Yield and Cost Analysis for 3D Stacked ICs

Mottaqiallah Taouil

Yield and Cost Analysis for 3D Stacked ICs

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College van Promoties,
in het openbaar te verdedigen op
vrijdag 5 september 2014, om 12:30 uur

door

Mottaqiallah TAOUIL
Master of Science in Computer Engineering
geboren te Al Hoceima, Marokko

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. K.L.M. Bertels

Copromotor:
Dr. ir. S. Hamdioui

Samenstelling van de promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. K.L.M. Bertels	Technische Universiteit Delft, promotor
Dr. ir. S. Hamdioui	Technische Universiteit Delft, copromotor
Prof. dr. E. Charbon	Technische Universiteit Delft, The Netherlands
Prof. dr. K. Chakrabarty	Duke University, USA
Ir. E.J. Marinissen PDEng	IMEC, Belgium
Dr. ir. H.G. Kerkhoff	University of Twente, the Netherlands
Prof. dr. J. Pineda de Gyvez	Technische Universiteit Eindhoven, the Netherlands
Prof. dr. ir. H.J. Sips	Technische Universiteit Delft, reserve lid

This work has been supported by 3DIM^3 via grants to Delft University of Technology.

ISBN 978-94-6186-331-7

Published and distributed by: Mottaqiallah Taouil Email: mo_taouil@hotmail.com

Subject headings: 3D stacked ICs, cost analysis, yield analysis, redundancy, fault coverage, test cost, test analysis, fault coverage, interconnect testing, and interconnect diagnoses

Printed in the Netherlands

Dedicated to my parents and my wife
for all the support they have been giving me.

Summary

3D stacking is an emerging technology promising many benefits such as low latency between stacked dies, reduced power consumption, high bandwidth communication, improved form factor and package volume density, heterogeneous integration, and low-cost manufacturing. However, it requires modification of existing methods and/or introduction of new ones with respect to design, manufacturing, and testing in order to facilitate production. In this thesis three challenges are addressed: one related to manufacturing (i.e., yield improvement) and two related to testing (i.e., cost modeling and interconnect testing).

Yield improvement - We propose two yield improvement schemes applicable for 3D Stacked-ICs (3D-SICs) with similar die sizes (such as memories and FPGAs): wafer matching and layer redundancy. Wafer matching is based on algorithms that select wafers with identical or similar fault maps for stacking to boost the compound yield. Our algorithms outperform yield-wise previously proposed schemes, and more importantly reduce memory and time complexity significantly. On the other hand, redundancy in 3D memories makes use not only of conventional spare rows and columns, but also of the third dimension to access either spare dies (layer redundancy) or spare cells (inter-layer redundancy). Layer redundancy showed to be effective from a yield point of view, but may seriously affect die area and cost. Inter-layer redundancy realizes even higher yield improvements; however, it requires through-silicon vias (TSVs) to scale down with one order of magnitude for area-efficient implementations.

Cost Modeling - Selecting an appropriate and efficient test flow for 2.5D/3D SICs is crucial for overall cost optimization. In addition, diverse products and applications require different quality levels resulting in different test flows; these flows may require different design-for-test (DfT) features, which need to be incorporated in the various dies during an early design stage. Therefore, an appropriate cost model used to evaluate test flows with their associated DfT, while taking into account yields and die production costs, is of great importance. A proper cost modeling tool for 2.5D/3D stacked ICs is developed; the tool is referred to as 3D-COSTAR. It considers all costs involved in the whole production chain, including design, manufacturing, test, packaging, and logistics, e.g., related to shipping wafers between a foundry and a test house. 3D-COSTAR provides the estimated overall cost for 2.5D/3D-SICs and its cost breakdown for a given input parameter set, such as test flows, die yield, stack yield etc. The crucial importance of 3D-COSTAR is demonstrated by analyzing trade-offs of different complex optimization test problems such as (a) the impact of test coverage of the pre-bond silicon interposer test, (b) the impact of pre-bond testing of active dies using either dedicated probe-pads or micro-bumps, (c) the impact of mid-bond testing and logistics, and (d) the impact of different test flows on the test escapes.

Interconnect Testing - A potential application of 3D-SICs is stacking of memory on logic. However, testing the TSV interconnects between such dies is challenging, as the memory and the logic die typically come from different manufacturers. Currently, proposed solutions fail to address dynamic and time-critical faults. In addition, memory vendors have in the past not been in favor to put additional DfT structures such as IEEE 1149.1 for interconnect testing on their memory devices. We propose a new Memory-Based Interconnect Test (MBIT) approach for 3D stacked memories. Our test patterns are applied by using read and write instructions to the memory and are validated by a case study where a 3D memory is assumed to be stacked on a MIPS64 processor. The main benefits of the MBIT approach include zero area overhead, detection of both static and dynamic faults, at-speed testing, flexibility, extremely short test time, and interconnect fault diagnosis.

