configuration are presented in Figure 4.6(a). The speedups of different IALU configurations cannot be mutually compared. Looking only at the speedup, the configuration with one IALU achieves higher speedups than configurations with four IALUs. However, the benchmarks with one IALU terminate much later than with four IALUs. For example, experiment one with one IALU terminates after 8 million cycles and for four IALUs after 3 million cycles. This is because with one IALU there are more candidate instructions, see Figure 4.6(b). That is why the speedups should not be compared among the different IALU configurations.

Experiment 11 has the greatest speedup because benchmark 1 running on CPU 1 terminates earlier than benchmark 2 at CPU 2. Thus, this will increase the hit rate because the FUs on CPU 1 are all free after benchmark 1 terminates. However, this experiment represents an ideal case, since the author of this thesis does not expect that in reality CPU 1 is (completely) idle. Therefore, we will look at the best realistic example, from Figure 4.6(a). Experiment seven (mpeg2encode and add_images) has the *highest speedup* for 1 and 2 IALUs. High speedup is mainly due to the very high number of candidate instructions (high ILP), since the hit ratio is only slightly higher, compared to the other experiments. The higher number candidate instructions can be explained by the fact that add_images, running on CPU 2, has a few dependencies (adding two individual pixels) and thus it needs many IALUs simultaneously (large ILP). The combination of a high candidate instruction ratio with a hit rate ratio is important for the speedup.

Experiments one (cjpeg and cjpeg) and four (djpeg and djpeg) have the *lowest speedup* (0%) for all the IALUs (one until four). It is because both CPUs run the same workload, the intensive computation periods coincide (large ILP) at the same moment on both CPUs, keeping all the FUs busy on both CPUs.

The *average speedups* (without experiment 11) are 7%, 6%, 4% and 2% for the configurations with one to four IALUs, respectively. The lowest average speedup of 2% for the configuration with four IALUs seems small. However, it is because the candidate instructions are nearly 0%. This can be expected, since there are sufficient IALUs available (four units) and thus most of the IALU instructions of benchmark 2 can be executed on CPU 2. The few candidate instructions that need a free IALU on CPU 1 can easily find one (high hit rate), since there are

also sufficient IALUs. The inverse case holds for the highest average result of 7% for the configuration with one IALU (many candidate instructions and low hit rate). It is because there is only one IALU, and thus the IALUs on CPU 2 become easily saturated. That is why the candidate instructions ratio is high. Furthermore, the IALUs of CPU 1 are also usually busy and easily saturated, resulting in a low hit rate.

Conclusively, the speedup is only 2% for the configuration with four IALUs, but that is due to the small numbers of candidate instructions (limited ILP at benchmark 2). Given the high (77%) average hit ratio, higher speedups for this configuration are expected when workloads with more ILP are available on CPU 2, resulting in more candidate instructions. The other IALUs configurations have longer execution times, compared to the configuration with four IALUs. However, due to the higher candidate instruction ratios they achieve more speedup, compared to the normal execution times. This performance scheme is beneficial, since speedup is achieved without extra FUs because the boundary between the FUs of the CPUs is removed.

4.3.3.2 Redundant mode

The speed reduction for CPU 1 is negligible, over the total execution time reduced with on average 0.09% and a maximum of 0.24%, due to the execution of extra instructions from CPU 2. Furthermore, no guarantee can be given that CPU 1 does not suffer more, due to extra issued instructions from CPU 2. As expected, all the experiments on CPU 2 show 0% speed reduction or increase, since CPU 2 runs all the same instructions as it runs the normal mode.

The redundancy ratio represents the fault coverage for the instructions executed at CPU 2 and it is equal to the FU hit ratio, since no instructions are re-scheduled and all instructions that need a FU on a different CPU are candidate instructions. Thus, the number of free FUs on CPU 1 is the only limiting factor, since there are many candidate instructions.

The individual redundant ratio results are depicted in Figure 4.7. Experiment eight (fibonacci and SAD) shows the *highest redundancy ratio* for all four configurations. This shows that benchmark 1 (fibonacci) on CPU 1 contains many stalls (keeping FUs often free), since the fibonacci sequence has many dependencies. Furthermore, it corresponds to the hit ratio of the performance mode, see Figure 4.6(c), where experiment eight also has the highest hit ratio at all the configurations, compared to the other experiments.

Experiment one (cjpeg and cjpeg) has the *lowest redundancy ratio* with 3 and 4 IALUs. The low redundancy / hit ratio indicates that benchmark 1 on CPU 1 keeps the IALUs busy because of a high ILP. Moreover, both CPUs run the same workload, the intensive computation periods coincide at the same moment on both CPUs, keeping all the FUs busy. This is confirmed by the results of experiment four (djpeg and djpeg) were also the same benchmark is executed on both CPUs, and the redundancy results are also very low. Conversely, the redundancy results are much higher at experiment two (cjpeg and djpeg) and three (djpeg and cjpeg), where different benchmarks are executed on different CPUs, see Figure 4.7.

Conclusively, the *average number* of instructions executed redundantly (excluding experiment 11) are 15%, 23%, 35% and 52% for the configurations one to four IALUs at CPU 2, respectively. These results shows the advantage of this proposal, especially for the configuration with four IALUs. The error detection is achieved at low cost (additional control logic and TSVs), but without any extra dedicated FUs.

4.4 Related work

Currently (early summer 2010) 3D integration has received a lot of research attention. In this section various known proposals are discussed and the difference between our proposal is indicated.

4.4.1 Performance mode

Articles are known that propose the general idea of stacking two processors on top of each other [8, 17, 69], or where a single processor is divided over two layers [11]. However, [8, 11, 17, 69] do not remove the boundary between the FUs of different processors. Conversely, our proposal *removes the boundary between the FUs* of two (or more) stacked processors to improve the performance of a processor in the stack. To the knowledge of the author of this thesis there are no articles or proposals known that issues instructions from one CPU to the FUs of another CPU in the 3D stack to improve the execution time. This proposal is novel because the boundary between the executional units of two (or more) processors is removed, and thus the throughput and hit rate is increased, which speeds up the execution time of a benchmark. Moreover, the performance improvement is achieved without any dedicated FUs.

4.4.2 Redundant mode

One solution, to create a more reliable processor is known, [70] uses the dynamic implementation verification architecture (DIVA) approach which uses a 2D checker core next to leading computational core on a single. [71] applies the DIVA approach to 3D technology, and it stacks the checker core on top of the leading computational core of the main processor to overcome wiring complexity, cycle time (wire lengths) and area, compared to the DIVA approach. However, our proposal does not need an extra dedicated core when two or more processors are stacked. Our proposal provides partial fault coverage and that leads to a reduction in cost, compared to full coverage. Another proposal is known ([65]) that is offering a partial fault coverage with a single CPU. [65] runs on a single CPU redundant instructions when the resources are unutilized. Thus, if the first IALU is free then the second IALU can run a redundant copy of its instruction on the first IALU. Our proposal also runs redundant instructions when the resources are unutilized, but does this on two separate processor planes, which reduces the redesign efforts and wire complexity, compared to [65]. Furthermore, our proposal is scalable and thus it can be used with multiple stacked processors without the large 2D wire overhead, compared to multiple processors on a planar plane.

To the knowledge of the author of this thesis there are no proposals known that issue instructions from one CPU towards another CPU (in a 3D stack) in order to achieve error detection. This proposal is novel because the boundary between the FUs of two (or more) processors is removed, and it offers possibilities to increase the reliability without the need of dedicated FUs.

In this chapter the answer on research question five is found.

Research question: What is the impact on the performance and fault coverage if two stacked processors share their functional units?

The impact on the performance is on average 7%, 6%, 4% and 2% for the configurations with one to four IALUs, respectively. The speedup is only 2% for the configuration with four IALUs, due to the small numbers of candidate instructions (low ILP). Given the high (77%) average hit ratio, higher speedups for this configuration are expected when workloads with more ILP are available on CPU 2, resulting in more candidate instructions. The other IALU configurations have longer execution times, compared to the configuration with four IALU. However, due to the higher candidate instruction ratios they achieve higher speedups, compared to the configuration with four IALU. This performance scheme is beneficial, since no extra dedicated FUs are needed and still speedups are achieved because the boundary between the FUs of the CPUs is removed.

For the fault coverage, the average number of instructions executed redundantly are 15%, 23%, 35% and 52% for the configurations one to four IALUs at CPU 2, respectively. These results show the advantage of this proposal, especially for the configuration of four IALUs. This redundant scheme is beneficial, since no extra dedicated FUs are needed and still fault coverage are achieved at low cost (only additional control logic and TSVs). This is because the boundary between the FUs of the CPUs is removed.

5 Conclusion

In the next section, Section 5.1, a summary of the whole thesis is presented. Subsequently, Section 5.2 presents the answers on the research questions, which were previously presented in this thesis. Thereafter, Section 5.3 presents answer on the main research question. Section 5.4 concludes this thesis and presents the recommendations.

5.1 Overall summary

This section summarizes the whole thesis, which is divided in three parts: (1) manufacturing, (2) architectural potential and impact, and (3) practical work.

5.1.1 Manufacturing

For an architectural designer it is important to understand the constraints and properties of the inter-layer interconnect. This knowledge allows a designer to understand the implications of its 3D design and thus if it is a *practical and useful* architectural design. That is why in this work we look at the *manufacturing part* for 3D integration, and in specific the inter-layer interconnects. The answers on the three research questions in Chapter 2 have as goal to *select the best inter-layer interconnect, and indicate its properties*.

In Chapter 2, 3D monolithic structures and 3D stacked structures are presented. The main difference between the approaches is that the monolithic approach has a start wafer where multiple silicon layers are fabricated upon. Conversely, the 3D stacked approach contains multiple single silicon wafers bonded together to form a multi-silicon wafer (or die).

A *3D monolithic device* is sequentially fabricated from the bottom layer up. There is only one main substrate, with one or more transistor layers on top. The transistor layers are divided by an isolation layer. There are two main approaches to manufacture a 3D monolithic device, which is the *layer-by-layer approach* and the *simultaneously multi-layer approach*. The layer-by-layer approach fabricates the MOSFETS per layer. Conversely, the multi-layer approach manufactures all the MOSFETs layers simultaneously. This process experiences the same temperatures as with the fabrication of a 2D plane.
One of the critical steps in the *3D monolithic manufacturing process* is forming a high-quality active silicon layer on top of the isolation layer. The upper silicon layer is manufactured with laser crystallization or with seed crystallization. However, unavoidable thermal steps are needed to form the upper silicon layer and MOSFET layer, such as gate oxidation [1]. The underling layer degrades (i.e. unwanted doping diffusion), due to the high temperatures. Therefore, a tight thermal budget must be imposed, resulting in a low manufacturing throughput. The heat shields between the lower and upper layer, used at fabrication time, have a negative effect on the heat dispersement during normal operation and this results in high temperatures in the upper

planes [2]. It is challenging to build more than three device layers on top of each other with the 3D monolithic approach. Furthermore, if a manufacturing fault occurs at one of the layers, then the whole die is lost. This results in lower yield. Most of the manufacturing technologies are sequential, which results in long manufacturing time.

The main challenge to form these 3D monolithic structures are: forming high-quality silicon film beyond the first device layer, the tight thermal budget, and methods to contact the various device layers.

A basic 3D stacked structure uses two or more individual processed wafers, which are grinded, aligned and then bonded. *The wafers are fabricated individually, and therefore it resolves the thermal budget problem (first problem)*, compared to the 3D monolithic approach. Furthermore, *wafer bonding can endlessly be done, and thus the second main problem is solved.* Moreover, *six different interconnect methods are possible for communication between the layers*, which solves the third problem. *Thus, 3D stacking solves the three problems from the 3D monolithic approach.*

The interconnects that were presented are the capacitive, inductive, TSV, micro bumps, Package-On-Package (POP) and wire bonded interconnects. A ranking table ranked these interconnects on the pitch, speed power consumption and maturity, and it shows that the TSVs and the micro bumps have good qualities. However, the disadvantage of micro bumps is that no more than two layers can be used. Conversely, two or endless layers are possible with TSVs and thus *TSVs have the most potential to become the 3D interconnect between layers*. Furthermore, four individual consortia see potential in 3D stacking with TSVs and are all focused on developing (affordable) TSV wafers. TSV wafers under $150USD per wafer are demonstrated.
Conclusively, the 3D stack approach and the TSVs as interconnect are (highly likely) going to be used in the (near) future for 3D silicon integration.

The latency of a TSV is similar to a global 2D on-chip wire, but TSVs are much shorter. Thus, the latency of a TSV can be neglected. In general, the success of the simultaneous power reduction and speedup in a design lies in *wire length reduction* compared to 2D wires. The area a TSV occupies depends on the pitch, the (current) smallest known pitch is 7μm and the area is similar to 17 CMOS gates with 45nm technology. Thus, vertical routing should be restrained, due to the size of a TSV [20, 49]. However, the use of TSVs remains a tradeoff between area (cost) and performance. Power reduction is possible when long 2D on-chip wires are replaces by short TSVs [49, sl.9]. Articles are known that reduced the power consumption by 5%-20%. Faulty TSVs can have many causes and occur during fabrication time or during normal operation. Yield and reliability can be improved by the use of redundant TSVs or with Time Division Multiplexing (TDM) schemes. The redundant TSV scheme uses a number of spare TSVs for every cluster of TSVs. This results in a small area overhead, although it improves the yield significantly. A Time Division Multiplexing (TDM) scheme shares one TSV for two or more signals and therefore no extra redundant TSVs are needed (see Fig. 2.22(a)). However, sharing one TSV with multiple signals increases the delay and area compared to the redundant scheme. The current characteristic life is low (1216 cycles), but it can be improved with redundant TSVs or with the Time Division Multiplexing scheme.

5.1.2 Architectural potential and impact

In Chapter 3 the potential and impact is shown for memory-on-memory, logic-on-logic, memory-on-logic, and a 3D NOC.

The benefits of 3D stacking include five key advantages: (1) wider and denser (on-chip) interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies [8], and (5) footprint reduction. The 3D stacking strategy can be divided into four main groups, and they are ordered from the coarsest to the finest level: *(1) core stacking, (2) Functional Unit Block (FUB) repartitioning, (3) logic gate splitting, and (4) transistor repartitioning.* For all the stacking strategies, the data bus width can be placed *horizontally or vertically*, and is in this thesis referred to as the *single-layer data approach and the multi-layer data approach*. The presented rule of thumb indicates the relation between the number of layers and the wire reduction for core stacking, FUB stacking, or logic gate splitting. The rule of thumb is as follows: *the wire reduction factor is equal to the square root of the number of layers [59, p.3], and the remaining wire length is equal to the original 2D wire length divided by the wire reduction factor.*

5.1.2.1 Memory-on-memory

Section 3.2 shows that the *access latency, wire length, the power consumption and the footprint of a 3D memory reduces*, compared to a 2D memory with an equal amount of bit storage. This can be done via bank stacking of array splitting. With bank and array stacking the latency variation of a (Non-Uniform Cache Access (NUCA)) cache reduces because the *distance between the closest and farthest bank (inter-bank wire) is reduced*. Furthermore, array stacking provides the main advantage of memory-on-memory, since it also diminishes the memory wall problem by *reducing the intra-bank wire lengths*, which speeds up the access latency of each memory bank.

With bank stacking, the planar 2D memory banks are stacked on top of each other. However, bank stacking only reduces the inter-bank latency. Conversely, memory array splitting reduces the inter-bank and intra-bank latency. In particular, the bitlines and wordlines within a bank are reconfigured. When the memory array is split across the vertical axis it is called column stacking, due to the stacking of the columns / wordlines. Conversely, if the array is split across the horizontal axis then it is called row stacking. Column stacking offers better latency reductions, and row stacking better power reductions. Bank and array stacking reduce the overall footprint, memory access time, and power consumption. The latency and energy reductions for bank stacking is 9.7% and 31.5%, and for array stacking is this 21.6% and 30.4%, respectively. Furthermore, the data can be stored horizontally or vertically, which is in this thesis referred to as *the single-layer data approach and the multi-layer data approach*. No thermal information is known about 3D memory stacking. However, it is known that memory circuits are cooler than logic circuits, and thus logic-on-logic stacking is much more problematic than memory-on-memory. However, Section 3.3 shows that logic-on-logic circuits can handle these temperature problems. That is why the author of this thesis *expects that the memory-on-memory circuits can handle the increase in temperatures* as well, since they produce less heat than the logic-on-logic circuits.

5.1.2.2 Logic-on-logic

The wire length reduction at logic-on-logic stacking can be used to eliminate pipeline stages and / or increase the operating frequency of a logic-on-logic stacked device.

The pipeline stages can be eliminated, since 3D integration compacts a 2D planar design. Therefore, the wires are shorter and thus some pipeline stages become superfluous. Conversely, the shorter wire lengths can be utilized to increase the operating frequency.

Although, 3D stacking is beneficial, dividing a die over multiple layers does *not* automatically results in a higher performance or a lower power consumption. Looking at results of the 16 and 32-bit log shifters from [51], it shows that stacking more than two layers results in marginal benefits. Thus, it is *not always beneficial to split a circuit over more planes*. It is because eventually the circuits become *gate dominated* instead of wire dominated [51].

The temperature impact for logic-on-logic stacking is important to consider, since multiple power dense and thus hot planes are stacked on top of each other. There are three differences between the 2D and 3D thermal situation. The first difference is that the *heat sink area reduces* equally to the footprint reduction. Thus, more heat should be dissipated by a smaller area. Secondly, the *distance between the heat sink and the lower layer is larger*, compared to a planar chip. Thirdly, *the power density increases*, and thus is the maximum temperature higher, compared to the same circuit on a planar chip. Examples are known where the maximum temperature for the 4-layer stack increased with 32.2°C (compared to the 2D processor) to 68.9°C. With a temperature difference of less than 2.5°C between the closest and the farthest layer from the heat sink. The author of this thesis thinks that the *temperature increases significantly*, but it is not four times hotter, at a 4-layer stack, because the *wire reduction diminishes the heat generation*. However, without extra cooling structures it is also possible to achieve a temperature neutral[1] stacked chip. By *scaling the supply voltage and the clock frequency* the power consumption, performance and temperature of a chip are controlled. [11] obtained a *temperature neutral 3D processor*, while achieving power and performance improvements of 34% and 8%, respectively.

5.1.2.3 Memory-on-logic

There are four main advantages for memory-on-logic stacking: (1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies.

The memory-on-logic configuration allows the use of *multiple memory ports at various positions in the design*, which provides parallel access, and that increases the total memory bandwidth [64]. Moreover, each memory port can have a wider data bus width, compared to the 2D situation [2]. The uses of a *larger cache provides a higher cache hit ratio*, which improves the overall performance of the processor [11]. Moreover, 3D caches have a lower power consumption, in comparison to 2D caches with the same amount of bits, due to the wire length reduction.

Another strategy is the use of the *heterogeneous* property of 3D integration, which allows different process technologies (nm) or substrates (SOI or bulk) to be stacked on top of each other. The cache misses are reduced by replacing the original 2D *SRAM cache* for a 2D or 3D *DRAM memory*, since 2D *DRAM contains eight times more bits* than a 2D SRAM plane with

[1] Temperature neutral is with respect to the original temperature of the 2D chip.

[2] No articles are known that presents these memory-on-processor (specific) improvements.

the same footprint [11]. Furthermore, DRAM reduces also the power consumption, compared to a SRAM cache with the same footprint [11]. Even though the main (DRAM) memory can be placed on-chip, the author of this thesis beliefs that an off-chip (DRAM) memory will remain between the processor and the hard disk for speed optimizations. It is because a 3D memory is limited by the number of layers, due to practical manufacturing reasons. For example, 50 layers can give a low yield and is not practical.

A SRAM cache plane is hotter than a DRAM plane [11], since SRAM circuits are power denser than DRAM circuits. Thus, it is better for the temperature to stack a DRAM cache on top of a processor than a SRAM cache. However, even when a hotter SRAM cache is stacked on a processor the temperature increase is less than 5°C, and it reaches a maximal temperature of 92.9°C. [11]. The increase is minor because a heat sink with convectioned air cools the chip.

5.1.2.4 3D NOC

With a 3D Network-On-Chip (NOC) there are three aspects important: (1) the total number of 3D routers in a network, (2) the inter-layer interconnect topology (i.e. bus, or point-to-point), and (3) a full crossbar scales inefficiently. Furthermore, 3D integration enables the uses of heterogeneous technology, which allows the separation of photonic and electrical planes.

It is possible to implement a full 3D connected NOC, where all the routers have connections to other layers. However, [12] shows that *a full 3D connected network is not (always) necessary*. The main advantage of a non-full 3D NOC is a reduction in energy and area. The key factor is to use a *healthy mix between 2D and 3D routers*. However, the drawback is that the latency increases. The simulation results of [12] show that not a particular mix of 2D and 3D routers is the best for reducing area or energy for all the simulated traffic types (uniform, hotspot, and transpose traffic). In general, the *non-full 3D schemes obtained energy and area reductions of 5% and 8%*, respectively. However, they induced a minor latency increase of 8% (on average), and the non-full 3D schemes can only be used at low and medium traffic loads. Otherwise, it results in an increase of the energy consumption and latency.

Another important aspect of the *NOC architecture* is the *inter-layer communication topology between 3D routers*. This can be achieved via two strategies: *(1) multi-hop communication (point-to-point) or (2) via single-hop communication (bus)*. With the multi-hop communication a *package travels multiple hops before it reaches the upper plane*, assuming that it started from the lowest plane. However, the *inter-layer communication latency is dominated (per hop) by the router*, compared to the latency of a (short) TSV. It is because every package is buffered and arbitrated within each router. Conversely, inter-layer single-hop communication via a bus is beneficial, due to the very small inter-layer (TSV) delay, compared to the latency of a router and 2D global wires.

An efficient crossbar is needed, since *crossbars scale inefficiently* [13]. The multi-hop and the single-hop architectures use a 7x7 and a 6x6 crossbar, respectively. The smallest crossbar (a 6x6 crossbar) contains 36 interconnection points per layer. Conversely, the 7x7 crossbar need 49 interconnection points per layer, which is much more, compared to the 36 of the 6x6 crossbar. The improved crossbar scheme reduces the area and power with at least 18% and 17%, respectively, compared to the 6x6 and the 7x7 crossbars. The improved decomposed crossbar is called 3D Dimensionally-Decomposed (DimDe), and uses *guided flit queuing, which decomposes the incoming traffic into three independent streams* before it reaches the crossbar. The traffic streams

are decomposed in a X, Y , and Z stream. This segregation of traffic allows the use of only two smaller 4x2 crossbars and extra two interconnection points. Conclusively, the improved crossbar (3D DimDe) is at least 48% and 36% more area and power efficient, respectively. Furthermore, it offers better latency and throughput of more than 20% compared to the 6x6 crossbars.

The *heterogeneous property of 3D integration allows the separation of photonic and electrical planes*. This separation is desirable, since sharp bends in optical wave guides should be avoided. Photonic interconnects offers ultra-high communications bandwidths in the terabits per second range [18]. Photonic signaling has low power consumptions, since the power consumption of an optical signal at the chip level is independent of the distance. Simulation results indicate that a separate photonic plane provides a high performance per watt, as well as a moderate throughput and latency improvement. However, [18] did not take into account the off-chip laser source, and this result might thus give a false picture.