Samenvatting

Het 3D stapelen van IC's is een opkomende technologie die vele voordelen met zich mee brengt, zoals een lage latency tussen gestapelde chips, gereduceerde energieverbruik, hoge communicatie bandbreedte, verbeterde form factor en package volume dichtheid, heterogene integratie en lage productiekosten. Echter vereist dit het wijzigingen van bestaande methoden en/of de invoering van nieuwe methoden met betrekking tot het ontwerp, de fabricage en het testen om de commerciële productie te vergemakkelijken. In dit proefschrift worden drie uitdagingen geadresseerd: één ervan verwant aan productie (d.w.z. yield verbetering) en twee verwant aan testen (d.w.z. kostenmodellering en het testen van interconnects).

Yield verbetering - Wij stellen twee yield verbeteringsschema's voor die van toepassing zijn op 3D-Stacked IC's (3D-SIC's) met soortgelijke chip oppervlakte (zoals geheugens en FPGAs). Dit zijn wafer matching en layer redundancy. Wafer matching is gebaseerd op algoritmes waarin wafers met identieke of soortgelijke chip defect locaties geselecteerd worden voor het stapelen, dit om de yield te boosten. Onze algoritmes presteren yieldsgewijs beter dan vorige voorgestelde schema's, maar belangrijker nog is de significante reductie in geheugen- en tijds-complexiteit. Anderzijds, redundantie in 3D gestapelde geheugens maakt niet alleen gebruik van de conventionele reserve rijen en kolommen, maar ook van de derde dimensie om ofwel gebruik te maken van reserve chips (layer redundantie) of reserve cellen (inter-layer redundancy). Layer redundancy is vanuit een yield oogpunt effectief, maar kan ernstige gevolgen hebben voor de chip oppervlakte en kosten. Inter-layer redundantie realiseert zelfs een nog hoger rendement; hoewel, dit vereist het neerschalen van through-silicon vias (TSVs) met een orde van grootte voor een area-efficiënte implementatie.

Kostenmodellering - Het selecteren van een geschikte en efficiënte test flows voor een 2.5D/3D-SIC is cruciaal voor totale kostenoptimalisatie. Daarnaast vereisen diverse producten en toepassingen verschillende kwaliteitsniveaus wat resulteert in verschillende test flows; deze flows kunnen verschillende design-for-test (DfT) functies vereisen, die in diverse chips moeten worden toegevoegd gedurende de ontwerpfase. Daarom is een geschikte kostenmodel die gebruikt wordt om test flows met bijbehorende DfT te evalueren, rekening houdend met yield en productiekosten, van groot belang. Een goede kostenmodelleringstool voor 2.5D/3D SIC's is ontwikkeld; de tool wordt aangeduid als 3D-COSTAR. Het beschouwt de kosten van de gehele productieketen, inclusief ontwerp, productie, testen, packaging en logistiek, die bijvoorbeeld betrekking heeft tot de transport van wafers tussen een foundry en een test house. 3D-COSTAR biedt de totale geraamde kosten voor 2.5D/3D-SIC's en de kostenverdeling voor een bepaalde input parameter set, zoals testflows, chip yield, etc. Het cruciale belang van 3D-COSTAR is aangetoond door het analyseren van trade-offs van verschillende complexe testoptimalisatie problemen, zoals (a) de impact van de pre-bond

silicium interposer test, (b) de impact van het pre-bond testen van actieve chips door middel van of probe-pads of micro-bumps, (c) de impact van het mid-bond testen en logistieke kosten en (d) de impact van de verschillende test flows op de test escapes.

Testen van interconnects - Een mogelijke toepassing van 3D-SICs is het stapelen van een geheugen op logica. Echter is het testen van de TSV interconnects tussen dergelijke chips moeilijk, omdat de geheugens en logica chips meestal van verschillende fabrikanten komen. Momenteel voldoen de voorgestelde oplossingen voldoen niet aan dynamische en tijdkritieke fouten. Bovendien gaven geheugen leveranciers in het verleden geen voorkeur aan het plaatsen van extra DFT structuren zoals JTAG op hun geheugenapparaten om interconnects te testen. Wij stellen een nieuwe memory Based Interconnect Test (MBIT) aanpak voor 3D gestapelde geheugens. De testpatronen worden uitgevoerd door lees en schrijf instructies naar het geheugen en zijn gevalideerd door een casus waarin verondersteld wordt dat een 3D geheugen wordt gestapeld op een MIPS64 processor. De belangrijkste voordelen van de voorgestelde MBIT aanpak zijn geen extra oppervlakte, detectie van zowel statische als dynamische fouten, testen op normale chip snelheid, flexibiliteit, extreem korte testtijd en de mogelijkheid om interconnect foutdiagnose toe te passen.

Acknowledgements

After a period of over four years, I can finally say that my Ph.D. dissertation has ended successfully. It has been a unique experience with many ups and downs. Luckily, there has been a great stimulating environment around me (both at home and at work) that facilitated me carrying out this work. I would like to dedicate my acknowledgments to everyone that was part of this environment.

First of all, I would like to thank my co-promotor and daily supervisor Assoc. Prof. dr. ir. S. Hamdioui for providing me the opportunity to pursue my Ph.D. thesis under his guidance. Not did I only learn how to do research and write scientific papers, but also what it means to be a dedicated researcher. Thank you for the continuous motivation and the proper guidance during this work. In addition to properly educating me as an independent researcher, you gave me many opportunities to develop myself further. For instance, the development of lab-courses, giving lectures, organizing IEEE conferences and participating in European Project proposals. Thank you for being my co-promotor! Prof. dr. ir. K.L.M. Bertels, you were more than only a promotor. As main professor and head of our CE laboratory I am very thankful for your efforts in creating a nice atmosphere and working environment in the group. You have always been encouraging us to attend social events, such as Karting, bowling, world cup matches, etc. Further, you always motivated us to maintain a strong network in the group (for example through brainstorm sessions), and not to forget to always provide us with cookies in the coffee room. I would also like to thank the remaining committee members for accepting their role, reading this dissertation, and providing feedback; thank you for all your efforts. Furthermore, I would like to mention several members specifically by name. Ir. E.J. Marinissen PDEng, co-author of many joint publications, I thank you not only for providing constructive feedback for this thesis, but also for the many discussions we had over the years, and not to forget the numerous paper corrections. Your invested time is highly appreciated! Prof. dr. K. Chakrabarty, as part of the same research community I still remember several of our constructive discussions during various conferences regarding 3D, biochips, etc. I really enjoyed them. I am sure that you are very inspiring professor.

I would like to express my gratitude to the Computer Engineering (CE) secretariat and staff for taking care of all the bureaucratic matters related to my day-to-day work. Thank you for always being helpful and for providing the necessary support. Lidwina, thank you for managing all the forms and other secretary-related tasks. Erik and Eef, thank you for creating and keeping the websites updated, managing the servers, fixing computer problems, installing various software, etc.

George, Saleh, Mihai, dr. ir. Demid, Winston, assoc. prof. dr. ir. S. Hamdioui, and assoc. prof. dr. S. Cotofana, thank you for your contributions as member of the CE 3DIM^3 team. The brainstorm sessions have been very fruitful. You have been a great team to work with, always helpful, and open to discussions.

I would like to thank Marius for organizing the CE weekly football matches. These matches provided me very often the right motivation to continue the challenging tasks that laid in front of me. Everyone that participated in these matches, thank you! You are too many to be mentioned by name. Also, special thanks to Andrew, Catalin, Mihai, Mafalda, and Mahroo for organizing various other CE social events. Everyone that participated and contributed to the nice atmosphere, thank you as well. Mihai, in addition to this, I also thank you for designing this thesis cover in such a short time; the cover looks excellent.

I would like to extend my thanks to my previous and current office mates. Christos, Kazeem, Nor Zaidi, Seyab, Mafalda, Innocent, Cristi, Hector and Mahroo. Thank you for the many but sometimes controversial discussions. As some of you are atheists, Christians, Muslims, or ex-Muslims, sometimes the discussions became intense; nevertheless, they always have been fruitful and enjoyable. I would like to address some words regarding Nor Zaidi; he has been a big inspiration and motivation for us. The sad news reached us that he has passed away. إنا لله وإنا إليه راجعون. All the other CE colleagues, you are too many to be mentioned by name; I thank each of you individually for the pleasant working environment.

I would like to thank Imran, Seyab, Faisal, Fakhar, and Laiq for our close friendship on the campus and in particular for the enjoyable daily lunches we had. Although the food was often too spicy it was very delicious; I really miss these lunches and the dialogues during them. I would like to also thank all the other Islamic community members, especially the ones I met regularly during the prayers. Thank you for always keeping up the spirit high.

Last but not least, I would like to express my deepest thanks to my family for all the support they gave me. In particular my mother and wife! شكراً. My mother, there are no words that can fully express my gratitude towards you. My wife, you have been an excellent support for me and always a reliable source to trust upon. I hope you can guard this distinctive quality of yours for the rest of your life.

Mottaqiallah Taouil

Delft, September, 2014
the Netherlands

Contents

Chapter 1

Introduction

Transistor scaling slowly reaches physical device limits and goes hand-in-hand with issues pertaining to process variations, power consumption, reliability, yield, cost, etc. Some of these problems could be alleviated by utilizing 3D-Stacked ICs (3D-SICs). The popularity of 3D-SICs is rising among research institutes and industry. 3D-SICs are emerging as one of the main candidates to continue Moore's Law. In this chapter, we first introduce the evolution leading up to 3D-SIC technology. Subsequently, we present the opportunities such a technology offers and discuss its main challenges. Thereafter, we briefly describe the research directions of this dissertation followed by the main contributions. Finally, we provide the outline of the remainder of this dissertation.

1.1 Introduction to 3D Stacked ICs

The aim of this section is to get the reader acquainted with 3D Stacked-ICs. Section 1.1.1 describes past and future semiconductor trends. Section 1.1.2 gives a general classification of stacking transistors in the vertical dimension. Section 1.1.3 explains the crucial 3D-SIC manufacturing steps.

1.1.1 Past and Future Semiconductor Trends

In the past years, the semiconductor industry has fulfilled IC functionality demand by transistor down-scaling adhering to Moore's Law [1]. The associated benefits (such as higher transistor density, higher performance, and reduced cost) of this transistor miniaturization have consistently been emphasized in prior International Technology Roadmap for Semiconductors (ITRS) roadmaps [2]. Although this "More-Moore" trend is still predicted for several future technology nodes, new demands arise concerning computational diversity and functionality in systems that include analog sensors, bio-chips etc. [3]. This diversification is referred to as "More-than-Moore". Figure 1.1 shows a technology roadmap that illustrates the customer need of "More-Moore" and "More-than-Moore", thereby progressing towards more complex and diverse systems.