5.1.3 Practical work

Practical work is done for a research project. This work simulated two identical stacked 2D processors, where *the Functional Unit (FU) boundaries between all processors are removed*. Thus, a processor can execute instructions on all the unutilized FUs of all the (remaining) processors. The free FUs on other processors can be utilized for *fault coverage* or for *performance improvement*. This proposal also holds for two or more (non-)identical processors, which are stacked on top of each other. This work follows the core stacking strategy, since two 2D processors are stacked on top of each other. It is preferred to have two processors on top of each other (3D integration) instead of using two CPUs alongside each other. This is because with 2D technology the wire lengths between the scheduler on CPU 2 and the FUs on CPU 1 are too long. 3D integration makes the long wires superfluous by the use of (short) TSVs. Furthermore, by the use of TSVs the length of a wire towards a FU on a different CPU can be kept similar, compared to the wire length towards its own FU.

Instruction issuing to a free FU of a different CPU is done for two different strategies. The first strategy, the '*performance mode*', aims to improve the performance (throughput), where free FUs of CPU 1 are utilized for instructions that cannot find a free FU on CPU 2. The second strategy, the '*redundancy mode*', uses the free FUs of CPU 1 for redundancy and thus it provides partial error detection without the need of extra dedicated FUs. The instructions that are executed on CPU 2 are concurrently executed on CPU 1 if and only if appropriate FUs of CPU 1 are free. Otherwise, the instructions are only executed on CPU 2 and not on CPU 1. Thus, no redundancy and hence no error detection is offered.

The impact on the performance and the redundancy modes is evaluated with a cycle accurate simulator '3D SimpleScalar (3D SS)'. 3D SS is an adapted version of the basic (2D) SimpleScalar (SS) (revision 3.0d) [66]. In specific, the *sim-outorder* simulator is used, which is the most accurate simulator of the total five simulators supported by Simple Scalar [14] [67, sl.13].

Eleven experiments are executed multiple times, with each time a different number of Integer ALUs (IALUs). This is done to show the impact on the performance and fault coverage between two stacked processors with shared functional units. The number of simulated IALUs per CPU is varied because the selected experiments contain more than 90% of integer instructions, and it is interesting to see the effects on the speedup and fault coverage.

The impact on the *performance* is an speedup of 7%, 6%, 4% and 2% on average for the configurations with one to four IALUs, respectively. The speedup is only 2% for the configuration with four IALUs, due to the small numbers of candidate instructions (low Instruction-Level Parallelism (ILP)). Given the high (77%) average hit ratio, *higher speedups for this configuration are expected when workloads with more ILP run on CPU 2*, resulting in more candidate instructions. The other IALU configurations have longer execution times, compared to the configuration with four IALU . However, due to the higher candidate instruction ratios they achieve higher speedups, compared to the normal execution times. This *performance scheme* is beneficial, since no extra dedicated FUs are needed and still *speedups are achieved* because the boundary between the FUs of the CPUs is removed. For the *fault coverage*, the average number of instructions executed redundantly are 15%, 23%, 35% and 52% for the configurations with one to four IALUs at CPU 2, respectively. This shows the advantage of this proposal, especially for the configuration with four IALUs. This *redundant scheme* is beneficial, since no extra dedicated FUs are needed and still *fault coverage is achieved* at low cost (only additional control logic and TSVs). This is because the boundary between the FUs of the CPUs is removed.

To the knowledge of the author of this thesis there are no proposals known that issue instructions from one CPU towards the other CPU in a 3D stack to improve the performance or fault coverage. This proposal is novel because the boundary between the executional units of two or more processors is removed, and thus the hit rate of finding a free FU is larger, which improves the performance or fault coverage. Moreover, performance improvement and fault coverage is achieved without any dedicated FUs.

5.2 Answers on the research questions

This section presents the answers on the sub-research questions. The answers lead to the answer on the main research question, which is answered in Section 5.3. The sub-research questions are bounded by the boundary conditions of Section 1.2. The sub-research questions are presented below and subsequently answered.

Manufacturing

1. What is the best structure to manufacture 3D chips?
2. Which inter-layer interconnect is the best to use with the best structure?
3. What are the properties of the best inter-layer interconnect?

Architecture

4. What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC?

Practical work

5. What is the impact on the performance and fault coverage if two stacked processors share their functional units?

5.2.1 Manufacturing

Research question one: What is the best structure to manufacture 3D chips?
The 3D stacked structures is the best, since it can stack unlimited layers, no heat budget restrictions, and there are six different inter-layer interconnects. Conversely, the 3D monolithic structure has three main problems, which are maximal three layers, unavoidable heat steps (thermal budget), and the inter-layer communications are difficult to manufacture.

Research question two: Which inter-layer interconnect is the best to use with the best structure?
A TSV is the best interconnect to use between multiple tiers, since it can stack an unlimited amount of tiers and it is the best in terms of pitch, (dynamic) power, and speed. Furthermore, it is expect that the maturity of the TSV improves, since many companies and institutions are doing research on it.

Research question three: What are the properties of the best inter-layer interconnect?

- The latency of a TSV is similar to a global 2D on-chip wire, but TSVs are much shorter. Thus, the latency of a TSV can be neglected.
- The area of a TSV occupies depends on the pitch, the (current) smallest known pitch is 7μm and the area is similar to 17 CMOS gates produced in 45nm technology.
- Power reduction is possible when long 2D on-chip wires are replaced by short TSVs. Articles are known that reduced the power consumption by 5%-20%.
- The current manufacturing and bonding yield of TSVs are low (68%), but the yield can be improved with redundant TSVs (98%) or with the Time Division Multiplexing scheme[1].
- The current characteristic life is low (1216 cycles), but it can be improved with redundant TSVs[1] or with the Time Division Multiplexing scheme[1].

5.2.2 Architectural potential and impact

In this section research question four is answered. Each topic (memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC) of the research question is answered in a separate paragraph.

Research question four: What is the architectural potential and impact of 3D integration for memory-on-memory, logic-on-logic, memory-on-logic, and 3D NOC?

5.2.2.1 Memory-on-memory

The potential for memory-on-memory stacking is that banks can be stacked and internally split, which is called bank stacking and array stacking, respectively. The impact is that bank and

[1] No exact numbers are known.

array stacking *reduces the latency variation of a (NUCA) cache* because the distance between the closest and farthest bank (inter-bank wire) is reduced. Furthermore, array stacking provides the main advantage of memory-on-memory, since it also *diminishes the memory wall problem* by reducing the intra-bank wire lengths, which speeds up the access latency of each memory bank. Bank and array stacking reduce the overall footprint, memory access time, and power consumption. The latency and energy reductions [3]for bank stacking is 9.7% and 31.5%, and for array stacking is this 21.6% and 30.4%, respectively. No thermal information is known about 3D memory stacking. However, the author of this thesis expects that the memory-on-memory circuits can handle the increase in temperatures, since the hotter logic-on-logic circuits can handle these temperatures as well.

5.2.2.2 Logic-on-logic

The potential for logic-on-logic stacking is that the wire length reduction can be used to *eliminate pipeline stages and / or increase the operating frequency* of a design, which has an positive impact on the performance and power reduction. For a 2-layers 3D Intel Pentium 4 processor this resulted in a pipeline stage reduction and performance improvements of 25% and 15%, respectively. However, the log shifters showed that it is not always beneficial to split a circuit over endless layers because eventually the circuits become gate dominated instead of wire dominated. Only two layers are beneficial for the log shifters, providing delay and power improvements up to 28% and 8%, respectively. No numbers are known about the footprint reduction, but the author of the thesis expects that the footprint also reduces for the stacked devices.

The temperature impact for logic-on-logic stacking is important to consider. There are three differences between the 2D and 3D thermal situation: (1) the heat sink area reduces, (2) the distance between the heat sink and the lower layer is larger, and (3) the power density increases. Possible solutions to solve the cooling challenge are: the use of water cooling, dummy TSVs, and scaling the supply voltage and the clock frequency. TSVs conduct heat well, since for a stacked Alpha 21364 processor of four logic-on-logic layers the temperature difference between the closest and the farthest layer from the heat sink is only <2.5°C. By scaling the supply voltage and the clock frequency it is even possible to achieve a temperature neutral 3D processor, while achieving power and performance improvements of 34% and 8%, respectively. However, it is shown for a stacked Alpha 21364 processor of four logic-on-logic layers, connected with TSVs, the maximum temperature increased with 32.2°C to 68.9°C [58], compared to the 2D processor. The author of this thesis thinks that the temperature increase is significant, but it is not four times hotter for a 4-layer stack because the wire reduction diminishes the heat generation.

5.2.2.3 Memory-on-logic

There are four potentials for memory-on-logic stacking: *(1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies.* The footprint reduction depends on the size of the memory and logic, for core stacking it provides 50% footprint reduction, given that the cache and logic are both 50% of

[3]No footprint reduction numbers are known.

the total die size. Power consumption reduces when more memory ports and larger 2D or 3D caches are used, since they reduce the communication distance and cache misses, respectively. Furthermore, the use of heterogeneous technology allows energy efficient planes to be stacked, such as DRAM memory. Performance improvements can be achieved by stacking 3D memory and 3D logic on top of each other. Furthermore, by the use of larger on-chip caches the cache hit ratio improves, and thus also the overall performance. Moreover, the overall performance also improves by using multiple memory ports at various locations, which communicate simultaneously.

SRAM has a higher power density than DRAM, and is thus hotter. However, even when a SRAM cache is stacked on a processor the temperature increase is less than 5°C (compared to the original 2D situation), and it reaches a maximal temperature of 92.9°C [11]. However, the increase is less than 5°C because a heat sink with convectioned air cools the chip. Nevertheless, the thermal impact of the stacked memory on to logic is not significant, while there are significant performance improvements.

5.2.2.4 3D Network-On-Chip

The potential of 3D Network-On-Chip (NOC) is that it enables the creation of a true physical 3D NOC topology, where the router and the network are themselves three-dimensional entities. The advantage between the vertical and the horizontal interconnects is that in the vertical direction, a TSV is just a few tens of μm long as compared to a few thousand μm for the horizontal direction [13]. Hence, *the latency of a TSV can be neglected*, compared to a global 2D wire. Three aspects are important. Firstly the total number of 3D routers in a network are important to reduce the energy and area. [12] shows that *a full 3D connected network is not (always) necessary*, and obtained energy and area reductions of 5% and 8%, respectively. However, it induced a minor latency increase of an average of 8%. Secondly, the use of an inter-layer communication bus is important because it can bridge multiple layers in a single-hop. Thirdly, crossbars scale very inefficiently, and with the uses of guided flit queuing the area, power and latency of a crossbar reduces with 48%, 36% and 20% (compared to a 6x6 crossbar), respectively. Another potential of 3D integration is the uses of heterogeneous technology. It enables, for example, the separation of photonic and electrical planes, and it is beneficial to limit (sharp) angles in the waveguide and the use of Photonic Switching Element (PSE).

5.2.3 Practical work

Research question five: What is the impact on the performance and fault coverage if two stacked processors share their functional units?

In this work two identical stacked 2D processors are evaluated, where *the FU boundary between the stacked processors is removed*. Thus, one CPU can execute instructions on all the free FUs. The free FUs on the other CPU can be used for redundancy or for performance improvement.

The impact on *the performance* is on average 7%, 6%, 4% and 2% for the configurations with one to four IALUs, respectively. The speedup is only 2% for the configuration with four IALUs,

due to the small numbers of candidate instructions (low ILP). Given the high (77%) average hit ratio, higher speedups for this configuration are expected when workloads with more ILP are available on CPU 2, resulting in more candidate instructions. The other IALU configurations have longer execution times, compared to the configuration with four IALU. However, due to the higher candidate instruction ratios they achieve higher speedups, compared to the configuration with four IALU. This performance scheme is beneficial, since no extra dedicated FUs are needed and still speedups are achieved because the boundary between the FUs of the CPUs is removed.

For the *fault coverage*, the average number of instructions executed redundantly are 15%, 23%, 35% and 52% for the configurations with one to four IALUs at CPU 2, respectively. These results show the advantage of this proposal, especially for the configuration of four IALUs. This redundant scheme is beneficial, since no extra dedicated FUs are needed and still fault coverage are achieved at low cost (only additional control logic and TSVs). This is because the boundary between the FUs of the CPUs is removed.

5.3 Overall conclusion

In this section the answer is given on the main research question.

> Main research question:
>
> *Compared to a 2D chip, what is the architectural potential and impact of a 3D stacked chip?*

The potential of 3D stacking include five key advantages: (1) wider and denser (on-chip) interconnects / busses, (2) wire length reduction (latency reduction), (3) lower power consumption, (4) the potential to use heterogeneous technologies [8], and (5) footprint reduction.

Memory-on-memory, reduces the *access latency, wire length, the power consumption and the footprint of a 3D memory*, compared to a 2D memory with an equal amount of bit storage. Furthermore, it *reduces the latency variation of a NUCA cache* and it *diminishes the memory wall problem*. With logic-on-logic, *the pipeline stages can be reduced or the frequency can be increased*, and that provides speedup. With memory-on-logic, there are four potentials: *(1) footprint reduction, (2) more and wider memory ports, (3) the use of larger 2D or 3D caches, and (4) the use of heterogeneous technologies.*

For the *temperature impact*, there are three differences between the 2D and 3D thermal situation. The first difference is that the *heat sink area reduces* equally to the footprint reduction. Secondly, the *distance between the heat sink and the lower layer is larger*, compared to a planar chip. Thirdly, the *power density increases*, and thus is the maximum temperature higher, compared to the same circuit on a planar chip. The temperature increases significantly (32.2°C [58]) at a logic-on-logic stack, but it is not four times hotter, at a 4-layer stack. The author of this thesis beliefs that it is because the *wire reduction diminishes the heat generation*. A temperature neutral 3D stack can be achieved by *scaling the supply voltage and the clock frequency.* Conclusively, the temperature increases is significant and challenging, but it is not an insuperable problem, since a temperature neutral processors can be achieved and cooling structures can cool the chip, such as a heat sink with convectioned air and water cooling.

The potential of 3D Network-On-Chip (NOC) is that it enables the creation of a true physical 3D NOC topology, where the router and the network are themselves three-dimensional entities. The advantage between the vertical and the horizontal interconnects is that in the vertical direction a TSV is just a few tens of μm long as compared to a few thousand μm for global on-chip wires in the horizontal direction [13]. There are three aspects important with a 3D NOC: (1) the total number of 3D routers in a network, (2) the inter-layer interconnect topology (i.e. bus or point-to-point), and (3) a full crossbar scales inefficiently. Furthermore, 3D integration enables the uses of heterogeneous technology, which allows the separation of photonic and electrical planes.

We have investigated two identical stacked 2D processors, where the *FU boundary between the processors is removed*. The impact on *the performance* is on average 7%, 6%, 4% and 2% for the configurations with one to four IALUs, respectively. Given the high (77%) average hit ratio, higher speedups for the performance mode is expected when workloads with more ILP run on CPU 2, resulting in more candidate instructions. For the *fault coverage*, the average number of instructions executed redundantly are 15%, 23%, 35% and 52% for the configurations one to four IALUs at CPU 2, respectively. The high redundancy numbers show the advantage of this scheme. The redundant and performance schemes are beneficial because no extra dedicated FUs are needed and still fault detection and higher performance are achieved at low cost (only additional control logic and TSVs), respectively.

5.4 Recommendations

This section discusses interesting options for future work.

5.4.1 Manufacturing

Sequential elements that are synchronized by the same clock signal can be located on multiple planes, and each plane has a slightly different clock signal. Therefore, it is interesting to perform a literature study to 3D clock distribution, to *understand the impact on the architecture* of various clock distribution networks, such as H-trees, rings, and meshes. [4]

5.4.2 Architectural

A 2D memory design has a very regular structure that makes it easy and beneficial to partition across multiple dies. The besides the known array stacking strategies, (new) efficient 3D *memory-on-memory* structure should be investigated. The regular structure allows high speedups, which are achieved with low redesign efforts, since it is replicated multiple times.

Logic-on-logic stacking via logic gate splitting is *especially effective for wire dominated cores / FUBs*, such as a SRAM memory [9] or a Kogge-stone / Sklansky adders [10,51]. Hence, it is recommended to investigate which FUBs of a processor are wire dominated, and thus *identify the candidate blocks* for 3D stacking via logic gate splitting.

It is interesting to explore the possibilities for stacking a (DRAM) memory plane with *multiple and wider memory ports* on top of a Chip Multi-Processor (CMP) (*memory-on-logic*). It is interesting because multiple ports provides parallel access, and that increases the total memory

[4] Clock distribution is out of the scope of this thesis.

bandwidth [64]. Furthermore, it is intersecting to know what the impact is if each banks get its own data and command bus, which allows parallel memory access to different banks of the memory. Moreover, a memory plane on top of a processor plane via core stacking is the easiest form of 3D integration and thus extra interesting for real production purposes with low redesign efforts.

A 3D NOC enables the creation of a true physical 3D NOC topology, where the router and the network are themselves three-dimensional entities. Given this extra third 'Z' dimension it is recommended to *study the impact of the extra dimension on a 3D Network-On-Chip (NOC)*, in terms of area, power, speed, protocols, routing algorithms, deadlocks and livelocks.

5.4.3 Practical work

For future work, it is useful and interesting to be able to simulate three or more processors on top of each other, running in performance or redundant modes. Furthermore, the ability of bi-directional instruction issuing between CPUs is recommended, to determine the total impact of both schemes. Another refinement for the redundant mode strategy that can be implemented is to search for three free FUs on the same CPU or on other CPUs, where at least one instruction is issued to another CPU. With a triple modular redundancy scheme fault correction is possible. Furthermore, it is interesting to perform fault injection experiments in order to determine how many soft-errors the redundant scheme detects. These recommendations require adaptation in the 3D SS simulator.

Bibliography

[1] C. Seng Tan, R. J. Gutmann, and L. R. Reif, *Wafer Level 3-D ICs Process Technology*. Springer, 2008.

[2] P. Vasilis and E. Friedman, *Three-dimensional integrated circuit design*. Morgan kaufmann, 2008.

[3] R. Canegallo, L. Ciccarelli, F. Natali, A. Fazzi, R. Guerrieri, and P. Rolandi, "3d contactless communication for ic design," in *Proc. IEEE International Conference on Integrated Circuit Design and Technology and Tutorial ICICDT 2008*, Jun. 2–4, 2008, pp. 241–244.

[4] J. Xu, J. Wilson, S. Mick, L. Luo, and P. Franzon, "2.8 gb/s inductively coupled interconnect for 3d ics," in *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, 2005, pp. 352–355.

[5] "Date 2009 presentations," 2009. [Online]. Available: http://www.date-conference.com/files/file/09-workshops/date09-3dws-digestv2-090504.pdf

[6] P. C. Viet H. Nguyen, "The impact of interstratal interconnect density on the performance of three-dimensional integrated circuits," in *SLIP 05: Proceedings of the 2005 international workshop on System level interconnect prediction*, 2005. [Online]. Available: https://nww.p2r.nxp.com/d/d/workspace/SpacesStore/f8c86feb-2063-11de-9029-af9af890b9b4/PR-MS%2024.595-1.pdf/

[7] M. Bschorr, H. Pfleiderer, P. Benkart, A. Kaiser, A. Munding, E. Kohn, A. Heittmann, H. Hubner, and U. Ramacher, "Yield-improving test and routing circuits for a novel 3d interconnect technology," *Advances in Radio Science*, vol. 4, pp. 225–229, 2006.

[8] G. Loh, Y. Xie, and B. Black, "Processor design in 3d die-stacking technologies," *IEEE Computer Society*, no. 0272-1732/05, 2007.

[9] K. Puttaswamy and G. H. Loh, "Implementing caches in a 3d technology for high performance processors," in *Proc. IEEE International Conference on Computer Design: VLSI in Computers and Processors ICCD 2005*, Oct. 2–5, 2005, pp. 525–532.

[10] B. Vaidyanathan, W.-L. Hung, F. Wang, Y. Xie, V. Narayanan, and M. J. Irwin, "Architecting microprocessor components in 3d design space," in *Proc. the international conference on VLSI design held jointly with 6th international conference on embedded systems*, Jan. 6–10, 2007, pp. 103–108.

[11] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, "Die stacking (3d) microarchitecture," in *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, Dec. 2006, pp. 469–479.

[12] A. Bartzas, N. Skalis, K. Siozios, and D. Soudris, "Exploration of alternative topologies for application-specific 3d networks-on-chip," in *Proceedings of the Workshop on Application-Specific Processors*, 2007.

[13] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. Yousif, and C. Das, "A novel dimensionally-decomposed router for on-chip communication in 3d architectures," *ISCA07, June 9*, 2007.

[14] D. Burger and T. Austin, "The simplescalar tool set, version 2.0," *ACM SIGARCH Computer Architecture News*, vol. 25, no. 3, pp. 13–25, 1997.

[15] P. Garrou, *Handbook of 3D Integration: Technology and Applications of 3D Integrated Circuits*, P. Garrou, C. Bower, and P. Ramm, Eds. Wiley-VCH, 2008, vol. 1.

[16] R. Anigundi, H. Sun, J.-Q. Lu, K. Rose, and T. Zhang, "Architecture design exploration of three-dimensional (3d) integrated dram," in *Proc. Quality of Electronic Design Quality of Electronic Design ISQED 2009*, Mar. 16–18, 2009, pp. 86–90.

[17] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, "Design space exploration for 3d architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, pp. 65–103, 2006.

[18] S. Bahirat and S. Pasricha, "Exploring hybrid photonic networks-on-chip for emerging chip multiprocessors," 2009.

[19] A. Shacham, K. Bergman, and L. P. Carloni, "The case for low-power photonic networks on chip," in *Proc. 44th ACM/IEEE Design Automation Conference DAC '07*, Jun. 4–8, 2007, pp. 132–135.

[20] V. Nguyen and P. Christie, "The impact of interstratal interconnect density on the performance of three-dimensional integrated circuits," april 2005. [Online]. Available: http://www.sliponline.org/SLIP05/Presentations/6.2_Viet.pdf

[21] G. E. Moore, "Progress in digital integrated electronics," in *Proc. International Electron Devices Meeting*, vol. 21, 1975, pp. 11–13.

[22] A. K. Deb, T. H. Hendriksson, and P. van der Wolf, "Memory integration technologies: overview, comparison and opportunities," NXP Semiconductors, Tech. Rep. technical note NXP-R-TN 2007/00145, 2007.

[23] K. Puttaswamy, "Designing high-performance microprocessors in 3-dimensional integration technology," Ph.D. dissertation, School of Electrical and Computer Engineering Georgia Institute of Technology, 2007.

[24] M. Awasthi and R. Balasubramonian, "Exploring the design space for 3d clustered architectures," in *3rd IBM Watson Conference on Interaction between Architecture, Circuits, and Compilers, Yorktown Heights*, 2006.

[25] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3d ics: the pros and cons of going vertical," *IEEE Design Test of Computers*, vol. 22, no. 6, pp. 498–510, Nov. 2005, advantages disadvantage 3d integration, capacitive, inductive,.

[26] J. Xu, S. Mick, J. Wilson, L. Luo, K. Chandrasekar, E. Erickson, and P. Franzon, "Ac coupled interconnect for dense 3-d ics," *IEEE Transactions on Nuclear Science*, vol. 51, no. 5 Part 1, pp. 2156–2160, 2004.

[27] "C4np shrinks and grows- tutorial91," flipchip.com. [Online]. Available: http://www.flipchips.com/tutorial91.html

[28] "3d-wlp - microbumping," IMEC. [Online]. Available: http://www.imec.be/ScientificReport/SR2008/HTML/1224990.html

[29] D. G. Riley, "Injection molding solder bumps- tutorial 56," Flipchip.com, October 2005. [Online]. Available: http://www.flipchips.com/tutorial56.html

[30] G. Carchon, "3d microwave module packaging," IMEC, 2007. [Online]. Available: https://escies.org/GetFile?rsrcid=19653

[31] K. De Munck, P. S. De Moor, D. Tezcan, K. Baert, E. Beyne, and M. R. V. H. C., "3d interconnect technology for space applications." [Online]. Available: https://escies.org/GetFile?rsrcid=1706

[32] D. Jang, C. Ryu, K. Lee, B. Cho, J. Kim, T. Oh, W. Lee, J. Yu, and D. KAIST, "Development and evaluation of 3-d sip with vertically interconnected through silicon vias (tsv)," in *Proc. 57th Electronic Components and Technology Conference ECTC '07*, May 2007, pp. 847–852.

[33] S. L. Hsien-Hsin, "Isca-35 tutorial 3d-ic microarchitecture," Georgia Institute of Technology, p. 20, 2008. [Online]. Available: http://www.cc.gatech.edu/~loh/3D-tutorial/ISCA-2008/ISCA-2008-tutorial-3D-Arch.pdf

[34] "3d-sic stacked ics - july 2009," Imec, july 2009. [Online]. Available: http://www.imec.be/ScientificReport/SR2007/html/1384072.html

[35] P. van der Wolf, Ed., *MPSoc'09 - The memory bottleneck in MPSoc for multimedia*, 2009. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1008339

[36] "3d interconnects through silicon vias (tsvs)," Sematech, 2009. [Online]. Available: http://sematech.org/research/3D/index.htm

[37] "Interconnect," international technology roadmap for semiconductors (ITRS), Tech. Rep. 2007 EDITION, 2007. [Online]. Available: http://www.itrs.net/Links/2007ITRS/Home2007.htm

[38] A. Fazzi, L. Magagni, M. Mirandola, R. Canegallo, S. Schmitz, and R. Guerrieri, "A 0.14 mw/gbps high-density capacitive interface for 3d system integration," in *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, 2005, pp. 101–104.

[39] E. Culurciello and A. G. Andreou, "Capacitive inter-chip data and power transfer for 3-d vlsi," vol. 53, no. 12, pp. 1348–1352, Dec. 2006.

[40] S. Kuhn, M. Kleiner, R. Thewes, and W. Weber, "Vertical signal transmission in three-dimensional integratedcircuits by capacitive coupling," in *1995 IEEE International Symposium on Circuits and Systems, 1995. ISCAS'95.*, vol. 1.

[41] "Elpida to ship first 3d tsv stacked dram memory in 2009," September 2009. [Online]. Available: http://www.i-micronews.com/lectureArticle.asp?id=3448

[42] "Gold bonding wire & ribbon." [Online]. Available: http://www.coininginc.com/gold_wire_and_ribbon.asp

[43] P. Siblerud, "Emc-3d extends the consortium life two additional years and expands goals for 300mm thru-silicon-via interconnect," p. 2, July 2009. [Online]. Available: http://emc3d.org/documents/pressReleases/2009/EMC3D_consortium_continuation_July2009.pdf

[44] M. Karnezos, "3d packaging: where all technologies come together," in *Proc. IEEE/CPMT/SEMI 29th International Electronics Manufacturing Technology Symposium*, Jul. 2004, pp. 64–67.

[45] "3d integration- opertunities, challengens and insustry readiness- perspective from design, manufacturing and eda," 2009.

[46] "2008 update - overview," international technology roadmap for semiconductors - (itrs), Tech. Rep. overview, 2008. [Online]. Available: http://www.itrs.net/Links/2008ITRS/Home2008.htm

[47] G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee, "A thermally-aware performance analysis of vertically integrated (3-d) processor-memory hierarchy," in *DAC '06: Proceedings of the 43rd annual Design Automation Conference*. New York, NY, USA: ACM, 2006, pp. 991–996.

[48] I. Loi, F. Angiolini, and L. Benini, "Supporting vertical links for 3d networks-on-chip: toward an automated design and analysis flow," in *Proceedings of the 2nd international conference on Nano-Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, p. 15.

[49] F. P. Abbas Sheibanyrad, "Tima - sls team contributions on wp3 - t3.2," Powerpoint - Email, 2009.

[50] P. Reed, G. Yeung, and B. Black, "Design aspects of a microprocessor data cache using 3d die interconnect technology," in *Proc. International Conference on Integrated Circuit Design and Technology ICICDT 2005*, May 9–11, 2005, pp. 15–18.

[51] K. Puttaswamy and G. Loh, "The impact of 3-dimensional integration on the design of arithmetic units," in *Proceedings of the International Symposium on Circuits and Systems*. Citeseer, 2006, pp. 4951–4954. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.1579&rep=rep1&type=pdf

[52] F. Gebali, H. Elmiligi, and M. El-Kharashi, Eds., *Networks-On-Chips: Theory and Practice*. CRC Press, Taylor and Francis group, 2009.

[53] I. Loi, S. Mitra, T. Lee, S. Fujita, and L. Benini, "A low-overhead fault tolerance scheme for tsv-based 3d network on chip links," in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2008, pp. 598–602.

[54] N. Miyakawa, "A 3d prototyping chip based on a wafer-level stacking technology," in *ASP-DAC '09: Proceedings of the 2009 Asia and South Pacific Design Automation Conference*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 416–420.

[55] N. Miyakawa, E. Hashimoto, T. Maebashi, N. Nakamura, Y. Sacho, S. Nakayama, and S. Toyoda, "Multilayer stacking technology using wafer-to-wafer stacked method," *J. Emerg. Technol. Comput. Syst.*, vol. 4, no. 4, pp. 1–15, 2008.

[56] S. J. Koester, A. M. Young, R. R. Yu, S. Purushothaman, K.-N. Chen, D. C. La Tulipe, N. Rana, L. Shi, M. R. Wordeman, and E. J. Sprogis, "Wafer-level 3d integration technology," *IBM J. Res. Dev.*, vol. 52, no. 6, pp. 583–597, 2008. [Online]. Available: http://www.signallake.com/innovation/koester.pdf

[57] T. Kuo, S. Chang, Y. Shih, C. Chiang, C. Hsu, C. Lee, Y. Chun-Te Lin, and W. Lo, "Reliability tests for a three dimensional chip stacking structure with through silicon via connections and low cost," in *Electronic Components and Technology Conference, 2008. ECTC 2008. 58th*, 2008, pp. 853–858.

[58] K. Puttaswamy and G. Loh, "Thermal analysis of a 3d die-stacked high-performance microprocessor," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*. ACM, 2006, p. 24.

[59] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3d chip multiprocessors using network-in-memory," in *Proc. 33rd International Symposium on Computer Architecture ISCA '06*, 2006, pp. 130–141.

[60] "Samsung develops world's first eight-die multi-chip package technology for multimedia cell phones," Januari 2005. [Online]. Available: http://www.physorg.com/news2631.html

[61] C. Kim, D. Burger, and S. Keckler, "An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches," *ACM SIGPLAN Notices*, vol. 37, no. 10, p. 222, 2002.

[62] G. H. Loh, "3d-stacked memory architectures for multi-core processors," in *Proc. 35th International Symposium on Computer Architecture ISCA '08*, Jun. 21–25, 2008, pp. 453–464.

[63] "Ibm work on 3d chip stacking will take moore's law to 2025," March 2010. [Online]. Available: http://www.i-micronews.com/lectureArticle.asp?id=4413

[64] E. Aho, J. Nikara, P. A. Tuominen, and K. Kuusilinna, "A case for multi-channel memories in video recording," in *Proc. DATE '09. Design, Automation. Test in Europe Conference. Exhibition*, Apr. 20–24, 2009, pp. 934–939. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5090799&isnumber=5090609

[65] M. Gomaa and T. Vijaykumar, "Opportunistic transient-fault detection," in *Computer Architecture, 2005. ISCA'05. Proceedings. 32nd International Symposium on*, 2005, pp. 172–183.

[66] "Simplescalar 3.0 (revision 3.0d) simulator." [Online]. Available: http://www.simplescalar.com/

[67] T. Austin, "A user's and hacker's guide to the simplescalar architectural research tool set," *Intel MicroComputer Research Labs*, 1997. [Online]. Available: http://www.cems.uwe.ac.uk/~rwilliam/ACA_ufeEHK-20-3/ACA_course/simplescalar_slides.pdf

[68] "Mediabench consortium." [Online]. Available: http://euler.slu.edu/~fritts/mediabench/

[69] "European pro3d consortium to focus on programming 3d manycore architectures," may 2010. [Online]. Available: http://www.i-micronews.com/lectureArticle.asp?id=4683

[70] T. Austin, "Diva: A reliable substrate for deep submicron microarchitecture design," in *Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture*. IEEE Computer Society, 1999, pp. 196–207.

[71] N. Madan and R. Balasubramonian, "Leveraging 3d technology for improved reliability," in *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2007, pp. 223–235.

[72] "Reactive-ion etching," 2008. [Online]. Available: http://en.wikipedia.org/wiki/Reactive-ion_etching

[73] W. Maly, "Semiconductor fabrication steps- lecture 4a," TU E.

[74] "2008 focus itwg tables: Emerging research devices (erd), emerging research materials (erm), front-end processes (fep), lithography, interconnect, factory integration, and assembly & packaging," 2008. [Online]. Available: http://www.itrs.net/Links/2008ITRS/Update/2008Tables_FOCUS_B.xls

[75] "Imis - intimate memory interface specification standard," 3D-IC Allicance, Tech. Rep., Jan. 2008. [Online]. Available: http://www.3d-ic.org/documents/IMIS%201.0.pdf

[76] P. Garrou, *Handbook of 3D Integration: Technology and Applications of 3D Integrated Circuits*, P. Garrou, C. Bower, and P. Ramm, Eds. Wiley-VCH, 2008, vol. 2.

[77] P. Siblerud, "Emc-3d consortium develops process and cost model for interconnect thru-silicon-via or (itsvtm) structures," p. 3, 2008. [Online]. Available: http://emc3d.org/documents/pressReleases/2008/EMC3D_iTSV_CoO_PressRelease_final_Sept4_2008.pdf

[78] "Press release - a*star and edb launch 3-dimensional through-silicon via consortium to boost next generation wafer manufacturing capability for singapore semiconductor industry," 9 2009. [Online]. Available: http://www.ime.a-star.edu.sg/img/pdf/News%20Releases/IME-ASTAR-EDB%203D%20TSV%20consortium%20press%20release_20090907.pdf

[79] "Specific heat capacity," Februari 2010. [Online]. Available: http://en.wikipedia.org/wiki/Specific_heat_capacity

[80] "Thermal resistance in electronics," February 2010. [Online]. Available: http://en.wikipedia.org/wiki/Thermal_resistance_in_electronics

[81] "Thermal conductivity," Februari 2010. [Online]. Available: http://en.wikipedia.org/wiki/Thermal_conductivity

[82] I. O'Connor, M. Briere, E. Drouard, A. Kazmierczak, F. Tissafi-Drissi, D. Navarro, F. Mieyeville, J. Dambre, D. Stroobandt, J. Fedeli *et al.*, "Towards reconfigurable optical networks on chip," in *Reconfigurable Communication-centric Systemson-Chip workshop*, 2005.

[83] "Silicon photonics for next generation computing systems," 2008. [Online]. Available: http://www.research.ibm.com/photonics/publications/ecoc_tutorial_2008.pdf

[84] "Wavelength division multiplexed photonic layer on cmos," 2008. [Online]. Available: http://wadimos.intec.ugent.be/uploads/media/ProjectPresentatioDocument_WADIMOS.pdf

[85] J. Goodman, F. Leonberger, S. Kung, and R. Athale, "Optical interconnections for vlsi systems," *Proceedings of the IEEE*, vol. 72, no. 7, pp. 850–866, 1984.

[86] "Wavelength-division multiplexing," March 2010. [Online]. Available: http://en.wikipedia.org/wiki/Wavelength-division_multiplexing

[87] S. Adya and I. Markov, "Fixed-outline floorplanning: Enabling hierarchical design," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, vol. 11, no. 6, pp. 1120–1135, 2003.

[88] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: illuminating future network-on-chip with nanophotonics," 2009.

[89] K. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashimoto, K. Park, H. Kurino, and M. Koyanagi, "Three-dimensional shared memory fabricated using wafer stackingtechnology," in *Electron Devices Meeting, 2000. IEDM Technical Digest. International*, 2000, pp. 165–168. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&isnumber=&arnumber=904284

[90] B. Black, D. Nelson, C. Webb, and N. Samra, "3d processing technology and its impact on ia32 microprocessors," in *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings*, 2004, pp. 316–318. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&isnumber=&arnumber=1347939

[91] Y. Kurita, S. Matsui, N. Takahashi, K. Soejima, M. Komuro, M. Itou, C. Kakegawa, M. Kawano, Y. Egawa, Y. Saeki, H. Kikuchi, O. Kato, A. Yanagisawa, T. Mitsuhashi, M. Ishino, K. Shibata, S. Uchiyama, J. Yamada, and H. Ikeda, "A 3d stacked memory integrated on a logic device using smafti technology," in *Proc. 57th Electronic Components and Technology Conference ECTC '07*, May 2007, pp. 821–829.

[92] K. Puttaswamy and G. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors," *Int. Sym. on High-Perf. Comp. Arch*, pp. 193–204, 2007.

[93] 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 *Proc. 35th International Symposium on Computer Architecture ISCA '08*, Jun. 21–25, 2008, pp. 251–261.

List of abbreviations

A

3D SS 3D SimpleScalar

ALU Arithmetic Logic Unit

BEOL Back End Of Line

BGA Ball Grid Array

CVD Chemical Vapor Deposition

CMP Chip Multi-Processor

CTE Coefficient of Thermal Expansion

CoO Cost of Ownership

DRIE Deep Reactive Ion Etching

DimDe Dimensionally-Decomposed

FP Floating Point

FEOL Front End Of Line

FUB Functional Unit Block

FU Functional Unit

IMS Injection Molded Solder

IPC Instruction Per Cycle

ILP Instruction-Level Parallelism

IALU Integer ALU

ITRS International Technology Roadmap for Semiconductors

KS Kogge Stone

LPCVD Low-Pressure Chemical Vapor Deposition

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor

MLDA Multi-Layer Data Access

NOC Network-On-Chip

NUCA Non-Uniform Cache Access

OTDM Optical Time Division Multiplexing

POP Package-On-Package

PSE Photonic Switching Element

PCT Pressure Cooker Test

PE Processor Element

RIE Reactive Ion Etching

RF Register File

ROB Reorder Buffer

SIMOX Separation by IMplantation of OXygen

SOI Silicon-On-Insulator

SOS Silicon-On-Sapphire

SS SimpleScalar

SIMD Single Instruction Multiple Data

SLDA Single-Layer Data Access

SLID Solid-Liquid InterDiffusion

SHC Specific Heat Capacity

SIP System-In-Package

TCT Temperature Cycling Test

TR Thermal Resistivity

3D Three-Dimensional

TSV Through Silicon Via

TDM Time Division Multiplexing

2D Two-Dimensional

WDM Wavelength-Division Multiplexing

3D Monolithic processes in use in the industry

B

Section 2.1.2 presented non-existing processes and it was indented to give a general idea for all possible processes. Conversely, this appendix presents two existing processes in use in the industry. The challenges and advantages are similar to the presented challenges, which were presented in Section 2.1.2. The first presented process is the single SOI wafer process, which follows the layer-by-layer approach (previously explained in Section 2.1.2). The second presented process is the FinCMOS process, which follows the simultaneous multi-layer approach. As stated previously, both sections assume that the lower and upper silicon layers are (already) fabricated. The lower silicon layer is fabricated with a conventional process and the upper silicon layer is fabricated with the techniques described in Section 2.1.1 (laser crystallization or seed crystallization). This section describes how the Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) are fabricated on the silicon layers. It is unknown if the single SOI process and the FinCMOS process are still in use.

B.1 Single SOI wafer

The single SOI wafer process follows the layer-by-layer approach, which means that the MOSFETs are fabricated per layer. A (single) SOI wafer consists out of two silicon layers, separated by an isolation layer (see Figure B.1). The process begins with a SOI wafer and then the shape of the gates (width and length) at both layers are fabricated. Afterwards, the drain and source (N-wells) at the bottom layer are doped. Thereafter, the drain and source of the top layer are doped (P-wells) (see Figure B.1.(c)-(f)). Eventually, the gates are fabricated and contacts are made (see Figure B.1.(g)-(i)). The gates, at the lower layers, are fabricated via a special tunneling process, but this process is very difficult to control (see Figure B.1.(f)).

The main advantage is that this method produces a compact CMOS structure, resulting in three advantages. The double gated MOSFETs align automatically to each other, because the top and bottom gates are manufactured at the same time. This means that there is no need for a extra alignment process step. Furthermore, the upper layer (PMOSFET) exists out of a thin silicon film. This results in low parasitic capacitance in the MOSFETs, and thus the switching time of the MOSFETs is faster. The stacked PMOSFET and NMOSFET have symmetrical rise and fall times and short interconnects between the upper and bottom MOSFET.

This method also has disadvantages. Even though the PMOSFETs and the NMOSFETs are fabricated together, the thermal budget problem remains. This is because the N-wells at the bottom layer (NMOSFETs) are fabricated in the early stages (see Figure B.1.(c)) and the higher temperature are needed the fabricate the upper MOSFETS, i.e. formation of the source and drain regions of the upper layer (Figure B.1.f). The high temperatures, needed for the upper layers, leads to an increase of dopant diffusion, which means that there is less dopant concentration in the bottom layer. Despite the attempt to reduce the thermal cycles, there are still unavoidable

steps remaining, which contribute to the extra dopant diffusion, such as gate oxidation. Therefore, the thermal budget remains a problem. [1] It is unknown if the single SOI process is still used.

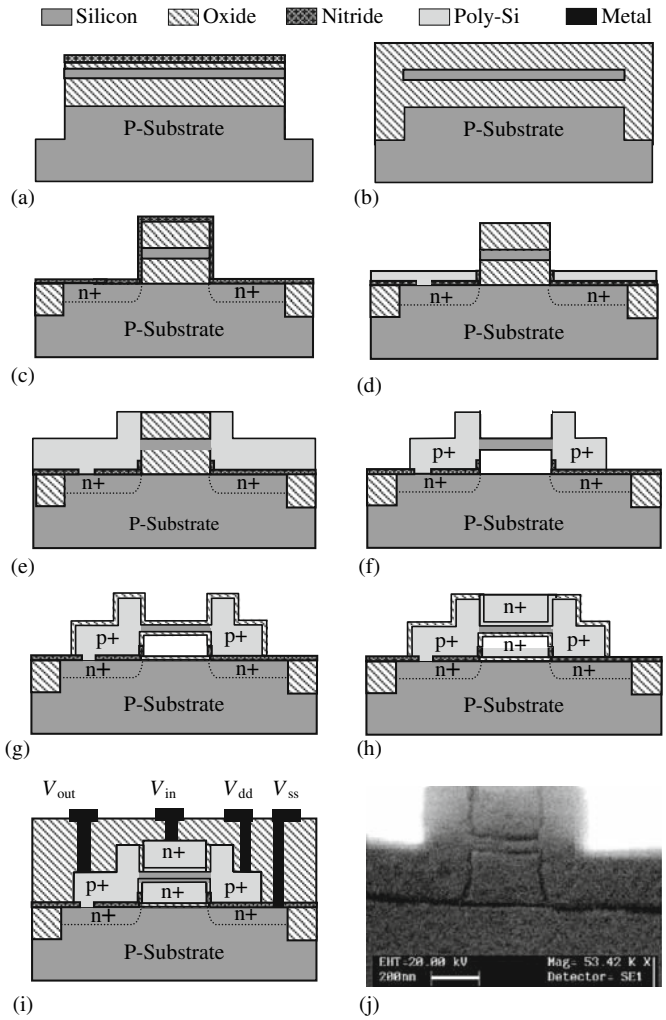


Figure B.1: (a) Start wafer build up from SOI and a thin silicon layer. (b) Nitride removed and a low temperature oxide is used to fill the trenches. (c) Gate regions are performed (d) Top of the gate is trimmed (e) N-wells are fabricated with boron doping (f) Trench beneath and on top of the silicon film is etched (g) PMOS and NMOS are grown together (h) Formation of gate electrodes (i) Deposition of a passive layer and making contact openings (j) Final structure [1]

B.2 FinCMOS Technology

The FinCMOS process follows the simultaneous multi-layer approach, which means that all MOSFETs are manufactured at all the layers at once (simultaneously). FinCMOS Technology is considered to be the most promising technology for manufacturing double gates [1]. A double

SOI substrate is used as starting material and should be processed into the right shape (see Figure B.2 a and b). The gate oxide is grown and then the gate is fabricated (see Figure B.2.(c) and (d)). The NMOSFETs and PMOSFETs have a shared gate. The bottom layer is doped with boron and the top layer with arsenic to create the P-wells and N-wells (see Figure B.2.(e)). Vias are introduced to connect the bottom layer with the upper layer (see Figure B.2.(f) and (g)). The final result is depicted in Figure B.2.h.

The simultaneous formation of top and bottom MOSFETs is the main advantage of this process. Thus there is no accumulation of thermal cycles (no dopant diffusion problems due to extra thermal heat), and thus it resolves the thermal budget problem from Section B.1. Another advantage is that the NMOSFETs and PMOSFETs are perfectly stacked, due to the simultaneous formation of the top and bottom layer. Furthermore, the ratio between the NMOSFETs and PMOSFETs current is adjustable, by using different "fin"/ silicon layer heights, but it is restricted by the height of the used material.

Despite the number of advantages, there are also some disadvantages. Threshold voltages of the NMOS and PMOS are not equal, because normally two different sized gates are used (in this case, only a single poly-silicon gate is used). Furthermore, only NMOS on top of PMOS is demonstrated. However, NMOS below and PMOS on top is not yet demonstrated, because it is more difficult to make a NMOS through the top layer.

Nevertheless, this technology was/ is used for a manufacturing devices, such as: inverters, NAND gates and static random access memory (SRAM). It is unknown of this technology is still used by foundries.