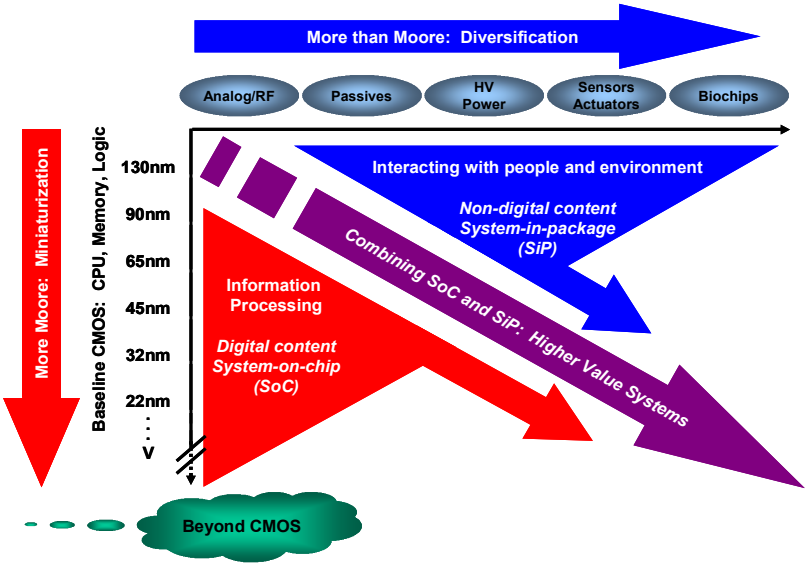


Figure 1.1: More-than-Moore [3].

The typical embodiment of such sophisticated and diverse systems predominantly was realized in the past by assembling multiple distinctive components on a printed circuit board (PCB) as depicted in Figure 1.2(a) [4]. Relatively long off-chip wires are used to deliver the communication between ICs, therefore, yielding to relatively low performance and

(a) Printed Circuit Board (PCB) (b) Multi-Chip Package (MCP)

Figure 1.2: Traditional packaging technologies [4].

power-hungry systems. To shrink the off-chip path length, several heterogeneous chips were brought in closer proximity by integrating them on a single Multi-Chip Package (MCP) as depicted in Figure 1.2(b). This reduces the form factor; however, it still requires off-chip communication between the dies through the package substrate. The need for more complex, faster, more power-efficient and diverse systems led to the utilization of the third dimension.

1.1.2 3D Technology Classification

The increasing demand for More-Moore and More-than-Moore has been mostly realized by transistor scaling. A technology that implies further increase in transistor density is the stacking in the vertical dimension; in addition, such technology likely also benefits from more computation diversity due to heterogeneous integration, better performance, and lower power dissipation, all at a smaller footprint. Figure 1.3 shows a general classification of 3D technology consisting of three main classes (i.e., 3D Packaging, 3D Die Stacking and 3D Monolithic) each described next.

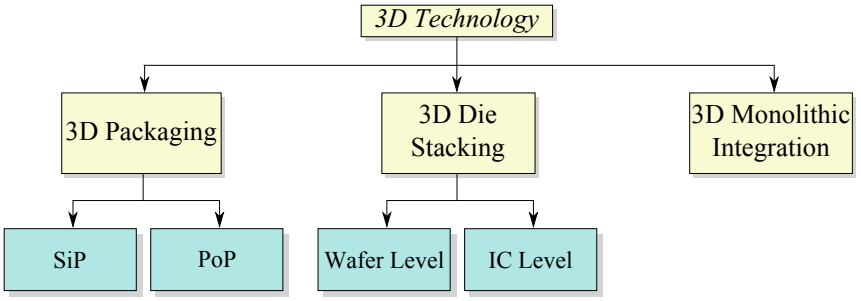
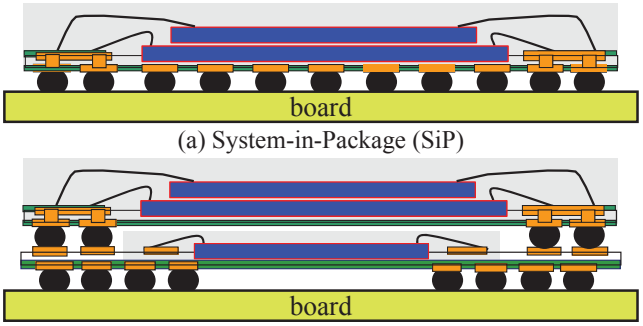


Figure 1.3: Classification of 3D systems.

3D Packaging

In *3D Packaging*, multiple dies are stacked vertically at the packaging level. Interconnects between the I/Os of the dies are typically formed by wire-bonding, flip-chip, or Ball-Grid-Array (BGA) stacking. This type of 3D stacking provides the lowest interconnect density. An example is a System-in-Package (SiP) depicted in Figure 1.4(a); here the system comprises several naked ICs stacked in the vertical dimension and is packaged in a single chip. The ICs are internally connected by fine wires that are bonded to the substrate. Benefits of



(a) System-in-Package (SiP)

(b) Package-on-Package (PoP)

Figure 1.4: 3D packaging technologies.

this system compared to PCBs and MCPs are manifold, such as reduced global wire length (leading to more performance), smaller footprint, increased transistor density, and the elimination of the need to package each die separately. SiPs are, due to these assets, widely used in mobile devices, music players, digital cameras, portable audio players, etc. [5].