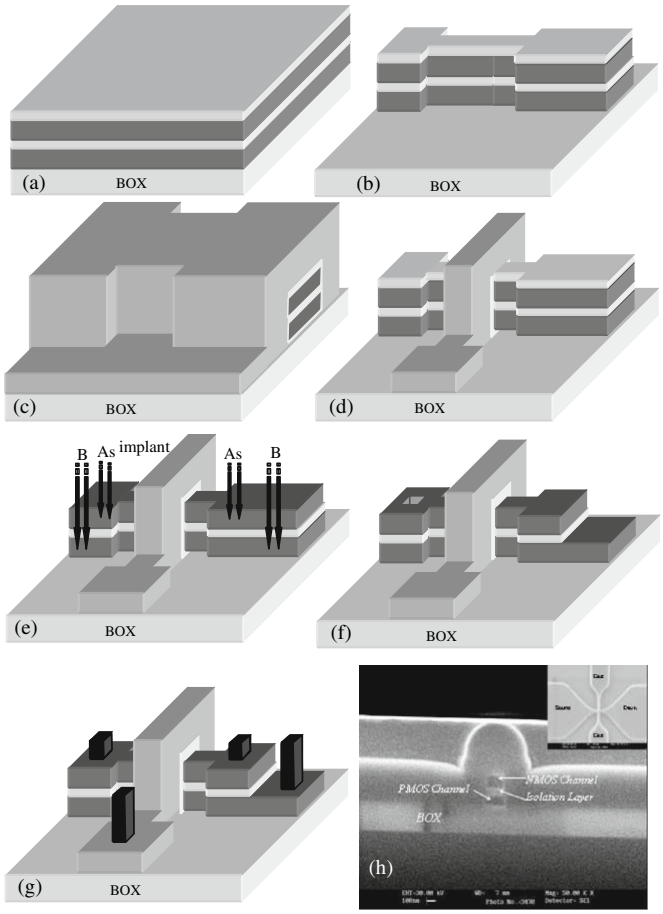


Figure B.2: Key processing steps to form the stacked FinCMOS inverter. [1]

Fabrication of 3D stack structures

C

C.1 TSV fabrication

This appendix provides background information on how the TSVs are manufactured, and the bond methods for multiple tiers are presented. This appendix concludes by presenting the process sequences that are in uses in the industry, since many process sequences can be made with the TSV manufacturing processes and the bonding methods.

In Section C.1.1, the diverse process positions and process names of TSVs are presented. Subsequently, two fabrication methods are presented for the TSV hole fabrication. The TSV holes in the silicon wafer are fabricated via a laser or with Deep Reactive Ion Etching (DRIE) [15, p.93], [1, p.86]. Laser drilling is presented in Section C.1.2 and the DRIE method is presented in Section C.1.3.

C.1.1 TSV process position and names

This section presents four different positions in a manufacturing process where a TSV can be fabricated.

A TSV can be manufactured at different stages of the 3D wafer stacking process. The TSV process names and the fabrication methods are dependent on the fabrication position in a 3D wafer stacking process. In this section, the process names and positions are presented. Fabrication of a TSV is divided into two groups, containing four different processes. The division is relative to the IC manufacturing process (see Figure C.1). The first group contains the TSV manufacturing processes, which are fabricated in the IC fabrication process. The second group contains the TSV manufacturing processes, which are fabricated after the IC fabrication process.

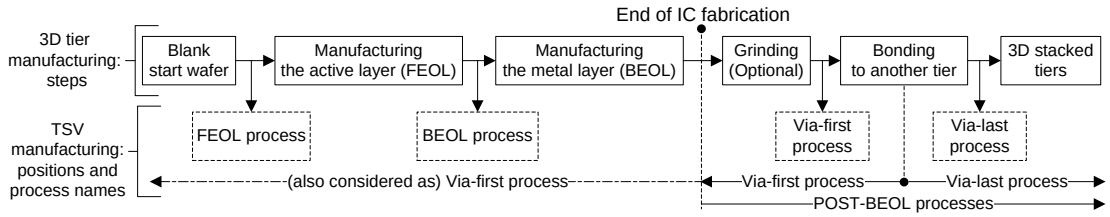
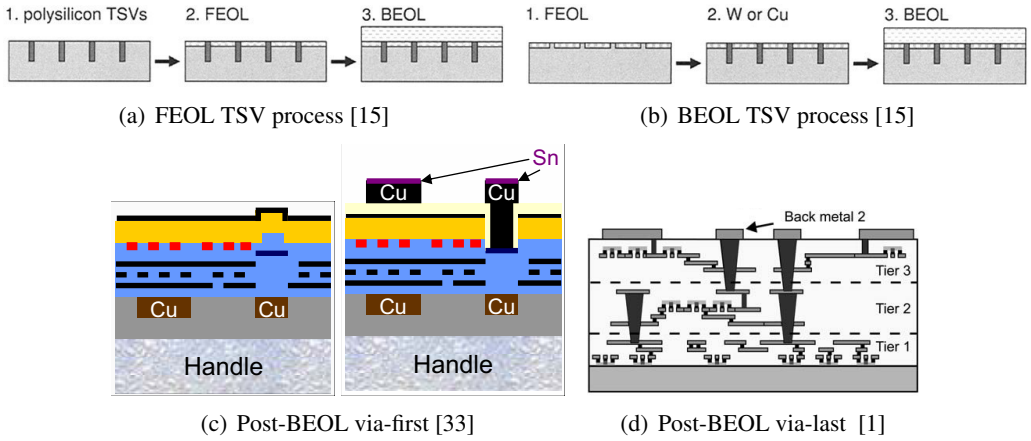


Figure C.1: The possible TSV manufacturing positions are indicated below the 3D tier manufacturing process, and the TSV process name is indicated at that particular position.

The first group contains the Front End Of Line (FEOL) and the Back End Of Line (BEOL) TSV manufacturing processes. FEOL TSVs are made of polysilicon, which has a drawback of high resistivity compared to metals. However, they can be made conductive enough for many applications. Diverse organizations are developing this technology [15, p.26], i.e. CEA Leti,



(a) FEOL TSV process [15]

(b) BEOL TSV process [15]

(c) Post-BEOL via-first [33]

(d) Post-BEOL via-last [1]

Figure C.2: TSV fabrication process options.

NEC and Zycube. The BEOL TSVs contain metals like tungsten (W) and copper (Cu). The TSVs are (usually) fabricated at the beginning of the BEOL process, so that the via does not occupy valuable area in the metal layer (see Figure 2.14(b)).

The second group is also referred to as post-BEOL processes. A post-BEOL process can be manufactured at IC foundries that do not (yet) support TSV fabrication, which is advantageous. The post-BEOL process is divided into a via-first or last process, relative to the bonding process. In a via-first process, the vias are fabricated from the rear of the tier (see Figure C.2(c)). These TSVs are connected directly (TSV nail) or via small bumps, to another tier (see Figure 2.14). In a via-last process, the tiers are (mechanically) bonded with a dielectric (non conductive) adhesive, and the vias are fabricated afterwards. The TSV is fabricated through the dielectric bond layer, onto the upper metal of the lower tier [15] (see Figure C.2(d)).

The BEOL and the FEOL processes of group one are also generalized, and referred to as via-first processes. It is because the via-first process is relative to the bonding process, and the FEOL and BEOL are fabricated before bonding. Therefore, it can be seen as a via-first process.

C.1.2 Laser drilling

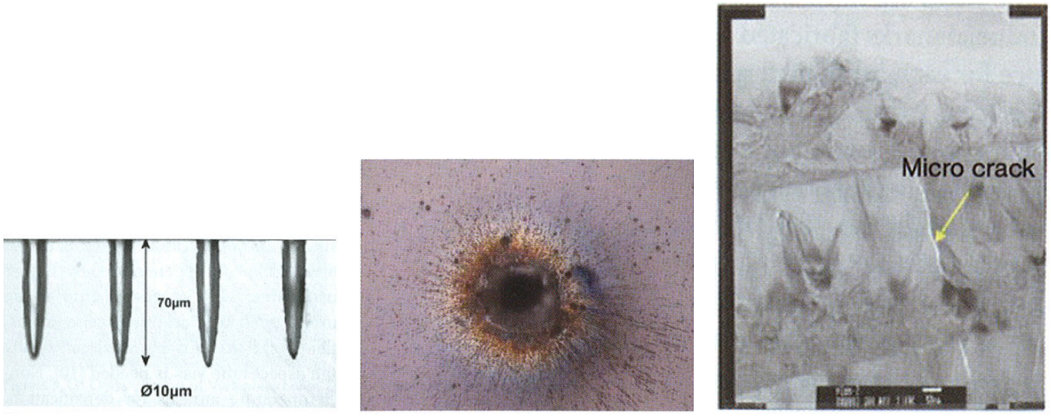
The TSV holes in the silicon wafer are fabricated via a laser or with a Deep Reactive Ion Etching (DRIE) process [15, p.93], [1, p.86]. DRIE is presented in Section C.1.3 and laser drilling in this section. The two methods have a main disadvantage, they occupy area in the active layer (the MOSFET layer) and sometimes in the metal layer. A TSV occupies area in the metal layer, when a POST-BEOL process is used, such as laser drilling and POST-BEOL via-last approach. This decreases the MOSFET density in the active layer.

The first method to make a TSV hole is with a laser. A laser punches a hole through the whole tier, or it can stop at any desired depth (see Figure C.3(a)) [15, p.35]. A laser based TSV is produced in two steps. The first beam makes a hole, which is filled with an isolation material. The isolation layer, isolate the TSV from the substrate and it decreases the parasitic capacitance. A second punch with a thinner laser is needed to make a hole in the isolation. That hole is filled with electric conductive material [15, p.98].

Laser drilling provides cost advantages over etching, if the TSVs are less than $\approx$10.000

(holes) and have a diameter of >10μm [15, p.93], [1, p.87]. Another advantage of laser ablation is that it makes a TSV with a high aspect ratio with one punch. An aspect ratio is the ratio between the depth and width of the TSV (depth : width).

Even though fabrication of the hole is a one-step process, the laser needs to make two holes per TSV, which is not advantageous. Furthermore, the size of the TSV is large, compared to a Deep Reactive Ion Etching (DRIE). This decreases the interconnect density. However, some expect that the diameter could be reduced to 1μm [15, p.35]. The laser punches with high energy through the tier. The heat that it generates in the surrounding zone could harm or decrease the performance of the device. Micro cracks appear around the TSV, due to the heat of the laser. This compromises the strength of a wafer (see Figure C.3(c)). Furthermore, the side wall of the drilled TSV is not smooth, but it has a saw-tooth texture. Another downside is the debris it produces, melted silicon is scattered around the drilled hole, which is not easily removed by a conventional cleaning processes [15, p94] (see Figure C.3(b)). However, methods have been developed to remove or reduce the debris. One such method uses a sacrificial coating, that is later removed when the debris has landed on it (see Figure C.3(b)).



(a) A laser drilled TSV can stop at any depth. TSV is 70μm deep and has a diameter of 10μm. [15]

(b) Laser debris around the hole. [15]

(c) A micro crack due to the heat of a laser [15]

Figure C.3: TSV made by a laser.

C.1.3 Deep reactive ion etching

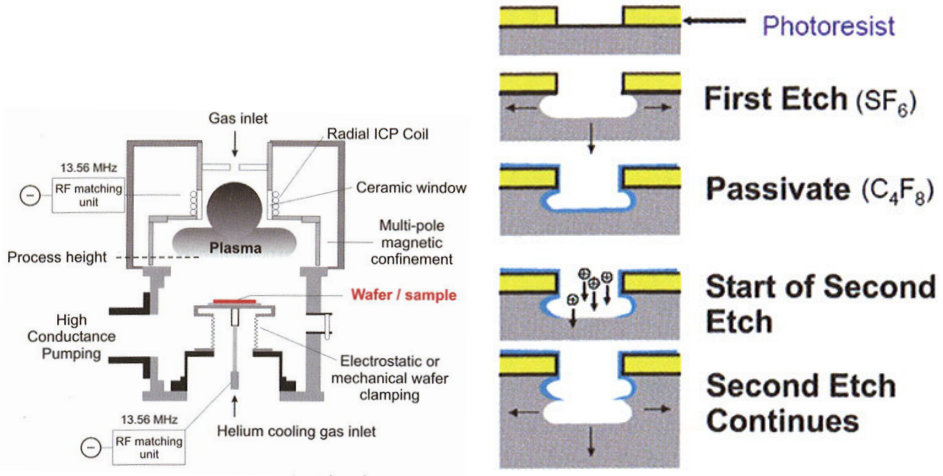
The TSV holes in the silicon wafer are fabricated via a laser or with Deep Reactive Ion Etching (DRIE) [15, p.93], [1, p.86]. As stated previously, both processes have a main disadvantage. They occupy area in the active layer (the MOSFET layer) and sometimes in the metal layer. This section presents the DRIE method. DRIE uses the same principle as Reactive Ion Etching (RIE), but it repeats the RIE process multiple times. Therefore, RIE is first presented and subsequently the DRIE process.

A typical RIE system consists out of a cylindrical vacuum chamber, ionized gas, and a wafer platter that is situated at the bottom portion of the chamber (see Figure C.4(a)). An ion is an atom or molecule where the number of electrons is unequal to the total number of protons resulting in a positive or negative electrical charged atom. Gas is ionized by a high power radio frequency

coil antenna. Ionized gas (positive charged) enters the chamber through a small inlets at the top, and exits the chamber via a vacuum pump at the bottom. The wafer sits on a powered electrode, which is negatively charged. This setup accelerates the positive ions towards the surface of the (negative charged) wafer. The ions react chemically with the materials on the wafer, but also knocks off (sputter) some material by transferring some of their kinetic energy [72, 73]. The walls of the etched hole are nearly vertical (anisotropic), because the ions are accelerated straight down. Only certain parts of the wafer are being etched, because the wafer is pre-patterned with a protective coating. Therefore, only the unprotected parts are etched, and after etching the protective coating should be removed.

The reactive ion etching (RIE) method is also known as dry etching or plasma etching. RIE is based on fluorine plasma chemistry and is used for micro electro mechanical systems (MEMS) [15, p.47]. The main advantage of etching is that all the TSVs are fabricated at once and smaller TSVs are possible (<10μm), compared to laser ablation. Therefore, the TSV density is high and DRIE has, compared to wet etching, nearly vertical walls (but in the long run still a conical shape).

The disadvantage is that there are multiple complex steps needed to perform RIE and DRIE, compared to laser ablation, which takes a long time (3-6μm deep / per minute) [1, p.88]. For an average wafer (100μm thick), etching would take around 25 minutes. However, when a wafer is thinned, the substrate is typically only 10μm thick, which results in an etch process of less than 2 minutes [1, p.88]. Substrate thinning has a second advantage, namely that the area that a TSV occupies decreases, because the TSV is less deep. Due to the slight conical shape, it is possible to use a smaller diameter.



(a) A basic Deep Reactive Ion Etching principle. [15]

(b) A basic deep reactive ion etching process. [15]

Figure C.4: Deep reactive ion etching process

As stated previously, deep reactive ion etching (DRIE) uses the same principle as reactive ion etching (RIE), but it repeats the RIE process multiple times. This is because a wafer has a deep substrate. DRIE is also known as a Bosch process, time-domain multiplexed etching and as a switched etching process [15, p.49]. After the protective photo-resist layer is placed, the wafer

is bombarded with ions. Afterwards, a passivation layer is applied to stop any chemical etching process (see Figure C.4(b)). During the next etching phase, the ions bombard the passivation layer (see Figure C.4(b)). When the ions reach the bottom of the TSV, they etch away the passivation layer, and afterwards the exposed substrate of the TSV (see Figure C.4(b)).

C.2 Bonding of tiers

The individually manufactured tiers should be bonded together to form a 3D stack. In this section, four bonding processes are presented. These are metal-to-metal bonding, eutectic bonding, oxide-to-oxide bonding, and adhesive bonding. These bonding methods are used for bonding two (or more) tiers, containing bumps or TSVs as interconnect.

C.2.1 Metal-to-metal

There are two approaches to bond metal contacts (micro bumps or TSVs) together. The first approach is the surface-activation bonding method, and the second approach is the thermal compression metal bonding. The thermal compression metal bonding is also known as the metal-to-metal bond (such as a copper-to-copper (Cu-to-Cu) bond) and this approach is more popular than the surface-activation bond. The metal-to-metal bonding could bond any metal objects onto each other, such as a TSV to a landing pad, a bond pad to a bond pad and a bump to a bump.

C.2.1.1 Surface-activation bonding

Surface-activation bonding is based on the adhesive force between two atomically clean solid surfaces [1, p.118]. There is no other material used to complete the bond or no elevated temperatures to melt the metals. It is purely based on the principle that all surfaces have an adhesive force between them. The bonding process is completed when two ultra-clean surfaces touch each other. This is done at room temperature and under ultra-high vacuum conditions ($\approx 10^{-8}$torr). The bond results in a strong and void-free bond. The main advantage is that it can be applied to all materials at room temperature, because all surfaces have adhesive forces. However, this process is not preferred (used) at mass production, due the process requirements, such as integration of cleaning equipment, Argon ion beam and an ultra high vacuum conditions in the bond chamber [1, p.118].

C.2.1.2 Copper-to-copper

Copper-to-copper (Cu-to-Cu) bonding uses a downwards force and heat to bond two metal contacts together. No other (adhesive) materials are used, except the two metal contacts [1, p.118]. The heat and pressure enables a deformation of the microscopic contact between the two copper regions, which increases the contact area. The copper structures diffuses into each other when they have enough thermal energy, hence no adhesive material is needed. This process does not have high requirements, due to the elevated temperature. Requirements, such as a very clean surface or ultra high vacuum conditions are not needed. The simple and low requirements make it more attractive for the use in the industry and academia, compared to surface-activation bonding.

The main parameter for this approach is the temperature, since this should not affect / damage the tiers especially the metal wires melt / degrade at high temperatures ($\approx 500 - 600$ degrees).

It has been demonstrated that a successful cu-to-cu bond, can be manufactured with a pressure of 400mbar, vacuum of $10^{-3}torr$ and bonded for 30 minutes at 400 degrees [1, p.119].

C.2.2 Eutectic bonding

Eutectic bonding uses melted metal to connect two metal contacts with each other. An example of this bonding approach is the Solid-Liquid InterDiffusion (SLID) bonding process, used since 1960 (see Figure 2.13(b)) [1, p.131]. The used metal contacts are usually gold, silver or copper with in between a layer of tin (Sn). Typically, copper contact points are used with a tin layer in between (CU/Sn). The SLID bonding concept is also known as isothermal solidification, Cu/Sn bonding and transient liquid phase bonding.

The solder alloy of a SLID process has a low melting point, but after bonding the bond has a high melting point and stiff joint. It is positive that the joint has a high melting point, so that afterwards the bonding process is irreversible. The temperature reaches the melting point of the solder alloy multiple times, especially when multiple tiers are bonded. If the melting point of the lower joints remained the same, then all the joints of the bonded tiers would melt again. The SLID joint is rigid and stable, and is thus suited for bonding substrates with similar thermal expansion coefficients. However, when the melting point of the alloy is almost reached, then liquefaction does not happen instantly. Liquefaction of the alloy starts at a few points across the wafer. If the downwards force is not precisely centered then the two tiers are bonded with a wedge shape in between (see Figure C.5).

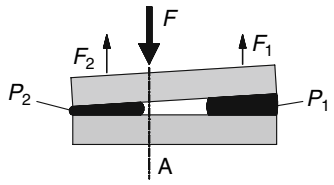


Figure C.5: A SLID bond with a downward force, which is not centered. [1]

C.2.3 Oxide-to-oxide

Two silicon tiers are bonded with oxide, which is a bond between silicon (Si) and oxygen (O) molecules. Oxide-to-oxide bonding is mainly used by the industry for Silicon-On-Insulator (SOI) wafer stacking [15, p.211]. It has been demonstrated that a layer of water molecules spontaneously bonds at molecular level with any silicon dioxide (SIO_2) materials, see Figure C.7.(a). At high temperatures ($\approx 300 - 400$ degrees), the water molecules diffuse progressively through the silica and more silicon and oxygen bonds are created, which increases the bonding strength, see Figure C.7.(b) [15, p.211]. The bonding strength is stronger if the wafer is thinned (grinded), and if the wafer has a planar clean surface.

Chemical Mechanical Polishing (CMP) is used to obtain a planar surface, and uses an abrasive and corrosive chemical slurry. The chemicals in the slurry also react with (weaken) the material, which should be removed. The abrasive accelerates this weakening process, and the

polishing pad helps to wipe the reacted materials from the surface. The dynamic polishing head is rotating with different axes of rotation (i.e. not concentric). This head removes material and makes the wafer planer. This is important, because any imperfection decreases the bond surface.

Lincoln laboratory uses the following oxide-to-oxide bonding process (see Figure C.6) [1, p.172]. Two tiers are bonded face-to-face with an oxide bond at a temperature of 275 degrees Celsius. After bonding the substrate is grinded down (see Figure C.6.(a) and (b)). The stacked tiers are patterned with a protective coating, which enables etching of the un-patterned regions. The through silicon vias (TSVs) are etched through the metal layer, the active layer, the silicon substrate and the oxide-to-oxide bond, until it lands on the metal layer of the lower tier (see Figure C.6.(c)). The vias are filled with the metal tungsten and metal pad is manufactured on top of them. The metal pad functions as a landing pad for the TSV of tier 3(see Figure C.6.(d)). Tier 3 is added back-to-face, following the same process. The TSVs, of tier 3, connects to the landing pads of tier 2 (see Figure C.6.(e)).

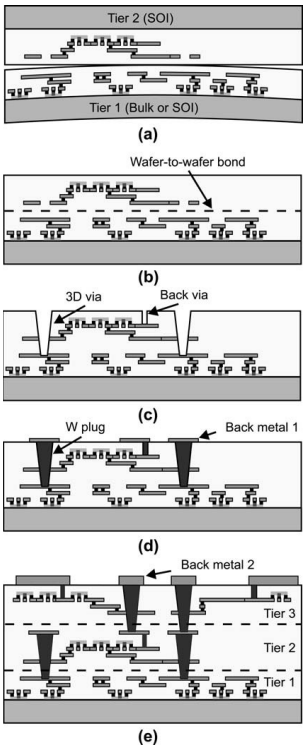


Figure C.6: Oxide-to-oxide-SOI bonding. (a) Two SOI face-to-face wafers, (b) Silicon substrate is removed, (c) 3D vias are etched through the MOSFET layer and bond of tier 1 and 2, (d) Vias are filled with tungsten, (e) Tier 3 is back to face bonded, vias are etched and filled [1].