Another example that fits into the 3D Packaging class is Package-On-Package (PoP) technology where multiple packaged chips are stacked vertically [8, 9]; an example is depicted in Figure 1.4(b). In this figure, the 3D-SiP package of Figure 1.4(a) is stacked on top of another package. SiPs and PoPs can take many forms as depicted by the examples in Figure 1.5. Figure 1.5(a) shows a SiP with multiple row bonding used to increase the interconnect bandwidth. Figure 1.5(b) shows a SiP delimited by spacers allowing the top dies to form interconnections to the substrate without reducing their die size. Figure 1.5(c) shows a

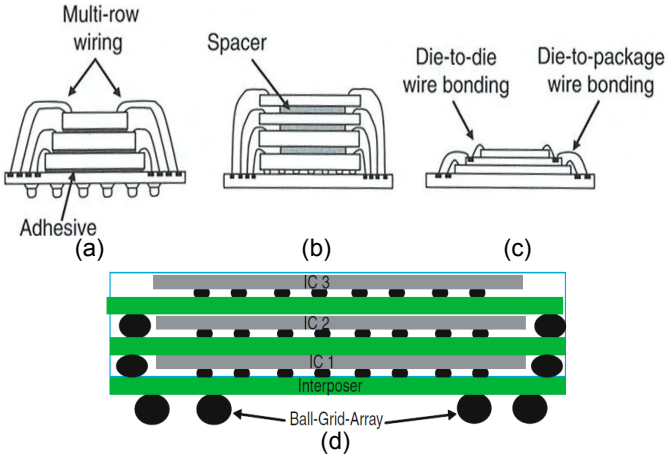


Figure 1.5: 3D Packaging: (**a**) Multiple row bonding [6], (**b**) Dies delimited by spacers [6], (**c**) Die-to-Die and Die-to-Package wire bonding [6], and (**d**) BGA-stack [7].

SiP that combines direct die-to-die bonding and die-to-package wire bonding. Figure 1.5(d) shows the interconnection of three dies through a BGA. Generally, communication between stacked ICs in the *3D Packaging* class is performed by off-chip communication by means of wire-bonding through the package substrate or through direct die-to-die communication. However, the wire-bonds in die-to-die communication do not go through the silicon die substrate.

3D Die Stacking

In *3D Die Stacking*, each separately manufactured tier can be stacked and bonded to another tier using a direct communication link between vertically adjacent tiers. A 3D-SIC consists of two or more dies stacked in the vertical direction. The interconnection between the dies can be implemented physically by micro-bumps and/or TSVs, or via contactless communication based on capacitive [10,11] or inductive coupling [12,13]. Among the interconnection schemes, TSVs are the most promising as contactless communication schemes face several challenges such as a stable power delivery [14].

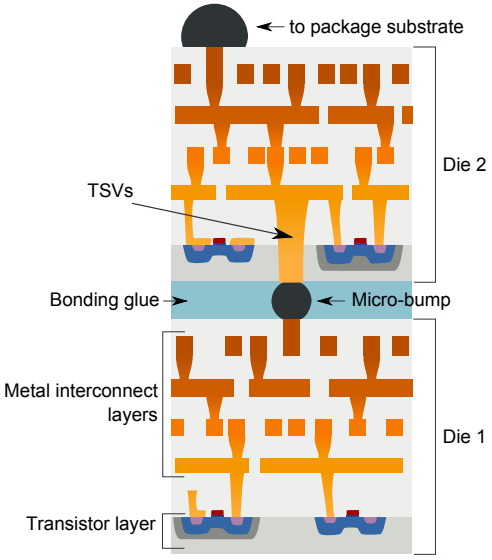


Figure 1.6: TSV-based 3D die stacking.

Figure 1.6 depicts a two-layer 3D-SIC with a face-to-back (F2B) stacking configuration. Compared to off-chip wire-bonds, TSVs enable extremely short connections as they go straight through the substrate of the dies. Between the stacked dies, micro-bumps are used to connect the TSVs from Die 2 to Die 1. TSV-based 3D-SICs can be used to empower *More-Moore* and *More-than-Moore* systems and have considerable advantages over planar ICs and SiPs, such as high-speed, less power consumption, small form factor, and heterogeneous integration [15–18]. A special class of *3D Die Stacking* are the 2.5D-Stacked ICs (2.5D-SICs)

in which two or more active dies are stacked side by side Face-to-Face (F2F) on a large passive silicon interposer. The interposer is only used to connect the active dies by means of TSVs and wires. 2.5D-SICs are in general easier to manufacture, but its advantages are typically also less than those of 3D-SICs (for example, power dissipation in interconnects, bandwidth, off-chip I/O density) [19].

3D Monolithic Integration

In *3D Monolithic Integration* active devices are created on-chip bottom-up in a single linear process flow; this process does not require bonding materials between the layers. The stacked active silicon areas are isolated from each other by dielectric layers. Among the stacking approaches, monolithic 3D integration provides the highest vertical interconnect density between stacked layers. Currently, the state of this technology is insufficiently enhanced to realize reliable high-performance 3D circuits, primarily due to its complex processing which leads to inferior quality of devices in the upper planes and limited number of layers due to thermal constraints [20].