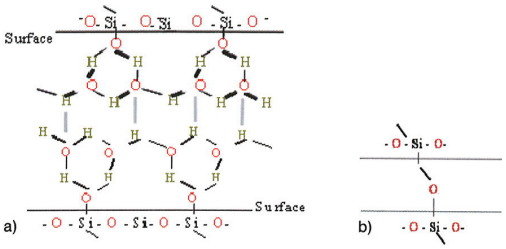


Figure C.7: Oxide-to-oxide dielectric bonding structure of water molecules with silicon molecules (a) At temperatures up to 800 degrees, (b) At temperatures over 1000 degrees. [15]

C.2.4 Adhesive bonding

Adhesive bonding uses a non-electric-conductive liquid or gel solution to bond two tiers together. In most cases, the liquid / gel is a polymer adhesive (benzocyclobutene BCB) and is applied to one or both tiers that need be bonded (see Figure C.8) [1, p.220], [15, 217]. BCB is used, because this polymer is frequently used at chip packaging and therefore well known in the microelectronics industry. After an adhesive is applied, the tiers are joined with a downwards pressure. The polymer is then converted from a liquid / gel in to a solid state. This is done by exposing the polymer to heat or UV light. UV light is mainly used for handle wafers. Naturally, the handle wafer should be transparent to UV light. The adhesive reflows during the conversion to a solid state, and it is useful to accommodate surface topography or small particles.

The main advantage is that the process uses a low bonding temperature (≈room temperature-450 degrees); the exact temperature is depended on the used polymer. The polymer layer can be applied in a pattern to enable interconnects, such as bumps or TSV bonding via a metal-to-metal approach. This is also known as a hybrid bonding.

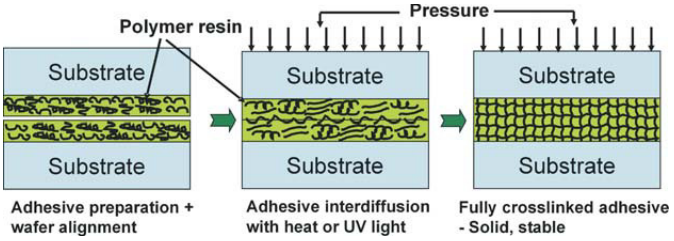


Figure C.8: Adhesive bonding. [1]

C.3 Process sequences in use in the industry

There are many possible process sequences with the fabrication of a 3D chip. This section presents a brief overview of the used / proposed processing steps, done by institutes and companies (see Figure C.9) [15, p.30].

Figure C.9(a) is a schematic depiction of the process steps used by Ziptronix and Tezzaron. Polysilicon TSVs are used by Ziptronix in a FEOL process and copper or tungsten TSVs are used by Tezzaron in a BEOL process. The BEOL and FEOL TSVs reach up to the bottom of the metal (wire) layer of the tier. Although not depicted in the figure, the metal wire layer routes the signal internally up to the top of the metal layer (if necessary). Routing the TSV signals to the top via the metal wires saves area, compared to a full TSV. "Face-down" bonding is used, which means that all the tiers are bonded with the face side down. In Figure C.9(a) face-down bonding is applied from the second tier. When the TSV are exposed, during thinning, they indicate when to stop thinning. Furthermore, TSVs also serve as a wafer alignment marker. The tiers are first bonded and afterwards they are thinned. This removes the need of a handle wafer (support wafer), which is advantageous. However, when a problem occurs at the thinning process, the whole 3D stack is lost. Another disadvantage is that the FEOL and the BEOL TSV process should be done in the IC fabrication process. This means that if a stack consists out of multiple

different tiers (i.e. analogue, processor and a memory tier), all the foundries should support TSV fabrication.

Figure C.9(b) depicts a FEOL and a BEOL process, used by NEC and IMEC respectively. The tier is bonded on a handle wafer and thinned down. With the handle wafer attached to the tier, other backside process steps are performed, such as the creation of bumps. The TSV is connected to the metal wire layer, which routes the signals to the top of the layer if necessary. The thinned tier is bonded face-up and this hold for all the tiers. The same arguments hold for a FEOL and a BEOL process, as stated previously. All the foundries that produced the tiers should support TSV manufacturing.

Figure C.9(c) depicts the process steps used by Tezzaron. This process uses a POST-BEOL via-first approach, which has an advantage, because the IC process is completed, and therefore it is possible to manufacture the TSVs at any foundry. The TSVs are manufactured from the front of the wafer. Therefore, the TSVs occupy space in the metal wire layer, which is a disadvantage. The POST-BEOL TSVs are also larger and complexer to manufacture than the FEOL or BEOL TSVs, because the etch depth is deeper. After the TSVs are manufactured, the tier is bonded and grinded. No handle wafer is needed, because the tier is already supported by the 3D stack. The previously mentioned disadvantage holds here as well: when a problem occurs (during grinding) the whole stack is lost.

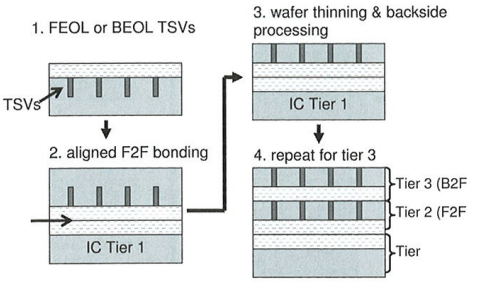
Figure C.9(d) depicts the process steps used by Fraunhofer Munich. A POST-BEOL via-first process is used and the previously mentioned advantage holds here as well (the TSV can be manufactured at any foundry). The TSVs are etched from the front of the tier and the same disadvantages as stated previously holds here as well (the TSVs are bigger and occupies area in the metal layer). After the tier is bonded to a handle wafer, it is grinded and bonded to the 3D stack. The handle wafer is released after the tier is bonded to the 3D stack. The TSVs from different tiers do not necessarily have to be manufactured above each other. The TSVs can be placed anywhere, due to metal landing pads and rerouting of the signals (see Figure C.9(d)) [15, p.33].

Figure C.9(e) depicts the process steps used by Intel and Lincoln laboratories. This process uses a POST-BEOL via-last approach, which means that after the whole IC is fabricated the tiers are bonded, and thereafter the TSVs are etched. All tiers (except the first), are bonded face-down. The tiers that are bonded to the stack and then grinded down. The TSVs are manufactured, and the TSV holes lands on a special metal landing pad in the metal layer. The main advantage is that the TSVs are etched from the back of the tier, which means that it does not occupy any space at the metal wire layer. Furthermore, no handler wafer is needed. However, if a problem occurs when etching the TSVs or grinding the whole 3D stack is lost, which has a high impact on the yield and is therefore disadvantageous.

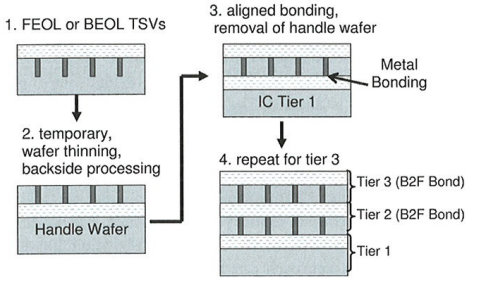
Figure C.9(f) depicts the process steps used by RTI. Fully processed ICs are temporarily bonded to a handle wafer and grinded down. The handle wafer is released after the tier is bonded onto the stack. When the tier is bonded to the stack, the TSVs are manufactured from the front of the tier. This process sequence is also known as POST-BEOL via-last process. All the tiers are bonded face-up. This approach (as stated previously) has a main disadvantage, when a problem occurs at the manufacturing of the TSVs the whole 3D stack is lost, which decreases the yield. Furthermore, the TSVs occupy area in the metal and active layers of the tiers.

Figure C.9(g) depicts the process steps used by IMEC, Zycube and Sanyo. A full process wafer is attached to a handler and is grinded down and afterwards the TSVs are etched from the

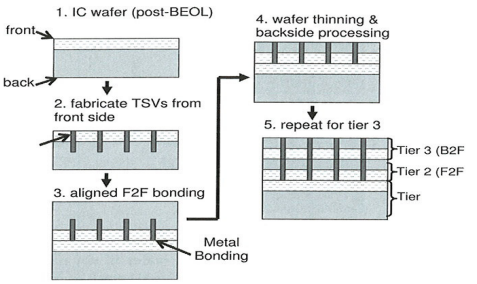
rear of the wafer. This process sequence is also known as a POST-BEOL via-first process. With this approach, the problem described at Figure C.9(f) is solved. Whenever problems occur at the manufacturing of the TSVs, only that tier is affected, and only the unaffected tiers are bonded.



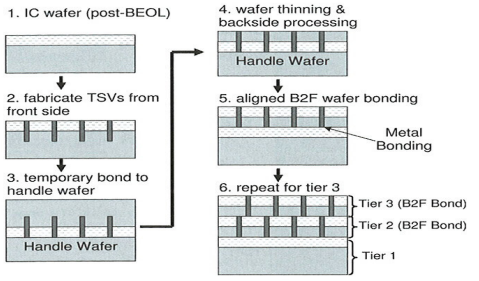
(a) Face-down bonding without a handle wafer, Ziptronix (FEOL), Tezzaron (BEOL). [15]



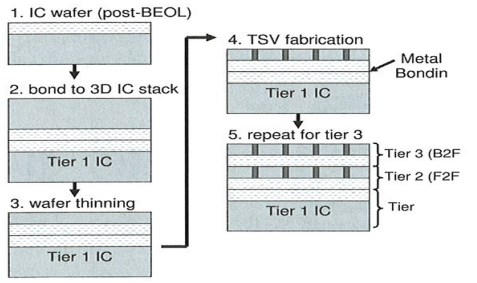
(b) Face-up bonding with a handle wafer, NEC (FEOL), IMEC (BEOL). [15]



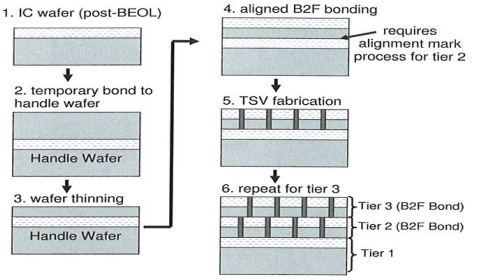
(c) Face-down stacking without a handle wafer, Tezzaron (POST-BEOL via-first). [15]



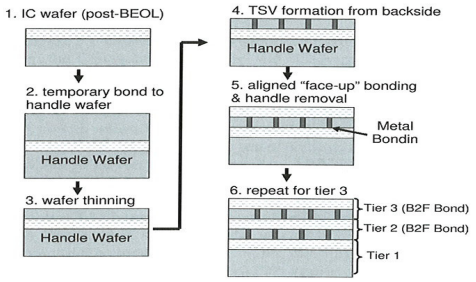
(d) Face-up stacking, with the use of a handle wafer, Fraunhofer Munich (POST-BEOL via-first). [15]



(e) Face-down bonding without a handle wafer, Intel, Lincoln laboratories (POST-BEOL via-last). [15]



(f) Face-up bonding with the use of a handle wafer, RTI (POST-BEOL via-last). [15]



(g) Face-up bonding with the use of a handle wafer, IMEC, Zycube, Sanyo (POST-BEOL via-last). [15]

Figure C.9: Overview of other companies

Consortia focus on TSVs

D

This section shows that the TSV has high potential to become the 3D interconnect for stacked devices, since four consortia are doing research in TSVs and not in other interconnects. There are five main institutes that do research into 3D integration. These are: International Technology Roadmap for Semiconductors (ITRS), Sematech, 3D-IC alliance, EMC-3D and 3D-TSV. The ITRS made a 3D TSV road map [46, p.38] and the other four (individual) institutes are all focusing on developing methods for 3D stacking with TSVs as interconnect (sometimes in combination with a micro bump). Hence, it indicates that TSV has high potential to become the 3D interconnect for stacked devices.

The ITRS is sponsored by chip manufacturers. The sponsors are the European Semiconductor Industry Association (ESIA), the Japan Electronics and Information Technology Industries Association (JEITA), the Korean Semiconductor Industry Association (KSIA), the Taiwan Semiconductor Industry Association (TSIA), and the United States Semiconductor Industry Association (SIA). The objective of the ITRS is: "To ensure cost-effective advancements in the performance of the integrated circuit, and the products that employ such devices, thereby continuing the health and success of this industry". This is conducted through cooperative efforts of the: global chip manufacturers, equipment suppliers, research communities and consortia. The road map teams identify critical challenges, encourage innovative solutions, and welcomes participants from the semiconductor community. In the road map of 2007 and 2008, the ITRS anticipate that 3D TSV, optics or nano wires will replace the current on-chip interconnections [37, p.41], [46, p.38]. The TSV road map predicts, over the period 2008 till 2015, TSVs with a pitch of 3.2-2μm and with a diameter of 1.6-1μm [74, TableINTC6].

SEMATECH (SEmiconductor MAnufacturing TECHnology) is a non-profit consortium that performs basic research into semiconductor manufacturing (not founded particularly for 3D stacking). However, Sematech has a 3D interconnect program, which is focused on TSVs. TSV offers the best of both for achieving very high densities for memory and logic. This is because Sematech believes that TSVs is the best technology for achieving very high densities between memory and logic. TSVs provide better density for the same footprint, higher performance, lower power consumption and lower cost [36]. Members of Sematech include: Hewlett-Packard, Infineon Technologies, IBM, Intel, Micron, NEC, Panasonic, Renesas, Samsung, Toshiba, TSMC and Texas Instruments.

The 3D-IC Alliance is an alliance between Tezzaron and Ziptronix. Its objective is to promote standards for three-dimensional integrated circuits (3D-ICs) in order to accelerate their availability and acceptance. The 3D-IC alliance has released the first standard in June 2008. The Intimate Memory Interconnect Standard (IMIS) specifies the physical interface characteristics for mounting memory onto a host device in a 3D configuration [75, p.3]. This standard consists out of two parts. Part A, specifies a "direct bond interconnect" interface (developed by Ziptronix), which is an oxide-to-oxide process, where the TSVs are etched from the front to the rear of the tier (previously presented by Section C.2.3) [75, p.14], [76, p.487]. Part B presents a

copper-to-copper bonding (developed by Tezzaron) and uses TSVs in combinations with micro bumps [76, p.463], [75, p.14]. These micro bumps are being bonded with a metal-to-metal bond (see Section C.2.1).

The Semiconductor 3D Equipment and Materials Consortium (EMC-3D) was created in September 2006 to develop a new 3D market and technology. EMC-3D currently (summer 2009), consists out of fifteen members such as: Fraunhofer IZM, SAIT (Samsung Advanced Institute of Technology), KAIST (Korea Advanced Institute of Science and Technology), TAMU (Texas A&M University), CEA LETI and NXP. The mission of the EMC-3D is to rapidly develop a cost-effective and manufacturable TSV for 3D chip stacking and MEMS integration. The consortium was originally targeting TSV wafers with a Cost of Ownership (CoO) of less than $200USD, which was achieved in September 2008. The achieved CoO was less than $190 USD / wafer. The TSVs have a diameter of 5μm and a depth of 30μm [77]. In July 2009, EMC-3D has announced that it is able to produce TSV wafers for a CoO less than $150USD per wafer [43].

The Institute of Microelectronics (IME) announced at the 8th of September 2009, the launch of a 3D-TSV consortium. The consortium is supported by the Singapore Economic Development Board (EDB) and A*STAR. The members include: Chartered Semiconductor Manufacturing Ltd., STATS ChipPAC Ltd., United Test and Assembly Center Ltd. The consortium has two main goals, which are executed in two phases and each phase should last eighteen months [78]. The goal of phase 1 is to design TSV processes, assemble and train personnel to support the manufacturing of new devices. Phase 2 demonstrates the integration of fully functional mobile devices, for TSVs on a 300mm wafer. No further information is known, because this consortium is announced at the same time this thesis is written.

Memory-on-memory

E

E.1 Reducing TSVs

The number of TSVs should be minimized in a design to accommodate the potential pitch size mismatch between wordlines / bit-lines and Through Silicon Vias (TSVs) [16]. Pitch size mismatch occurs if the TSV diameter cannot scale down as much as the circuit technology. It is because a scaled down TSV requires a higher aspect ratio (depth and width ratio), since the diameter (width) is reduced and the depth remains the similar to the current situation (<100μm depth). Only minor depth reduction is expected by the author if this thesis, since the depth is already reduced by grinding. This high aspect ratio is difficult to manufacture with the current technologies, such as Deep Reactive Ion Etching (DRIE) and laser ablation. By having fewer TSVs it is possible to use TSVs with larger diameters, which are easier and cheaper to fabricate [16]. Therefore, this architectural strategy tries to limit the number of TSVs with a inter-layer bus (see Figure E.1(a)). Conversely, column and row stacking need a TSV for every stacked wordline or bitline. Thus, in total a large number of TSVs (preferably) with a small pitch is required.

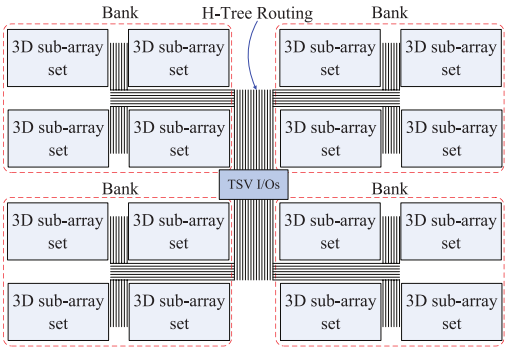
Figure E.1(a) has the same floor plan as a 2D design scheme with the exception that each leaf (3D sub-array set) is divided over multiple layers. Four 3D sub-array sets are repartitioned with FUB repartitioning, and is shown in Figure E.1(b). This scheme is called the Single-Layer Data Access (SLDA) scheme. Contradictory, Figure E.1(c) shows a single 3D sub-array set that is split, via logic gate splitting. This scheme is called the Multi-Layer Data Access (MLDA) scheme. The main difference between both schemes is that for the SLDA scheme per (individual) address the data width is read / written to one layer. Conversely, the MLDA scheme divides for each (individual) address the data width uniformly over all the layers (vertically). For example, with (MLDA)32 bits wide data of is written / read to / from an single address, then each layer processes eight bits (8= 32/4, four layers are assumed). Conversely, the 32 bits are written / read to / from only one layer with the SLDA scheme.

Each 3D sub-array set has its own (split or stacked) address decoder, wordline / bitline driver and sense amplifiers. They are identical to the 2D counterparts, to keep the redesign low. The 3D sub-array sets share only the address and data bus. This requires much lower number of TSVs, compared to row and column stacking.

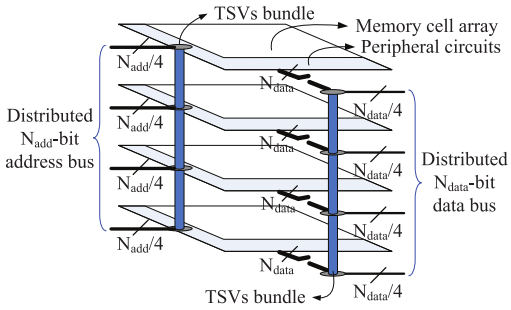
In Figure E.1(b) and Figure E.1(c), N_{add} and N_{data} denotes by the 2D address width and data bus width, respectively. It is assumed that they can be divided by 'n' planes.

E.1.1 Area, latency and energy results

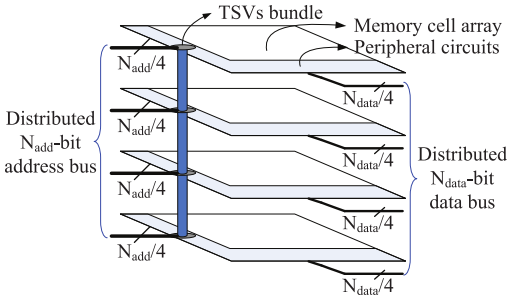
A modified CACTI5.0 simulator is used to simulate a 1 Gb DRAM device with 8 banks, 256-bit data I/Os, and 10μm x 10μm TSVs at 65nm process technology. For comparison, the results of the SLDA and MLDA are compared to a 2D and a 3D scheme where the memory layers are



(a) 3D DRAM with shared data bus and address bus. [16]



(b) Four 3D sub-array set, having each an own layer. Single-Layer Data Access (SLDA) scheme where only one sub-array / layers is activated at a time. The data width is located over one plane (horizontal direction). [16]



(c) A single 3D sub-array set, divided over multiple layers. Multi-Layer Data Access (MLDA) scheme where all the sub-arrays / layers are activated per operation. The data width is split up and divided over all the planes (vertical direction) [16]

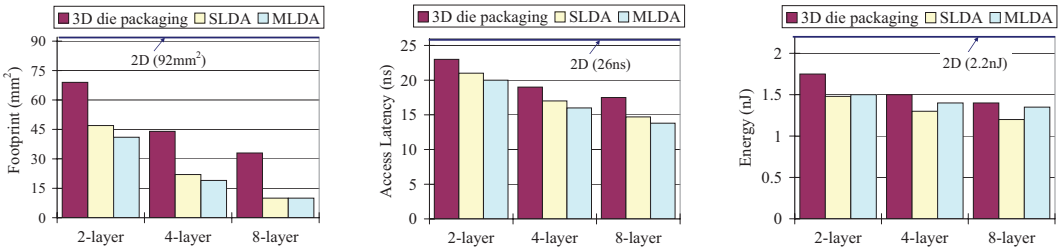
Figure E.1: The banks and 3D sub-array sets have a shared data bus and address bus with TSV pillars in the middle. N_{add} and N_{data} denotes the 2D address and data bus width, respectively.

bonded by bondwires (instead of TSVs). However, the bondwires parasitics are not taken into account on the delay and energy analysis for simplicity purposes [16]. If the bondwires parasitics where taken into account then that would lead to a higher speedup for the SLDA scheme and MLDA scheme and it can thus be neglected.

The simulation result indicates that the SLDA scheme consumes less energy compared to the MLDA scheme (see Figure E.2(c)). This is because of the difference between the SLDA and the MLDA designs. With the SLDA, N_{data} TSVs are used to connect all the 3D sub-array layers to the main data bus. Conversely, the MLDA uses a true 3D processor, that is where all the data widths are uniformly divided over all the planes, and hence it does not need any TSVs at the data bus. Thus, the SLDA scheme uses (extra) TSVs at the data bus which will consume extra power, compared to the MLDA scheme (see Figure E.1(b) and Figure E.1(c)) [16]. However, [16] indicates that the MLDA uses all the layers per data access and the SLDA schemes uses only one layer, and it shuts down the remaining unused layers to save energy. Nevertheless, the results indicate the SLDA scheme is more energy efficient.

It is assumed that the data bus of the MLDA scheme does not contain any TSVs, which saves area. Thus, the MLDA scheme has a better foot print reduction and latency performance, com-

pared to the SLDA scheme (see Figure E.2(a) and E.2(b)). The same lack of TSVs (capacitance) gives the MLDA design its (slightly) faster access latency time, compared to the SLDA design. However, the author is this thesis thinks that when both schemes have the same number of TSVs, then the energy consumption, area, and access latency is similar.



(a) Footprint area. (b) Access latency time. (c) Energy consumption.

Figure E.2: Estimated results of (a) Footprint, (b) Access latency, and (c) Energy consumption for the 1 Gb DRAM design using different design approaches. Note that the line in top of all the figures indicates the result of the 2D design. [16]

Logic-on-logic

F

F.1 Pipeline reduction and thermal herding

This section shows that the pipeline stages of a 3D stacked Intel Pentium 4 can be removed, and that increases the performance with 15% and the temperature with 14°C (compared to the 2D case). Furthermore, it is explained why how to achieve a temperature neutral processor.