In this dissertation, we focus on a particular subset of *3D Die Stacking* in which the vertical interconnects are realized by through-silicon vias (TSVs), the TSV-based 3D-SICs. The process to manufacture such ICs is described next.

1.1.3 Manufacturing

Recent enhancements in process development enabled the fabrication of TSV-based 3D-SICs [17]. Critical steps to manufacture such ICs are the formation of TSVs, and the bonding and thinning of dies. They are described next.

TSV Manufacturing

TSVs are holes that go through the silicon substrate filled with a conducting material (e.g., copper or tungsten). These holes are shaped by deep reactive ion etching (DRIE) [21] or laser ablation [17]. The size, pitch, conductivity, and conducting material of TSVs are heavily impacted by the stage they are constructed at [17]; either during the conventional manufacturing of planar ICs (via-first, via-middle, via-last) or during 3D processing steps (via-last, via-after-stacking) as depicted in Figure 1.7. Via-first TSVs are manufactured prior to the front-end of line (FEOL), i.e., before the transistors are fabricated, and must be filled with doped poly-silicon which has a relatively high resistance [17]. A lower-bound temperature constraint, dictated by the FEOL processing, excludes the usage of copper TSVs. Via-middle TSVs are manufactured between the FEOL and back-end of line (BEOL), i.e. before the metal layers are fabricated and typically utilize copper or tungsten as filling material. Via-Last TSVs are manufactured after BEOL either prior or post thinning and have the advantage over via-first and via-middle TSVs that foundries without TSV processing

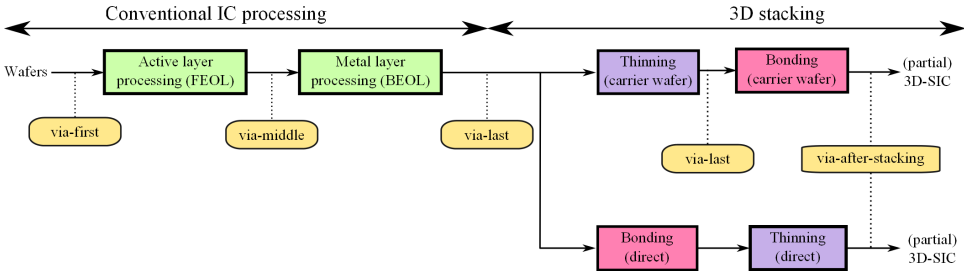


Figure 1.7: TSV manufacturing stages.

equipment already may manufacture the whole IC. Finally, the option exists to create TSVs (via-after-stacking) as the last 3D processing step.

Thinning

Thinning of wafers, performed by wet-etching, is required to expose the TSV tips to form electrical contacts; TSVs have limited aspect ratios and therefore, pose a major challenge on the filling processes [4]. A typical thinning process is described in [17]. First, a coarse-grind process removes inaccurately large portions of the back-side of the substrate. As a byproduct, surface and sub-surface damages are created up to a depth of approximately 10-20 μm. A fine-grind process is followed which minimizes surface and subsurface damages typically up to a depth of 2 um. In the last step, these damages are removed by a stress-relief step to avoid propagation of cracks during bonding and to increase the bonding strength by enlarging the contact area of two bonded dies.

Bonding

Bonding can be of temporary or permanent type [4]. In temporary bonding, dies are attached to a carrier wafer for TSV or thinning processes only. This type of bonding is used in several 3D process flows (see Figure 1.7) and is commonly realized by polymer adhesive or electrostatic bonding [17].

In permanent bonding, the (thinned) die is bonded permanently by using direct Cu-Cu, SiO_2/SiO_2, Au/Au, polymer adhesive, gel adhesive or eutectic bonding [17]. There are three permanent bonding methods, as depicted in Figure 1.8. They are Die-to-Die (D2D), Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) bonding [17]; each comes with its own merits. Although complex, a high alignment accuracy is feasible in D2D and D2W bonding at the cost of a low throughput. In addition, the handling of very small dies becomes impractical for these bonding methods. Nevertheless, a major benefit of the D2D and D2W bonding methods is the ability to apply pre-bond testing, which may prevent faulty dies from entering the stack [17] leading to improved compound yield. On the other hand, achieving a high alignment accuracy is simpler in W2W bonding, particularly if small dies are used.

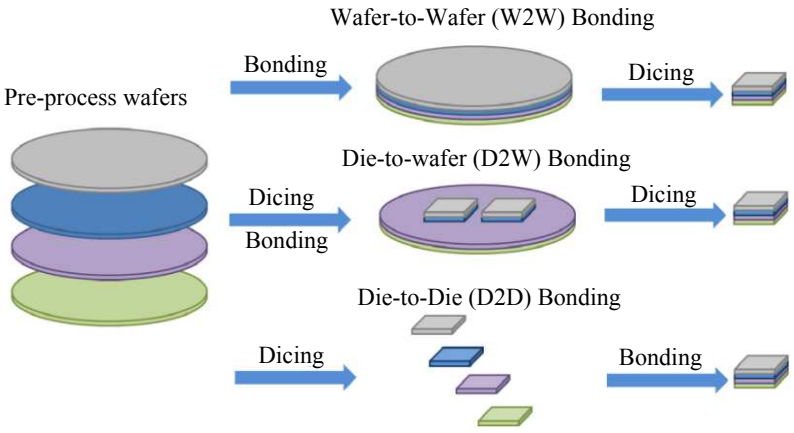


Figure 1.8: 3D bonding techniques [22].

However, W2W stacking negatively impacts the compound yield as the stacking of good dies on top of bad dies cannot be prevented. In addition, W2W bonding requires the stacking of dies with same sizes; this makes them suitable for limited applications such as memories and FPGAs. These applications have a high degree of regularity.

1.2 Opportunities and Challenges

The previous section introduced briefly 3D-SICs. In this section, the main drivers, advantages and disadvantages of 3D-SIC technology are described.

1.2.1 Opportunities and Drivers

The prospects and potential benefits that 3D-SICs offer is leading to an expansion of research work both in academia and industry [14, 17, 18, 23–26]. However, prior to be accepted as a solid and mature technology, each new technology must demonstrate its market and technological advantages such as the ones depicted in Figure 1.9. They are discussed next.

- **Cost:** A key condition to shift from the design and prototype phase to large-scale production is a manageable cost figure. 3D-SICs are able to reduce cost by splitting up large dies over multiple smaller layers. A benefit of this approach is that the compound yield of the 3D-SIC with smaller die sizes may exceed the yield of the single large die [27]. Another way to reduce cost in 3D-SICs is by integrating multiple stand-alone chips. For example, by stacking DRAM on logic more than a bandwidth improvement is realized. The physical size of vertically pilled-up dies reduces the footprint, volume area, and weight, which in turn increases the package density. Nevertheless, for 3D-SICs to be widely accepted for a wide range of applications cost is still a limiting factor

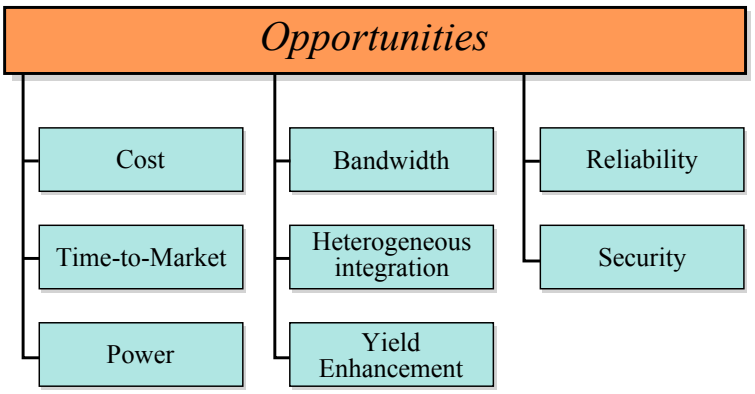


Figure 1.9: 3D-SIC opportunities.

as the cost depends on the yield learning curve driven by the cumulative produced 3D-SICs.

- **Time-to-Market:** Once the stacking technology matures, time-to-market may be reduced due to die reusability. Figure 1.10(a) shows the technology requirements (such as performance, power, etc.) of several application markets (such as consumer, automotive, medical, etc.). It shows for each market and technology combination the More-Moore and/or More-than-Moore driver impact. The figure illustrates for the different market segments the need for diversification, which may be offered by heterogeneous integration in a modular 3D die design. For example, DRAM, sensors, MEMS, and other analog and RF designs might be reused without a redesign or left

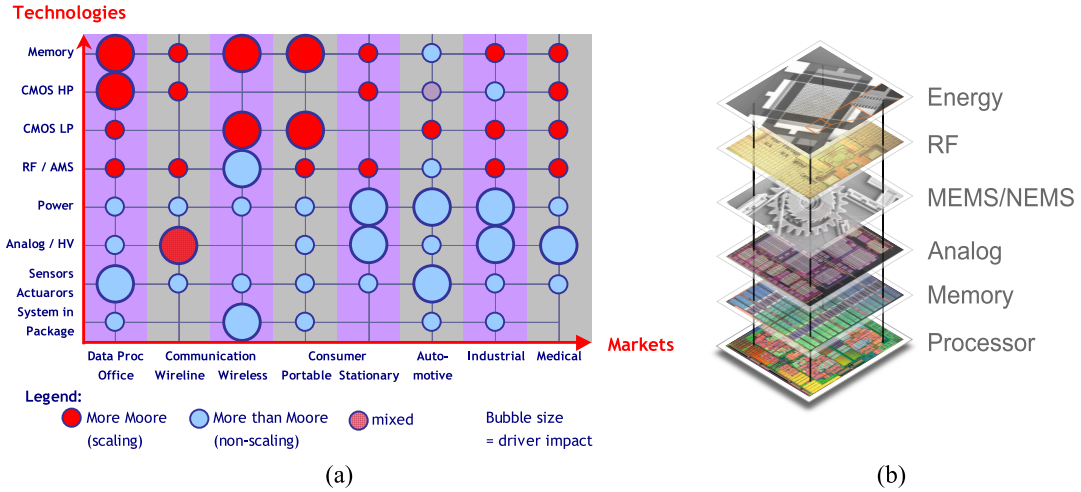


Figure 1.10: (a) Market vs technology requirements [28], (b) Heterogeneous 3D-SIC.

implemented at older and cheaper technology nodes. Therefore, 3D stacking supports an additional level of flexibility in the (re-)design of systems as compared to SoCs.