F.1.1 Intel Pentium 4

The floor plan of a 2D Intel Pentium 4 is depicted in Figure F.1(a). This processor is deeply pipelined and has a branch miss penalty of more than 30 clock cycles. This penalty is predominantly caused by the long pipelines and it is thus beneficial to use 3D integration to remove these [11]. The 3D Intel Pentium 4 processor is divided over two layers by splitting and stacking the FUBs to achieve wire length reduction[1]. The wire length reduction makes many of the original pipeline stages superfluous. The frequency of the 3D processor is kept equal to the original planar processor and thus the increase in performance comes solely from the removed pipeline stages. The top and bottom dies are depicted in Figure F.1(b) and Figure F.1(c), respectively.

With the first example, two FUBs are stacked on top of each other to reduce the number of pipeline stages and power consumption of the inter-FUB (FUB-to-FUB) communication. The load-to-use delay (fetch delay) is critical to the overall performance, such as in Figure F.1(a). This correspond to the path between the first level data cache (D$) and the functional units (F). This path is indicated by the bold arrow in Figure F.1(a). The worst-case path occurs when (load) data must travel from the far (left) edge of the data cache across the data cache and then to the farthest functional unit (on the right hand side). This wire delay takes one clock cycle and is in addition to the access time of the data cache [17, p.95]. An overlap between the data cache and the functional units is created by stacking the top die onto the lower die. In this new layout the (worst-case) load data travels from (any place inside) the data cache to the center of the data cache and is then routed upwards via the TSVs to the center of the functional units. As result the worst-case path is reduced with (at least) 50%, since the data is only traversing half of the data cache and half of the functional units. Hence, one clock cycle of the load-to-use delay is eliminated. Cache-on-logic stacking is also favorable for thermals [11] because caches are relative cool structures and the functional units are relative hot structures. The simulation results are discussed in section F.1.2.

With the second example, the Single Instruction Multiple Data (SIMD) unit is (intentionally) placed between the Floating Point (FP) and Register File (RF) units at the planar layout because the architecture is optimized for the critical SIMD applications (see Figure F.1(a)) [11]. This SIMD unit adds two latency cycles to all the FP instructions. In the 3D layout the FP unit is

[1] It is not indicated by [11, 17] which FUB are stacked (by FUB repartitioning) and which are split (by logic gate splitting), except for two examples which are presented in this section.

stacked onto the RF and the SIMD unit (see Figure F.1(b) and Figure F.1(c)). This eliminates the two cycles of wire delay compared to the original situation and thus improving the performance of all FP applications, while not harming SIMD applications [11]. The simulation results are discussed in section F.1.2.

F.1.2 Simulation results

Two simulators are used to test the wire delay, performance and temperature. The performance simulator from the Pentium 4 design team is used to accurately model the wire delays [11, p.3] and the overall performance. With this simulator 650 (single threaded) benchmarks are executed, such as SPECINT and SPECFP, hand written kernels, multimedia, productivity, server, and workstation applications. The other simulator, also made by Intel, is used to simulate the temperature of the 3D model. The modeling tool considers the heat sink, integrated heat spreader, die, bonding layer, metal wires, package, socket, and the motherboard [11]. Both designs (2D and 3D) are assumed to have a heat sink on top with forces convection air over it.

With FUB repartitioning, 25% of all pipe stages in the microarchitecture are eliminated and this results in a 15% overall performance improvement. The performance improvement is due to the elimination of the wire delay and (consequently) the pipeline stages. Besides the overall performance improvement and the smaller footprint, the power consumption is reduced by 15%. The temperature (112.5°C) only increases with 14°C (compared to the 2D situation), because of the power reduction and the heat sink on the top. Thus, by reducing the pipeline stages the performance increased with 15% and the temperature with 14°C (compared to the 2D case).

The power, performance and temperature of a chip are controlled by scaling the supply voltage and the clock frequency. By scaling voltage and / or frequency it is possible to trade-in a part of the performance to lower the temperature of the 3D processor (see Table F.1). Subsequently, by adjusting these parameters it is even possible to achieve a temperature neutral 3D processor, while achieving power and performance improvements of respectively 34% and 8% (see Table F.1). Conversely, a high power reduction of 54% is achieved by equalizing the performance , in respect to the 2D processor. The low power consumption and the reduced footprint is an appealing optimization for the handheld market. Thus, the performance improved while having a lower power consumption or a similar temperature, in respect to the planar processor.

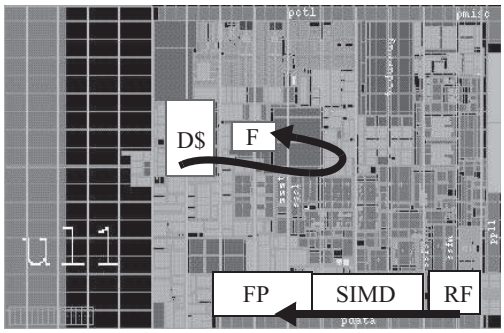
Table F.1: The 3D results are in respect to the 2D situation and the bold numbers are indicating similar 2D values.

	Power (%)	Temperature (%)	Performance (%)	Vcc (%)	Frequency (%)
2D baseline	100	99	100	1	1
Same Pwr	**100**	127	129	1	1,18
Same Freq.	85	113	115	1	**1**
Same Temp	66	**99**	108	0,92	0,92
Same Perf.	46	77	**100**	0,82	0,82

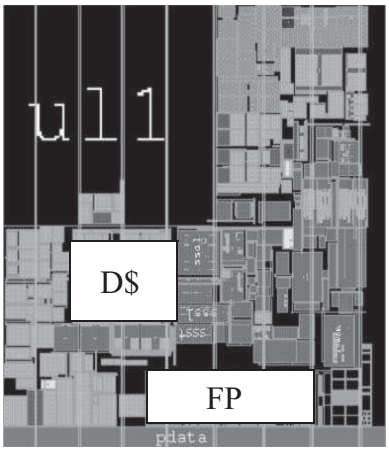
F.2 Improved frequency, larger structures and a cross-cluster bypass

This appendix shows that the wire reductions are used to increase the frequency or the cache size of a processor.

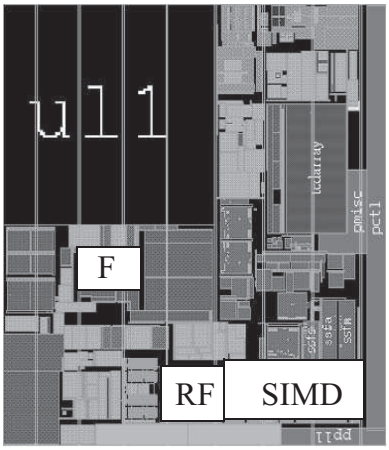
This appendix is an summary of an article [58] and a journal [17]. They discuss the same proposal and are published in the same year. Moreover, they have one author in common, and thus it is assumed that they describe the same proposal. The summary is made because some information is only presented in the article and some only in the journal. In Sections F.2.1, three 3D floor plans of the Alpha 21364 processor are presented. The subsequent sections preset the simulation methodology and simulation results. This is done for the latency, performance and thermal simulations, which are presented in Sections F.2.3, F.2.4, F.2.6, respectively. The latency simulation results determines the new (faster) frequency. With this new frequency the performance simulation is executed as well for two simulation configurations. Finally the impact on the temperature is evaluated.



(a) Intel Pentium 4 planar processor layout. The bold arrows indicate the data path. [11]



(b) Top die of a 3D Intel Pentium 4 processor. [11]



(c) Bottom die of a 3D Intel Pentium 4 processor. [11]

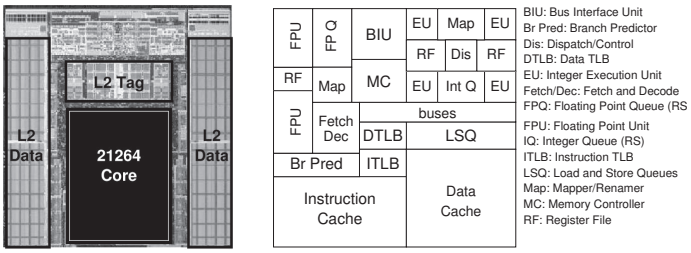
Figure F.1: A 2D and 3D Intel Pentium 4 processor.

F.2.1 The 3D Alpha 21364 processor floor plan

The main cores of the Alpha 21364 processor consist out of a 21264 core and L2 caches. The Alpha 21364 processor is shown in Figure F.2(a) and the L2 caches flanks the 21264 core on both sides. In this section, an Alpha based core is used that is microarchitecturally identical to the 21364 core, which means that it has the same number of physical registers, functional units, issue queue sizes, etc [17, p.87]. A more detailed layout of an Alpha based core is depicted in Figure F.2(b). In the remaining part of this section the name Alpha 21364 processor / cores refers to the Alpha 21364 *based* processor / cores.

All the critical paths of the Alpha 21364 processor are reduced. SRAM-based blocks such as caches, but [17, 58] does not present which stacking method is used (i.e. core stacking, column or row stacking). Logic blocks, such as arithmetic and logical units, are stacked via Functional Unit Block (FUB) stacking or logic array splitting depending on the benefits and the feasibility. Unfortunately, for the rest of the cores it is not presented in [17, 58] how they are stacked / split (i.e. core stacking or FUB repartitioning) [17, p.87].

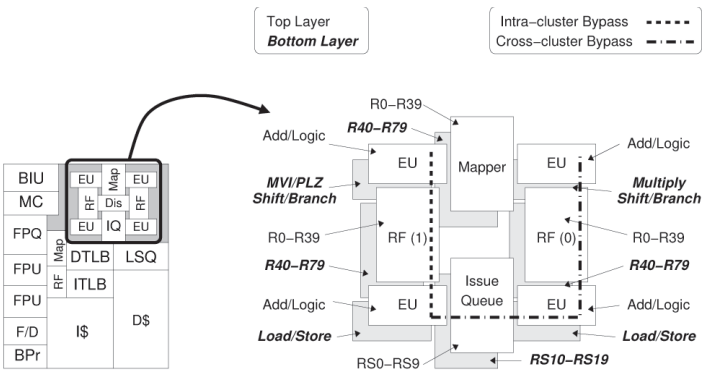
The whole 21364 core is compacted into a 2-layer and a 4-layer floor plan [17, p.88]. The integer execution core is depicted in detail in Figures F.3(a), F.3(b), and F.3(c) without the dispatch and control logic cores (located in the center) for clarity. The 3D 2-layer 21364 core has only 50% of its original footprint [17, 58] (see Figure F.2(b) and Figure F.3(a)). The first 4-layer layout of the 3D 21364 core is presented at Figure F.3(b) and it is similar to the 2D case, except when in the two layer case two planar FUB are stacked on top of each other (via FUB repartitioning). In that case each planar FUB is divided over two layers that form a block. Two (double layer) blocks that are stacked on top of each other results in four layers (see Figure F.3(b)). Another alternative 4-layer floor plan is depicted in Figure F.3(c). This layout is identical to Figure F.3(b), except for the integer execution core. The left and right integer Execution Unit (EU) cores and the Register File (RF) core form two separate clusters. These clusters are stacked on top of each other (see Figure F.3(c)). This horizontal cross-cluster bypass is indicated by the arrow '$\beta \rightarrow$' at Figure F.3(b) and it is replaced by TSVs in Figure F.3(c). In the remaining part of this section the floor plans of Figure F.3(b) and Figure F.3(c) are referred to as the 4A-layout and the 4B-layout, respectively.



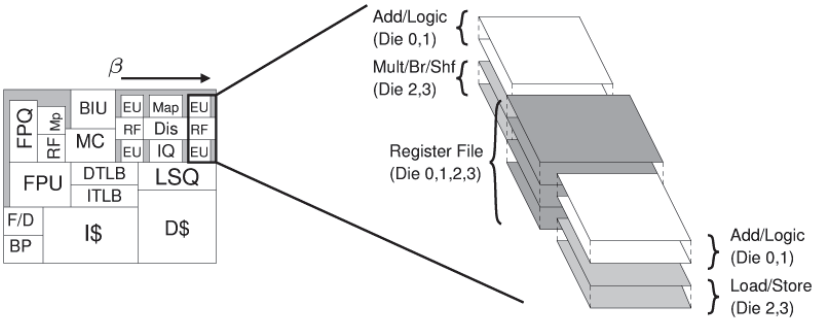
(a) The 2D floor plan of an Alpha21364. [17]

(b) The 2D core of an Alpha 21364 processor. [58]

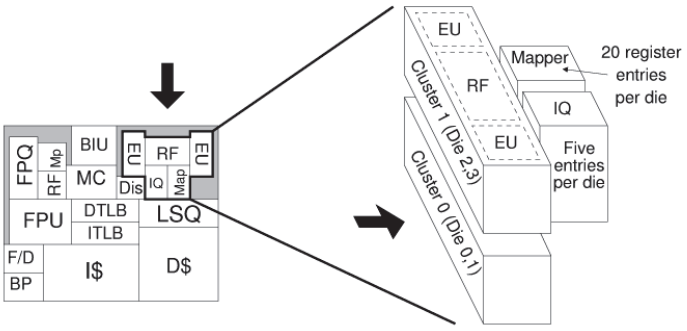
Figure F.2: 2D Alpha 21364 layout. The figures are not proportional with respect to each other. [17]



(a) On the left side, a two layer Alpha 21364 3D Core on the left side. The footprint is 50% smaller than the 2D layout. The right side depicts the integer execution core. [17]



(b) On the left side, a four layer Alpha 21364 3D Core (4A-layout). The footprint is 75% smaller than the 2D layout. The right side depicts a particular part of the integer execution core. [17]



(c) An alternative four layer floor plan (4B-layout). The left and right EU and RF cores are stacked on top of each. The orientation is indicated by the bold arrows. [17]

Figure F.3: An Alpha 21364 divided over two and four layers. The figures are not in perspective with respect to each other. [17]

F.2.2 Simulation results of the Alpha 21364 processor

First, the latency simulation results are presented and thus the new (faster) frequency is determined. With this new frequency the performance simulation is executed together with other simulation configurations. Finally, the impact on the temperature is evaluated.

F.2.3 Latency simulation results

In Hspice, the 3D Alpha 21364 core is simulated for the two and four layer layouts and the results are depicted in Figure F.4(a). The 3D cores exhibit different speedups, as shown in Figure F.4(a). If the core is wire dominated, then the speedup is larger than whenever the core is gate dominated [17, p.90]. The lowest improvement gives the upper bound for the frequency increase, which is 10.3%, and 15.1% and stands respectively for the 2-layer and 4(A&B)-layer case (see Figure F.4(a)). Note that both of the four layer layouts have the same intra-core delay. The only difference in the 4B-layer case is the inter-core delay between the clusters. This horizontal cross-cluster bypass saves 1-cycle communication time [17, p.92]. Thus, for both four layer cases (4A & 4B), it is possible to increase the clock frequency with 15.1% compared to the 2D Alpha 21364.

Table F.2: The size of different structures are listed. They are used by the three configuration types. [17]

Module	2D	3D 2-layer Max size	3D 4-layer Max size
L1 Cache	64KB	128KB	256KB
L2 Cache	1MB	1MB	2MB
BPred - (Local/Global/Meta)	(2/1/1KB)	(2/2/2)	(2/4/4)
Load/Store Queue	32 each	44 entry	80 entry
Int RAT	80 regs	120 regs	160 regs
Int Issue Queue	20 entry	40 entry	80 entry

F.2.4 Performance simulation methodology

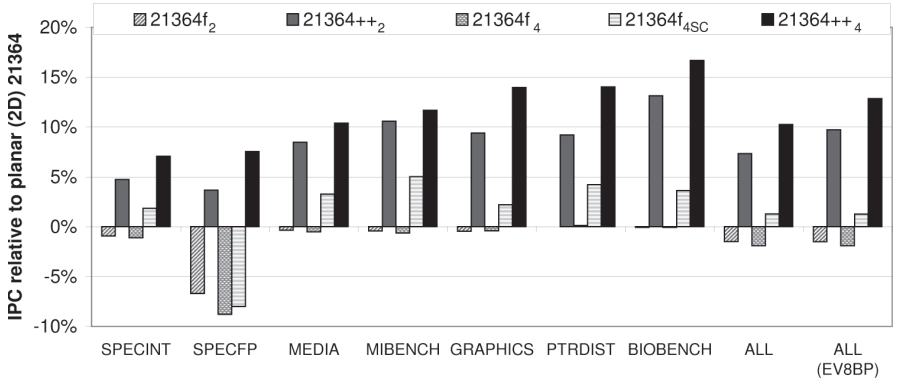
MASE from SimpleScalar is used for the Instruction Per Cycle (IPC) and the overall performance simulation. The overall performance and Instruction Per Cycle (IPC) are simulated for all the three presented 3D Alpha 21364 processor layouts (see Figure F.3). These layouts are divided over three different configuration types.

The first configuration type uses the 2-layer and the 4A-layer layouts, they are simulated with a higher clock frequency of 10.3% and 15.1%, respectively, and with the same structure sizes (registers, load/store queues, ect.) as the original 2D processor. This configuration is denoted in Figure F.4 by $21364f_2$ (f denotes fast) and $21364f_4$, and it stands for the 2-layer and the 4A-layer layouts, respectively.

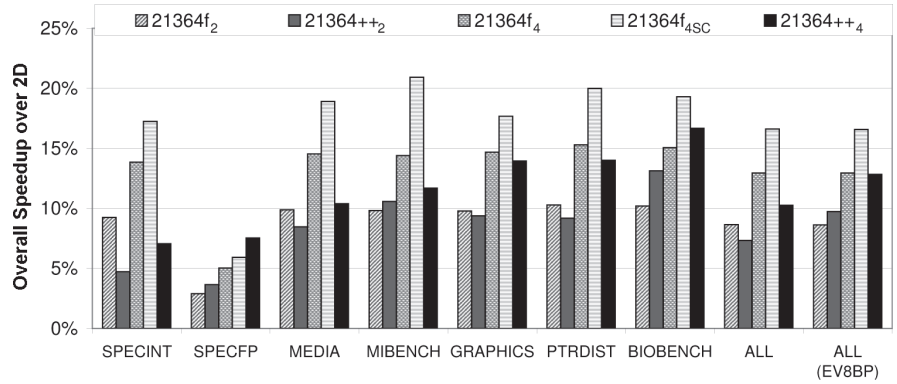
The second configuration type uses the same 2-layer and the 4A-layer layouts in the simulation, but with one difference. It uses larger structures in an attempt to expose more IPC / less cache misses, compared to the $21364f_2$ and $21364f_4$ processors (see Table F.2) [17, p.92]. The

	L1 Cache (64KB)	L2 Cache (1MB)	BPred - (Local/Global /Meta:2/1/1K	Load/Store Queue (32 each)	Int RAT (80 regs)	Int Issue Queue (20 entry)	Intra-Cluster Int Bypass	Cross-Cluster Int Bypass
3D 2-layer	24,50	20,20	12,30	26,50	30,50	11,00	10,30	30,70
3D 4-layer	36,30	40,10	19,10	38,80	45,60	15,10	15,30	45,00

(a) Latency improvement of the Alpha 21364 (2-layer and 4-layer) cores. The latency improvement is relative to the 2D situation.



(b) The IPC impact on the simulated layouts. [17]



(c) Overall performance when the IPC and frequency are taken into account. [17]

Figure F.4: The latency, IPC and overall performance simulation results.

size of the larger structures is specifically chosen so that they run at the same frequency as its original 2D counterpart. It is because the speedup at the overall performance comes solely from the improved IPC and not from the higher frequency (see Table F.2). This configuration is denoted in Figure F.4 by the 21364_{++2} and 21364_{++4} that stands for the 2-layer and the 4A-layer layouts, respectively.

The third configuration type uses the 4B-layer layout, where the cross-cluster bypass delay (1-cycle) is removed, in comparison to the 4A-layer $21364f_4$ simulations. It is because the '$\beta\rightarrow$' communication distance at Figure F.3(b) is replaced by TSVs (see Figure F.3(c) and the TSV delay is neglected. This simulation is only simulated for the 4B-layer processor (and not for the other presented processors). This configuration is denoted in Figure F.4 by $21364f_{4sc}$.

F.2.5 IPC and overall performance simulation results

The Instruction Per Cycle (IPC) and the overall performance are depicted in Figure F.4(b) and Figure F.4(c), respectively. The benchmark 'ALL' denotes the average simulation result, and the benchmark 'ALL EV8BP' denotes the average simulation result which is using a better branch predictor (the individual 'EV8BP' results are not depicted). In the remainder of this section the benchmark 'ALL' is discussed for the IPC and the overall performance when it is not specified. To calculate the overall performance the IPC and the clock frequency are taken into account [17, p.91].

The IPC for the fast processor configurations ($21364f_x$) decreases, as depicted in Figure F.4(b). This is because the clock frequency of the (off-chip) main memory remained unchanged, while the on-chip clock frequency is higher (than the off-chip frequency). This results in a larger penalty (in clock cycles) at a cache miss, compared to the original 2D situation. However, this IPC reduction does not mean a reduction in the overall performance. The overall performance for the fast processors increases, since the $overal\_preformance = IPC * frequency$ and it is illustrated in Figure F.4(c). This is because the fast configurations are faster clocked [17, p.92].

The processors with larger structures (21364_{++x}) performed worse in the overall performance, compared to the faster processors ($21364f_x$) (see Figure F.4(c)). The 2-layer $21364f_2$ processor obtained a speedup of 8.65%, while the 2-layer 21364_{++2} processor obtained a speedup of 7.33%. It is even worse in the 4-layer case, a speedup of 12.84% is obtained for the $21364f_4$ processor, which is much better compared to a speedup of 11.65% obtained at the 21364_{++4} processor. This is because the performance of the branch predictor did not scale with the larger structures (registers, load/store queues, ect.), which leads to an underutilization of the large resources [17, p.93]. Therefore, a more accurate EV8 branch predictor is used. This resulted in a speedup of 9.73% and 12.95% achieved at the 21364_{++2} and the 21364_{++4} processor, respectively (see Figure F.4(c) at the benchmark 'ALL EV8BP'). It is interesting to observe that the performance difference between the larger structures with a better branch predictor and the faster processor is small. Unfortunately, it is not discussed in the articles [17, 58]. The $21364f_{4sc}$ processor showed the highest speedup compared to the other processor layouts. The overall performance results are shown in Table F.3.

At first sight, the overall speedup may seem modest (max. 16.61%). However, one should keep in mind that by stacking and splitting of FUBs a reasonable performance improvement is obtained [17, 9p.3].

Table F.3: Overall performance results, also shown in Figure F.4(c) at the benchmarks 'ALL' and 'ALL (EV8BP)'. [17]

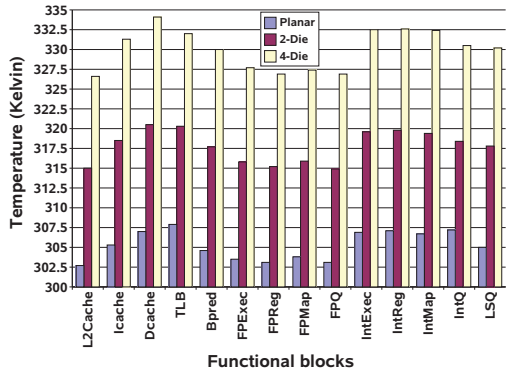
		Overall performance (%)	Clock frequency (%)
2-layer	$21364f_2$	8.65	10.3
	21364_{++2}	7.33	0
	21364_{++2} (EV8BP)	9.73	0
	$21364f_2$ (EV8BP)	8.65 [1]	10.3
4-layer	$21364f_4$	12.84 [1]	15.1
	21364_{++4}	11.65	0
	21364_{++4}(EV8BP)	12.95	0
	$21364f_4$ (EV8BP)	12.84	15.1
	21364_{sc}	16.61	15.1

F.2.6 Thermal simulation methodology

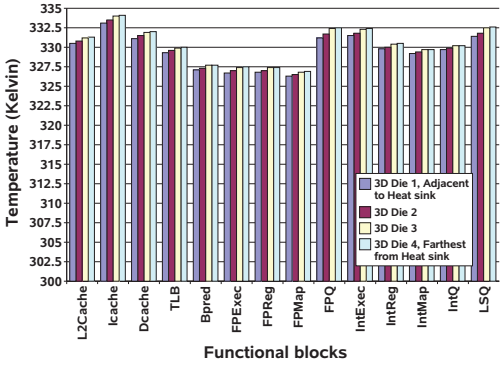
The thermal simulation is presented by [58] for the 2-layer and the 4A-layer processor floor plan. It is not described how the four layer processor floor plan is stacked. The author of this paper assumes that [58] uses the same method as [17] because both articles are from the same authors and year, and in both paper it states that 'the 4-layer processor is similarly partitioned as the 2-layer processor' which is referring to the 4A-layer processor in [17, p.88]. The critical path and latency reduction at the four layer processor are used to speedup the clock frequency [58].

The Hotspot 3.0 simulator is used and it can model multiple layers of silicon and metal on a 3D chip [58]. Hotspot takes the power consumption data and device layer parameters as input and generates the steady state temperatures for the Functional Unit Block (FUB) [58]. The three basic sub-layers of a layer (bulk silicon, active silicon and the metal layer) is simulated in Hotspot by [58]. The dies are bonded with metal-to-metal bonding and thus besides the copper TSVs there is an air gap between the two dies. The die-to-die interfaces are modeled as 25% copper occupancy and 75% air [58]. The copper TSVs between two dies also conduct heat and the average Specific Heat Capacity (SHC) and the average Thermal Resistivity (TR) are computed, based on the TSV surface (25%). The SHC is also known as specific heat and it indicates the amount of (heat) energy required to increase the temperature of a unit quantity (J/M^3) of a substance by one unit K Kelvin [79]. TR is the temperature difference across a structure when a unit of (heat) energy flows through it [80] and it is the reciprocal of thermal conductance [81]. The Spice simulator is used to generate power consumption data for the Hotspot tool per Functional Unit Block (FUB). The total power of each FUB is assumed to be uniformly distributed. Note, a two layer devices has only 50% of a 2D planar foot print to dissipate the heat while the power density increases [58]. Hotspot generates steady states temperatures for various functional blocks and [58] executed it for two iterative cycles. The first cycle uses the initial temperature of 77°C and the second cycle uses the result of the first cycle as initial temperature and the result of the second cycle is presented.

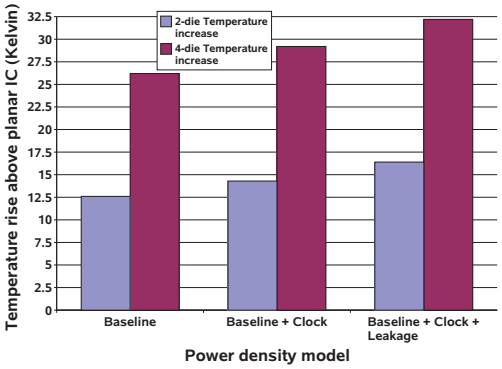
[1] This information is not presented in [17, 58] and thus it is estimated from Figure F.4(c).



(a) Temperatures of FUB on 2D and 3D ICs [58]



(b) Temperatures specified per layer and per FUB for the 4-layer processor. [58]



(c) Temperature increase with clock and leakage power modeling, normalized to a planar IC. [58]

Figure F.5: Temperature increase of an 2D and 3D Alpha 21364 processor.

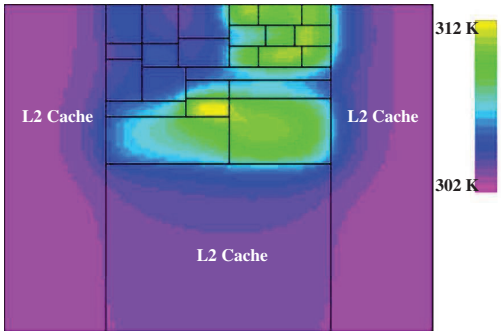
F.2.7 Thermals simulation results

The temperatures of FUBs are shown in Figure F.5(a) and the used processors layouts are the planar, 2-layer and 4-layer layouts, and the maximum temperatures for these layouts are 307.9 K (34.8°C), 320.5 K (47,4°C) and 334,1 K (61.0°C), respectively. The difference between the maximum temperatures of the 3D (2-layer and 4-layer) layouts and the original 2D processor are 12.6 °C and 26.2°C, respectively. This increase is not twice the 2D temperature, because TSVs conduct heat to the heat sink. Secondly, if two functional blocks are stacked then they may not be activated at the same time. This also holds for split FUB, such as a row stacked SRAM, which only enables one row line per data access.

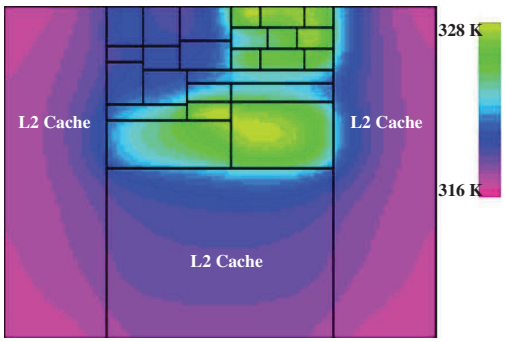
All the processors (2D, 2-layer and 4-layer) in this section have a heat sink on top. This is done to make a fair comparison between the 2D and 3D processors. The temperature of the 4-layer processor is specified per layer and is depicted in Figure F.5(b). The further away from the heat sink the higher the temperature. However, the overall difference between the closest layer (fourth layer) and the farthest layer from the heat sink is not large (<2.5°C). It indicates that the TSVs are efficient in conducting thermals between the layers (see Figure F.5(b)) [58].

The circuit model is refined by adding clock power and leakage power, which leads to a

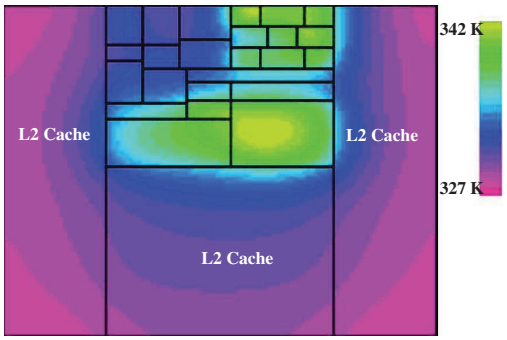
higher temperature (see Figure F.5(c)). The temperatures on the 2-layer and 4-layer, after taking the clock and leakage power into account, increased compared to the 2D processor with 16.4°C and 32.2°C, respectively. These temperature differences are significant, and thus the processor need better cooling mechanisms [58] or different architectural layouts.



(a) Thermal Profile of the Planar Processor. [58]



(b) Thermal Profile of the 2-layer 3D Processor. [58]



(c) Thermal Profile of the 4-layer 3D Processor. [58]

Figure F.6: Thermal profiles of an 2D and 3D Alpha 21364 processor. The figures are not shown in perspective with respect to each other. Purple indicates cool areas and the yellow and green hotter areas.

The thermal profile of the planar IC from Figure F.6(a) shows that the execution unit, the instruction TLB and data cache are the hotspots. These areas are similar to the 2-layer and the 4-layer thermal profiles, as shown in Figure F.6(b) and Figure F.6(c), respectively. Note, the colors in Figure F.6 are inverted to the general analogy, the purple colors indicates the hottest areas and the green/yellow colors indicates the coolest areas. In conclusion, the temperatures increased because of the decrease in the footprint, the larger power density and the heat dissipation paths towards the heat sink are longer [58]. Furthermore, the temperature difference between the top and bottom layer is minor (<2.5°C) due to the heat conductance of TSVs.

3D Network-On-Chip

G.1 Full 3D NOC vs. non-full 3D NOCs

In a 3D stacked device, the NOC should be adapted to enable communication between FUBs at different layers. It is possible to implement a full 3D connected NOC, where all the routers have connections to other layers, as illustrated in Figure G.1(b). However, [12] shows that this is not (always) necessary. The main advantage of a non-full 3D NOC, is a reduction in energy and area. The key factor is to use a healthy mix between 2D and 3D routers. However, the drawback is that the latency increases. This section discusses different configurations of 2D and 3D routers.

In this section a 3D router refers to a router that can route packages from one layer to another and to the same layer. The 3D router uses a 7 x 7 crossbar, whereas the 2D router uses as a 5x5 crossbar. The remainder of the section it is assumed that each layer has the identical 2D and 3D layouts, so that all the 3D routers align perfectly with each other.

In Section G.1.1 various mixes of 2D and 3D routers are presented, and on these configurations simulations are executed. The results are then compared to the full 3D NOC, as well to the 2D NOC (all having the same number of nodes). The main strategy is to trade latency in to achieve lower energy. Thus, lower energy is the key factor. However, the timing results still shows that a full 3D network is not always needed.

G.1.1 The non-Full 3D NOC schemes

Each layer of a 3D NOC grid is assumed to have the dimensions X and Y, and contains K% 3D routers. There are six non-Full 3D NOC schemes simulated with various mixes between 2D and 3D routers. These mixes are uniform, center, periphery, full custom, odd, and side-based.

G.1.1.1 Uniform

A certain uniform distribution pattern of 3D routers is used. The patterns arise by the placement of 2D routers in a full 3D grid and this is done in two steps.

1. As starting point, a 2D router is placed on a single (x, y) position.
2. The four neighboring 2D routers are placed on the positions:
   - (x+r+1, y, z)
   - (x-r-1, y, z)
   - (x, y+r+1, z)
   - (x, y-r-1, z)

the parameter 'r' represents the number of 2D routers among consecutive 3D routers, and 'r' is defined in Equation (G.1). This (uniform) scheme is also referred to as the by_'r' scheme, where the 'r' is replaced by the value of r. Thus, if r=4 then the scheme is called the by_four scheme. The parameter K controls the percentage 3D routers in the design. In Figure G.1(c) an example of this scenario is depicted. With K=25%, meaning that r=3. Thus, in every x and y direction a 2D router is placed after 'r' 3D routers.

$$r = \left\lfloor \frac{1}{K\% - 1} \right\rfloor \tag{G.1}$$

G.1.1.2 Center

All the 3D routers are placed at the center of each layer, such as depicted in Figure G.1(d). Therefore, the remaining routers (at the edges) are 2D routers.

G.1.1.3 Periphery

Contradictory to the center scenario, the periphery scenario only positions the 3D routers at the edges. This is illustrated in Figure G.1(e)

G.1.1.4 Full Custom

The position of all the 3D routers is customized per design, resulting in a perfect match between the needs of the FUBs and the 3D NOC architecture. This solution fits best for the needs of the application, while it minimizes the occupied area and the number of 3D routers. However, the redesign efforts are high, which is costly [12]. Due to the use of various synthetic / generated traffic patterns this scenario falls out of the scope of [12].

G.1.1.5 Odd

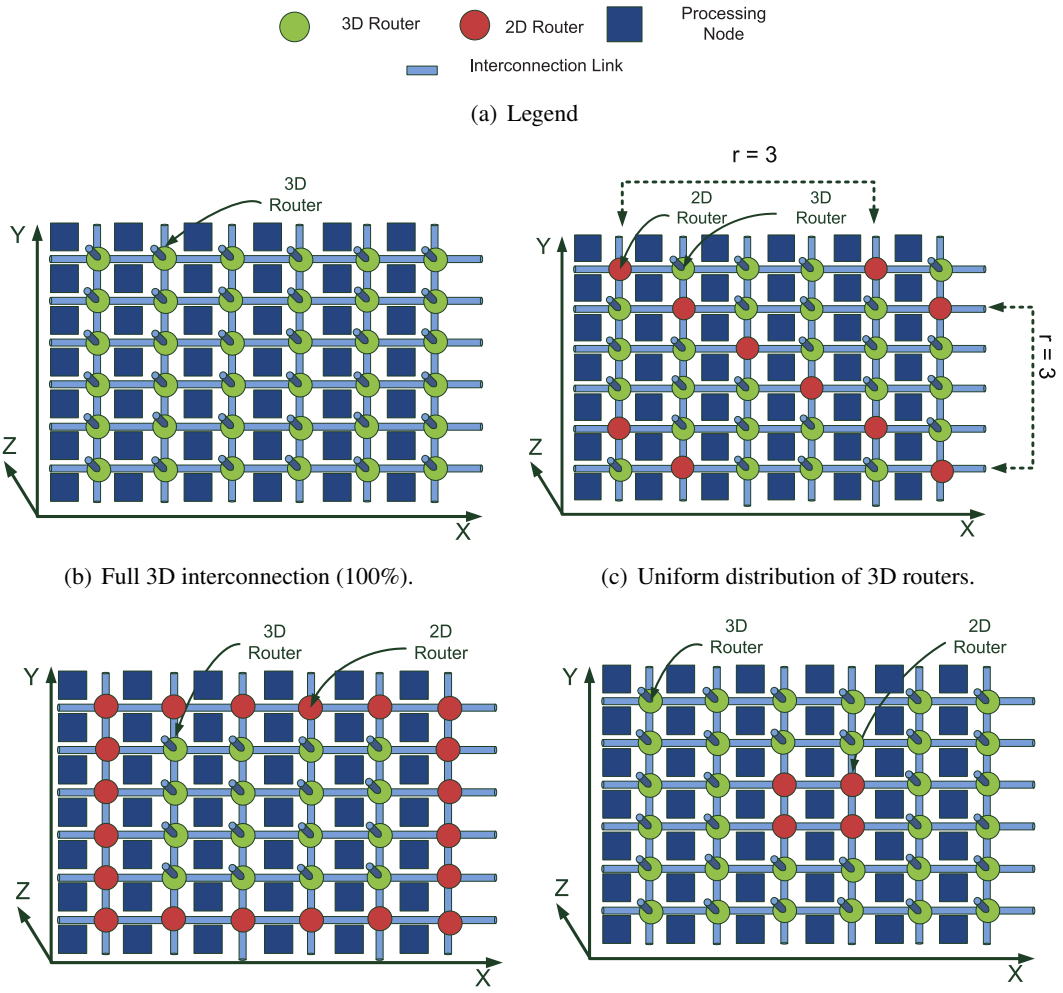
With the odd pattern, all the routers of one row are the same type. Furthermore, two adjacent rows have never the same router type (2D or 3D routers).

G.1.1.6 Side-based

There are three side-based designs and a design contains one, two and three edges with 2D routers along the whole side, respectively. The sides with the 2D routers are located on the north, east and south. The rest of the routers in the network are 3D routers.

G.1.2 Simulation methodology

The presented 2D and 3D routers mixes from Section G.1.1 are simulated with uniform, transpose and hotspot traffic patterns. This is done with 2D and 3D mesh and torus topologies, such as illustrated in Figure G.2. The energy consumption, average packet latency, and the total router (crossbar) area are simulated. The results are then compared to the full 3D NOC, as well to the 2D NOC (all having the same number of nodes). The main strategy is to trade latency in to



(a) Legend

(b) Full 3D interconnection (100%).

(c) Uniform distribution of 3D routers.

(d) Positioning of 3D routers at the center of the NOC.

(e) Positioning of the 3D routers at the periphery of the NOC.

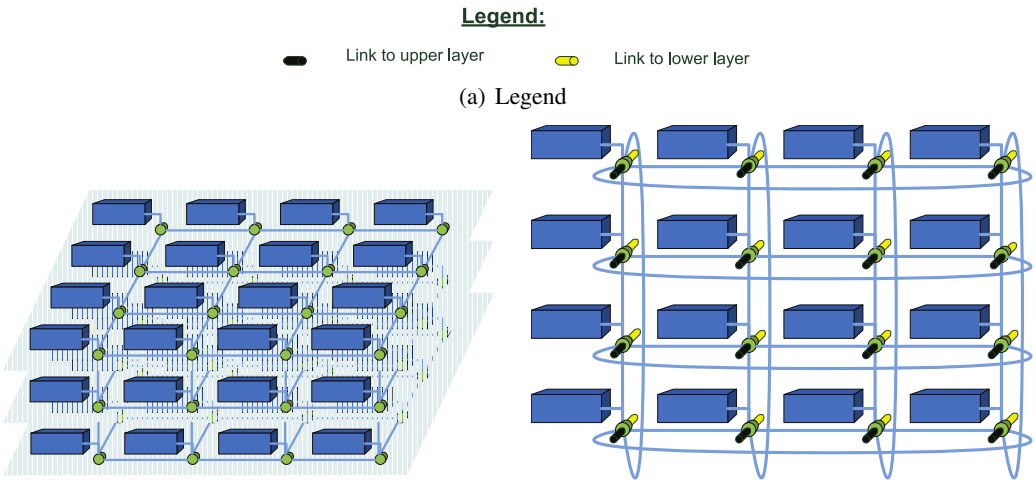
Figure G.1: Various options for a 2D and 3D router mix. It is assumed that each layer contains the exact same (6x6) 3D router layout [12]

achieve lower power consumption (energy). Thus, achieving low power consumption is the key factor.

An adapted Worm_Sim simulator, which uses worm-hole switching, is used for the experiments. The adapted simulator simulates 2D and 3D mesh and torus networks with different grid dimensions and configurations. The dimensions of the simulated 3D NOCs are 4x4x4 and 6x6x4, whereas the 2D NOCs are 8x8 and 12x12. The simulator generates uniform, transpose, and hotspot traffic. With a uniform traffic distribution all the routers / nodes across the 3D NOC receive approximately the same number of packages. With the transpose traffic scheme the data is sent, such that a 3D matrix is transposed. Thus, if the sender has the node coordinate of (x, y, z), then the destination coordinate of the package is (X-z, Y-y, Z-z), where X, Y, Z denotes the dimensions of the NOC. With Hotspot traffic, a minority of nodes receive an increased number

of packages (at least 100% more) than the (majority) remaining nodes. The remaining nodes receive packages in a uniform manner. The hotspot nodes in the 2D grids are positioned in the middle of every quadrant. Whereas, in the 3D NOC, a hotspot is located in the center of each layer.

The energy consumption of the router models is also simulated by the adapted Worm_Sim simulator, including the vertical links [12]. The vertical links is assumed to have the same electrical properties as horizontal wires with the same length [12].



(a) Legend

(b) 3D Mesh with three layers.

(c) A single layer of a 3D Torus. The remaining layers are equal to the depicted layer.

Figure G.2: 3D NOC architecture with legend. [12]

G.1.2.1 Overall result

In the article [12, 52] there is not one particular 3D scheme recommended per individual traffic type, nor did it propose / recommended any scheme in general. No particular network scheme is indicated to be the best for a single or all three traffic types. This also holds for the mesh or torus topologies. However, in general the non-full 3D schemes have better energy reductions for larger networks (144-node mesh NOC), compared to the 64-node mesh NOC. Furthermore, by reducing the 3D routers only gives a minor latency increase with some schemes. However, the non-full 3D schemes can only be used at low load traffic. Otherwise, it results in an increase of the energy consumption and latency.

The general impact for all the schemes are depicted in Table G.1. The table illustrates that there are (in general) energy and area reductions, due to the smaller cross bar, compared to the full 3D connected NOC. An energy reduction between the 1.1% and 12.5% is obtained. Furthermore, the area reduction lies between the 5.3% and the 17.8%. The down side is the extra latency due to the reduced numbers of 3D routers (vertical interconnects), which is between the 2% and 17%. Furthermore, with medium or high traffic loads the full 3D connected NOC performs better than the schemes with a 2D and 3D router mix [52].

Table G.1: General energy, latency, and area results. The positive numbers indicate an increase, and the negative numbers a reduction, compared to the full 3D connected NOC.

64-node architecture	Energy (%)		Latency (%)		Area (%)	
	Min	Max	Min	Max	Min	Max
Uniform	-1.1	-5.4	2.9	14.3	-7.1	-7.1
Transpose	-1.1	-7.1	2.2	8.4	-5.3	-7.1
Hotspot	-3.1	-12.5	2.8	17.6	-5.3	-17.8

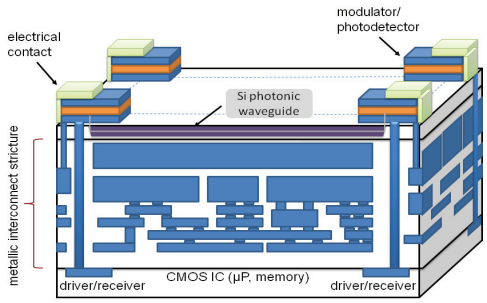
G.2 Photonic Network-On-Chip

A photonic network is also known as an optical network. Photonic communications can meet the large bandwidth demands of high-performance computing systems by bringing the high modulation rates and parallelism of wavelength division multiplexing. Decades ago, the long copper communication cables are replaced by optical fibers in data centers. Moreover, the copper cables that connect racks in data centers are being replaced by optical fibers. Following the same trend, optics can be beneficial for even shorter wires, and thus replacing the short copper wires. Eventually it (can) leads to optical on-board communication and even an optical NOC. The metallic interconnects are being replaced because they are costly in terms of power, latency, bandwidth, and (silicon) area. Subsequently, communication is becoming a bottleneck in modern systems [18, 82], such as server-to-server communication, rack-to-rack, chip-to-chip, and NOC communication [82, 83]. Companies, such as IBM, ST microelectronics, LETI [82–84], and various universities are doing research into photonic NOC to overcome the metallic NOC communication problems. The concept of photonic on-chip interconnects was first introduced by Goodman et al. in 1984 [18, 85]. However, until a few years ago photonic NOCs were practically inconceivable, but photonic elements have recently become available as library cells in standard CMOS processes [18]. Thus, it is now practical and realistic to consider a 2D photonic NOC. However, sharp bends are not desirable in the wave guides, the heterogeneous property of 3D integration allows the separation of photonic and electrical planes, see Figure G.3(b).

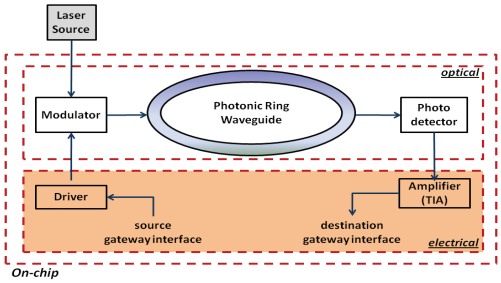
The volume of data being transferred between on-chip functional blocks is rising fast due to the need for higher resolutions (sound and video) and the need for faster computations. Integrated photonic Network-On-Chip (NOC) can overcome the (predicted) interconnect limitations for these demands, as identified by the ITRS roadmap [82]. Photonic interconnects offers (theoretically) ultra-high communication bandwidths in the terabits per second range [18]. Photonic signaling also has low power consumption, since the power consumption of optically transmitted signals at the chip level is independent of the distance covered by the photonic signal.

G.2.1 Photonic architecture building blocks

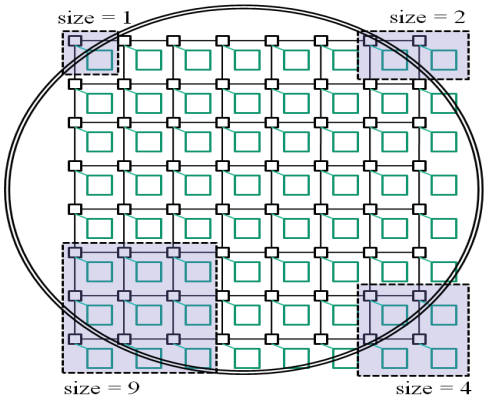
The basic building blocks are illustrated in Figure G.3(b) and G.4. There are five main components in a photonic network:(1) a laser source, (2) an opto-electric modulator (transmitter), (3) a photonic waveguide, (4) a Photonic Switching Element (PSE), and (5) a photonic detector (receiver). The electrical network routes data to the photonic gateway (see Figure G.3(b)). This gateway interface consists of driver buffers, Trans-Impedance Amplifier (TIA), circuitry for clock synchronization and recovery, and as well serialization and de-serialization circuits. The



(a) A 3D implementation of a hybrid photonic NOC and waveguide, located on the upper layer. The bottom layer contains the logic and memory



(b) High level overview of 3D photonic transmission architecture. Note, the laser source is off-chip.



(c) Four different region sizes of influence.

Figure G.3: 3D photonic on-chip network. [18]

modulator receives the data from the gateway, and converts the electrical signals into photonic signals(E/O), which are send through the photonic waveguide. When the photo detector detects the photonic signal (light) then it converts the photonic signal back into an electrical signal at the receiver (O/E). That signal is amplified at the TIA and the data leaves the gateway via electrical NOC.

G.2.1.1 Off-chip laser

There are still significant challenges in efficiently integrating a silicon-based laser on-chip. Using an off-chip laser can actually be beneficial because it leads to lower on-chip area and power consumption. Furthermore, when the inter-chip or the board-to-board communications use the same laser source then the total returns of a special off-chip laser is becoming beneficial, in terms of power consumption and area. It is because the power consumed by the optical waveguide is almost independent of the interconnect length and thus one or multiple chips consumes similar power, induced with the a single light source (area). In the remainder of this chapter it is assumed that off-chip laser is the light source for the optical NOCs in the chips. Optical communication rate can reach up to 12.5Gb/s [18].

G.2.1.2 The modulator and the photo detector

In order to transmit data from the cores through the photonic ring, electronic to optical (E/O) conversion is needed at the side of the sender. This conversion is done in the modulator, shown in Figure G.3(b). When the optical signal is received by the photo detector a reverse optical to electrical (O/E) conversion is performed at the receiver. The TIA amplifies the small analog signal to a digital signal.

G.2.1.3 Waveguide

The network topology of the waveguide can be a ring, as depicted in Figure G.3(b), or (for example) a mesh or torus. Without the use of optical switches, a photonic wave guide with highly angled structures, such as commonly found in electrical topologies, may result in significant signal degradation. Consequently, a ring like structure is much simpler and thus suites better to the physical characteristics of photonic waveguides, according to [18]. However, with a PSE an angled network structures can be achieved [19]. A PSE is based on a microring-resonator structure, and it is illustrated in Figure G.4. The switch is essentially an intersection of two waveguides with two ring resonators alongside of it. The ring resonators have a certain resonance frequency, which is derived from material and structural properties. In the OFF state, the ring resonance frequency is different than the wavelength of the carrier (light). Consequently, the light passes uninterrupted through the waveguide intersection as if it is a passive waveguide crossover (see Figure G.4(a)). Conversely, the switch is turned ON by injecting electrical current into the p-n contacts that is surrounding the rings. This changes the resonance of the rings such that the transmitted light is coupled into the rings, and thus it makes a right / left angle turn (see Figure G.4(b)). When the switches are OFF, they act as passive devices and consume nearly no power. In the ON position they consume less than 0.5mW [19].

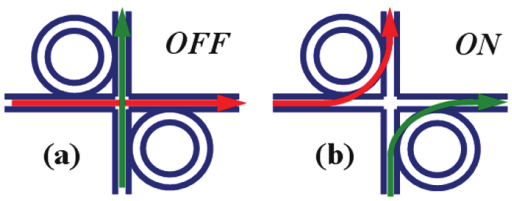


Figure G.4: A Photonic Switching Element (PSE). (a) In the off position. (b) In the on position. [19]

G.2.2 Photonic network on a separate layer

This section shows that the 3D integration enables the separation of photonic and electrical planes, which is beneficial to limit (sharp) angles in the waveguide and the use of PSE.

3D integration allows the separation of photonic, and memory and logic planes, as illustrated in Figure G.3(a). In the figure, the bottom plane consists of a microprocessor and memory cores, while the top plane consists of a photonic waveguide. Naturally, the lower layer could be separated in multiple layers, such as a separated logic or memory layer. TSVs are needed to

connect these planes, which includes the photonic plane. The benefit of a separated 3D photonic plane is because it is preferred to limit (sharp) angles in the waveguide and the use of PSE [18]. In particular, an electric network is needed to setup the optical path (PSE) before the (burst of) data can flow over a particular optical network, such as an optical mesh or torus [19]. This setup approach is similar to circuit switching, a technique for long-lasting connections. The necessity of setting up a path is because real-time header processing is relatively difficult to implement for photonic NOCs. The photonic signals are just traveling too fast, and optical buffering is very difficult [18, 19]. Therefore, the data is sent without buffering to the destination. Thus, the two main drawbacks of photonic NOCs are buffering and header processing, and that is why a hybrid structure between photonic and electrical NOCs is needed.

Despite the drawbacks, photonic NOC also has advantages. A package can quickly travel to the far end of the (same) chip via the photonic waveguide. Thus, the photonic wave guide acts as a global communication channel. Furthermore, photonic paths rely on a property of the photonic medium known as bit-rate transparency. Bit-rate transparency is property that photonic switches switch on and off once per message and not per bit, and thus the energy dissipation does not depend on the bit rate. Conversely, CMOS based routers must switch with every bit of transmitted data, and that leads to a dynamic power dissipation that scales quadratically with the bit rate. Bit-rate transparency provides a very high bandwidth while avoiding the power cost, typically associated with the traditional electronic networks. Furthermore, one waveguide can guide multiple carrier signals via Wavelength-Division Multiplexing (WDM) by using different wavelengths (colors) of laser light [86]. This technique is similar to the frequency division multiplexing, where one signal contains one or more frequencies. Furthermore, Optical Time Division Multiplexing (OTDM) can be used to share one signal (frequency) between multiple senders, where every sender gets a different time slot. Another advantage comes from the low energy loss in optical waveguides. The power dissipated on a photonic link is independent of the transmission distance. Thus, the energy dissipation remains essentially the same whether a message travels between two cores that are 2mm or 2cm apart [19]. Furthermore, photonic NOC communication enables seamless integration of (future) optical interconnects for off-chip communications.

G.2.2.1 3D photonic waveguide ring example

A separate photonic plane is used with a waveguide ring topology on top of a 8x8 mesh (see Figure G.3(c) and G.3(b)). The waveguide ring eliminates the need for switching elements, and thus it eliminates the circuit setup phase. The waveguide supports WDM with 8 different wavelengths. In addition a 12x OTDM is used to achieve higher transmission capacities [18]. Furthermore, the optical signal should be prevented from circulating more than one time over the waveguide ring, which is left as future work by [18]. The upper optical network and lower electrical network are running simultaneously, and they can be seen as a hybrid form between circuit switching and package switching [18], respectively.

At the lower layer, the logic and memory cores are located. These cores are connected via an electric mesh, using wormhole switching with XY routing and ACK/NACK flow control. Some electrical routers have besides the usual north, east, south and west ports also an extra port towards the photonic plane. The routers that lay in the region of influence have this extra port, and they are modified to consider the photonic path for global communication, as seen in

Figure G.3(c). There are four restricted regions of influences where the size of the regions is varied from one till nine routers. Although Figure G.3(c) depicts various region sizes in the same figure, this is only for illustrative purposes. The same region sizes per design are used. There are four gateways, one in each corner at the photonic layer. Each region has one gateway that contains a modulator and a photo detector. At the gateway, data from the electrical network is buffered and serialized. Thereafter, the data is sent at once over the waveguide towards the receiver. However, the light remains on the waveguide ring, and thus it should be removed. This removal mechanism is left as future work by [18]. At the gateway of the receiver the optical signal is converted back to an electrical signal, and then it is buffered. Subsequently, it is routed over the electrical network towards the end destination. The routers located in the region are reached within one hop.

G.2.3 Simulation methodology

The 3D hybrid photonic and electric NOC are compared to a 2D all-electrical mesh NOC. This is done for 64 cores (8x8) and 100 cores (10x10) with a uniform random packet injection rate of 0.35. Both electric meshes use wormhole switching with XY routing and ACK/NACK flow control. The clock frequency for the photonic ring and the communication network is set at 2GHz. The consumed power can be divided in two parts: (1) the power consumed in the electrical network, and (2) the power consumed in the optical ring. The distinction between the static and dynamic power consumption of electrical routers is taken into account by [18]. The average power consumption of each optical modulator (transmitter) and photo detector (receiver) is assumed to be 18.4 mW and 0.3 mW, respectively. Furthermore, the power consumed by the optical waveguide is almost independent of interconnect length, and since the length is relatively short the power loss in the waveguide is negligible [18].

Photonic waveguides provide faster signal propagation compared to electrical interconnects because they do not suffer from RLC impedances. However, the electrical signals must be converted into light and then back into an electrical signal. This process requires time and power overhead, which is taken into account in the analysis. The simulation uses 65nm process technology, and assumes a 400 mm2 die area. A high level floor planner of [87] is used to determine core placement and link lengths.

G.2.4 Simulation results

Figure G.5 shows the results of the power, throughput and latency comparison between the 3D hybrid photonic NOC (waveguide ring) and a traditional 2D all-electrical mesh NOC architecture. The hybrid photonic NOC architecture is configured with a serialization degree of four, meaning that four (parallel) carriers are used. Serialization degrees of higher than four are not practical because it increases the photonic modulator and photonic detector complexity and thus cost [18]. Furthermore, serialization degrees of lower than four results in higher power consumption, but a lower throughput. It uses more power due to the serialization circuit (shifter). The region of optical influence is set to nine, because it significantly improves the performance. With this configuration, the waveguide ring architecture provides up to a 13x power reduction, an improvement of 1.9x for throughput and a reduction by 1.55x for latency compared to an all-electrical mesh NOC. Ultimately, these results indicate that this architecture provides a high

performance per watt, as well a moderate throughput and latency improvement. However, [18] did not take into account the off-chip laser source, and thus this result might give a false picture. Other results with an on-chip laser source are unknown to the author of this thesis. Furthermore, the author of this thesis can imagine that other 2D optical schemes , such as [88], can be used but implemented via 3D integration. The photonic network components of the 2D schemes are resided on a separated 3D layer and then higher performance improvements (57%) can be expected, due to the uses of a different scheme.

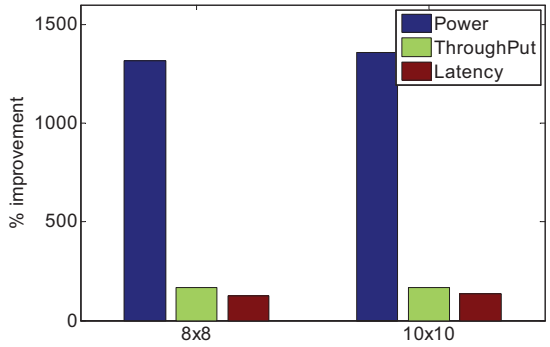


Figure G.5: Overall improvement of the hybrid photonic NOC, in respect to a 8x8 and 10x10 2D mesh. Four parallel carriers are used and the region of influence is nine. [18]

Article overview tables

H

The following tables give an overview of the articles that are discussed in this thesis. Furthermore, other interesting articles are also placed the table. The articles are divided over four tables, since there are four 3D stacking strategies:(1) core stacking, (2) FUB repartitioning, (3) logic gate splitting and (4) transistor repartitioning. Every table uses a different stacking strategy. The hyphen '-' denotes that there are no articles found about this topic, and the cross 'X' denotes that the cell (topic) is already covered in the table.

Table H.1: Core level articles sorted.

Core level	Memory	Logic	NOC
Memory	Multiport memory [89]	X	X
Memory	Stacking of Data banks [50]	X	X
Logic	Cache-on-processor [9], SRAM (L2) and DRAM (L3) cache on Intel core 2 Duo [8, p.46] [11, p.5] L2 and L3 cache on an Alpha21264 [47] Intel P4-data cache on functional unit [8, p.42] [90]	-	X
NOC	SMAFTI- with Feed-Through Interposer [91]	SMAFTI- with Feed-Through Interposer [91], CPU near of the pillar [59]	2D routers-on-routers (hop-by-hop) [13, p.2] 3D hybrid photonic NOC [18]

Table H.2: FUB repartitioning articles sorted.

Functional Unit Block (FUB)	Memory	Logic	NOC
Memory	Less TSV-single-layer data access [16]	X	X
	Bank stacking [9]		
	multi-channel memories [64]		
Logic	-	(floating point register file (RF) & single instruction multiple data (SIMD) unit) on top of the floating point (FP) unit, at an intel P4 proc [11]	X
		Alpha 21364 based proc [17] [8, p.44]	
NOC	-	-	3D Dimde router (seperated switch bar) [13, p.6]
			2D router connected via TSV&MOSFET [13, p.5]

Table H.3: Logic gate splitting articles sorted.

Logic gate splitting	Memory	Logic	NOC
Memory	Array splitting [9] [17, p.82]	X	X
	Less TSV-multi-layer data access [16]		
Logic	3D memory next to 3D proc [92]	3D Alpha 21364 based proc [17, p.86]	X
		Brent-Kung adder, Sklansky adder, Barrel shifter [51]	
		Log shifter,Kogge-stone adder, Instruction scheduler [10]	
		3D proc (Intel 2 architecture). [92]	
NOC	-	-	MIRA-multi-layered 3D router [93]

Table H.4: Transistor repartitioning articles sorted.

transistor repartitioning	Memory	Logic	NOC
Memory	Port splitting [23]	X	X
Logic	-	-	X
NOC	-	-	-

Detail article overview

I

The following tables presents the same articles of appendix H, but a more detail overview is given, compared to appendix H. In particular, the simulation tools which are used by the articles are listed.

The articles are divided over four tables, since there are four strategies, which are core stacking , FUB repartitioning, logic gate splitting and transistor repartitioning. The column 'stack method' indicates which layers are stacked, where M denotes memory, the L denotes logic and the N denotes network. Thus, the MM is the memory-on-memory stack, the LM is the logic-on-memory stack, and the NL is the network-on-logic stack. The hyphen '-' denotes that there are no articles found about this topic, and the cross 'X' denotes that the cell (topic) is already covered in the table.

In general, a circuit simulator tool is used to determine the new frequency and a second simulator is used to determine the overall impact on a Chip Multi-Processor (CMP). *Hspice and Spice are commonly used as circuit simulators. The SimpleScalar is commonly used to evaluate the overall performance in a system. Furthermore, the (only) temperature simulator used by the articles is Hotspot.*

Table I.1: The first part of the detail core stacking overview.

Stack method	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps)	Simulator	Comments	Year	Company
MM	[89]	2	Multiport memory	-	TSV+ micro bumps	no information	NMOS gate length= 1,5μm	2000	Tohoku university, Japan
	[50]		Stacking of data banks	65		Yes, not mentioned	Characterized in a 65nm technology	2005	Intel
LM	[9]	5	Cache-on-processor	70	Micro bumps	SPICE, SimpleScalar 4.0	F2F- 2 layer	2005	Georgia Institute of Technology
	[8, 11]	- 2	SRAM and DRAM cache on Intel core 2 Duo	-		Mem-logic->trace driven memory hierarchy sim. logic-on-logic-> microarchitecture performance sim.	Internally developed research tool	2006	Intel
	[47]	2	L2 & L3 cache on Alpha21264	L2+L1+ proc=130 on-chip Main mem=150	TSV+ micro bumps	sim-alpha, SPEC2000	Three mem applications from SPEC2000	2006	University of California
	[8]	42	Intel P4-data cache on functional unit	-		Yes, not mentioned	-	2007	Georgia Institute of Technology Pennsylvania State University Intel
	[90]		Intel P4-data cache on functional unit	-		Internal Intel developed cycle accurate model	-	2007	Intel

Table I.2: The second part of the detail core stacking overview.

Stack method	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps)	Simulator	Comments	Year	Company
LL	[11]	3	(floating point register file (RF) & single instruction multiple data (SIMD) unit) on top of the floating point (FP) unit,at an Intel P4 proc	-		internal Intel developed performance simulator	-	2006	Intel
NM	[91]	821, 826	SMAFTII- with Feed-Through Interposer	n/a	TSV+ micro bumps	NO, pitch of 50μm, wires of 10μm wide	SMAFTI Is an interposer	2007	NEC electronics, Oki Electronic industry, Elpida
NL	[91]		SMAFTII- with Feed-Through Interposer	n/a	TSV+ micro bumps	NO, pitch of 50μm, wires of 10μm wide	SMAFTI Is an interposer	2007	NEC electronics, Oki Electronic industry, Elpida
	[59]		CPU near of the pillar	90	n/a (pillars)	Verilog HDL with TDMC libraries	-		
NN	[13]	1	2D routers-on-routers (hop-by-hop) (Dimde)	90		Stand-alone cycle-accurate 3D NoC simulator, a hybrid 3D NoC/cache simulation running commercial and scientific benchmarks	-	2007	Intel, Pennsylvania state university
	[18]		3D hybrid photonic NOC	65		Latency via calculations, Orion simulator for power	-	2009	Colorado State University

Table I.3: First part of the detail overview of FUB repartitioning.

Stack method	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps	Simulator	Comments	Year	Company
MM	[9]		Bank stacking	70	Micro bumps	SPICE, SimpleScalar 4.0	F2F- 2 layer	2005	Georgia Institute of Technology
	[16]	2	Less TSV-single-layer data access	65	TSV	CACTI 5		2009	XiŠan Jiaotong University, Rensselaer Polytechnic Institute,
	[64]	2	Multi-channel memories		bus	SystemC electronic system level (ESL) environment	Multi-channel also applicable for 3D	2009	Nokia Research Center
LM	X	X	X	X	X	X	X	X	X

Table I.4: Second part of the detail overview of FUB repartitioning.

Stack method	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps)	Simulator	Comments	Year	Company
LL	[17]	89	Alpha 21364 based proc	70	TSV	Hspice	F2F TSV 2μm, B2B TSV 4μm,	2006	IBM, Pennsylvania state university, Georgia Institute of Technology
	[8]	44	Alpha 21364 based proc	-	TSV	Yes, not mentioned	-	2007	Georgia Institute of Technology Pennsylvania State University Intel
	[11]		(floating point register file (RF) & single instruction multiple data (SIMD) unit) on top of the floating point (FP) unit,at an Intel P4 proc	-	-	Internal Intel developed performance simulator	-	2006	Intel
NM	X	X	X	X	X	X	X	X	X
NL	X	X	X	X	X	X	X	X	X
NN	[13]	5	2D router connected via TSV&MOSFET	90	-	stand-alone cycle-accurate 3D NoC simulator, a hybrid 3D NoC/cache simulation running commercial and scientific benchmarks,	-	2007	Intel, Pennsylvania state university
	[13]	5	3D Dimde router (seperated switch bar)	90	-	stand-alone cycle-accurate 3D NoC simulator, a hybrid 3D NoC/cache simulation running commercial and scientific benchmarks,	-	2007	Intel, Pennsylvania state university

Table I.5: First part of the detail overview of logic gate splitting.

	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps	Simulator	Comments	Year	Company
MM	[9]	5	Array splitting	70	Micro bumps	SPICE, SimpleScalar 4.0	F2F- 2 layer	2005	Georgia Institute of Technology
	[17]	82	Array splitting	-			-	2006	IBM, Pennsylvania State University, Georgia Institute of Technology
	[16]	2	Less TSV-multi-layer data access	65	TSV	CACTI 5	-	2009	XiŠan Jiaotong University, Rensselaer Polytechnic Institute,
LM	[92]	6	3D memory next to 3D proc	65	-	SimpleScalar/MASE, HotSpot 3.0.2	no hetrogenious integration possible	2007	Georgia Institute of Technology

Table I.6: Second part of the detail overview of logic gate splitting.

	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps	Simulator	Comments	Year	Company
LL	[10]		Kogge-stone adder+ Log shifter+ Instruction scheduler	70	TSV	3D Magic (MIT), HSPICE, Simplescalar	-	2007	Pennsylvania Sate University
	[51]	3	Kogge-stone adder+ Brent-Kung adder+ Sklansky adder+ Barrel shifter	70	-	HSPICE	-	2006	Georgia Institute of Technology
	[58]	2	3D Alpha 21364 processor	70	-	Spice, HotSpot 3.0	Only models power and thermal	2006	Georgia Institute of Technology
	[17]	89	3D Alpha 21364 processor	70	-	SimpleScalar / MASE, HSpice	-	2006	IBM, Pennsylvania State University, Georgia Institute of Technology
	[92]	6	3D proc (Intel 2 architecture)	65	-	SimpleScalar / MASE, HotSpot 3.0.2	-	2007	Georgia Institute of Technology
NM	X	X	X	X	X	X	X	X	X
NL	X	X	X	X	X	X	X	X	X
NN	[93]		MIRA-multi-layered 3D router	90	-	Cycle-accurate NoC simulator	-	2008	Pennsylvania State University

Table I.7: A detail overview of transistor repartitioning.

	Bibtex	Page	Proposed topic	Technology (nm)	Interconnects (TSV or micro bumps	Simulator	Comments	Year	Company
MM	[23]	22	Port splitting	65		Hspice	-	2007	Georgia Institute of Technology
LM	X	X	X	X	X	X	X	X	X
LL	X	X	X	X	X	X	X	X	X
NM	X	X	X	X	X	X	X	X	X
NL	X	X	X	X	X	X	X	X	X
NN	X	X	X	X	X	X	X	X	X