- **Electrical performance (power, bandwidth, latency, etc.):** For the past couple of decades, performance enhancement of successive transistor generations was carried out by transistor down-scaling leading to increased speed and higher transistor density. Currently, research shows that obtain further scaling benefits beyond 32nm are challenging [29]. In addition, the gained improvements at transistor level did not solve the bandwidth and latency problems at system level, which for example lead to a serious bottleneck between CPU and memory speed referred to as Memory Wall [30]. Utilizing the third dimension, for example by stacking DRAM layers [31], might be the only way to significantly reduce memory latency and power consumption for future generations of multi-core microprocessors [17]. In addition, stacking provides additional benefits such as reduced power consumption (up to 50% and 25% for standby and active power respectively for four stacked memory dies) [32], reduced noise levels due to the shorter global interconnects and the need of smaller I/O drivers [33]. In general, any efficient partitioning of IP cores reduces long global wires and therefore also the delay and power dissipation [14, 34].

- **Heterogeneous integration:** Stacking dies in 3D makes heterogeneous integration possible as depicted in Figure 1.10(b). This is a promising concept for 3D-SICs, since each layer can be manufactured with different technology and optimized for specific needs such as speed, area, power, etc. This affects yield, performance, and lithography cost positively. For example, DRAM, FLASH, sensors, MEMS, etc., could potentially be integrated into a single 3D-SIC. Heterogeneous integration could also make the complete stack more reliable. Traditionally, fault tolerance and fault prevention methods are used to increase system reliability. Fault tolerance focuses on recovering systems in the presence of faults, while fault prevention targets initial reliable systems by using for example reliable materials or designing the chip with extra safety margins. Using the third dimension the reliability may be increased in several ways. For example, (a) functional units may be shared vertically between dies to increase the fault tolerance, and (b) critical system parts may be implemented using more reliable dies (i.e., with larger feature size) to reduce failures, while the less critical cores may use dies with the latest but less-mature technology.

- **Yield improvement:** Traditionally, yield improvement for 2D memories is based on the use of spare rows and/or columns [35–37]. 3D stacked memories provide additional repair features in the vertical dimension as spares can be accessed on neighbor dies. Preliminary research results shows the significant benefits of using this vertical direction [18, 38, 39].

- **Security:** 3D stacking opens new avenues to increase security, such as [40]:

  - A Face-to-Face (F2F) stacked IC conceals most of its circuitry making it hard

for attackers to access parts of the chip.

– The 3D structure is inherently resilient against most reverse engineering attacks. De-layering a 3D-SIC is very difficult. Obtaining voltage images of the layers is challenging due to overlap of dies. Moreover, the bonding materials used to attach dies would likely blur and attenuate signals.

1.2.2 Challenges

Although 3D-SICs have a lot of potential due its opportunities, several challenges still need to be addressed. These challenges must be resolved prior to mass production. A list of the most challenging issues is provided in Figure 1.11; they are classified according to design, manufacturing, test, and supply chain. Each class is briefly described next.

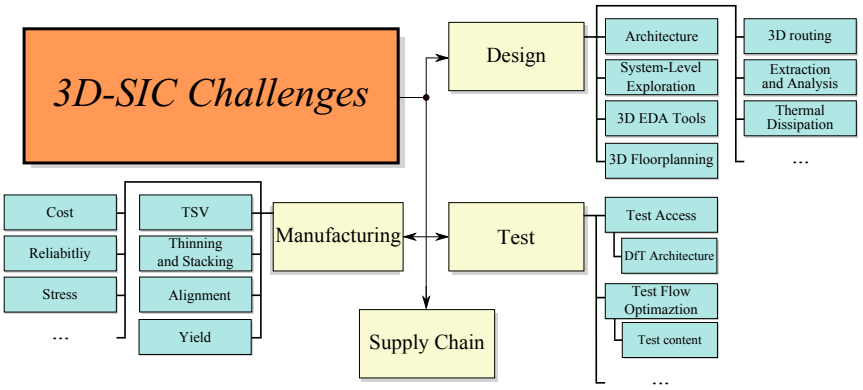


Figure 1.11: 3D-SIC challenges.

- **Design:** One of the key design questions is how to map architectures efficiently into the third dimension. It is therefore important to develop tools that support early but sufficiently accurate system-level explorations in terms of electrical performance (power, frequency), area, thermal budget, cost, etc. [33]. This exploration should guide designers to determine the optimal die sequence in the stack, the technology node for each die, and optimize the interconnection between them. Some system-level exploration tools start to appear such as 3D PathFinding [41] but need more features and automation steps. Once the architecture and rough stack layout are determined, tools are required for the floorplanning and routing.

  TSVs are relative large objects and their number and placement are decisive, especially as a Keep-Out-Zone (KOZ) must be taken into consideration [42, 43]. This KOZ guarantees safe transistor operation as the mechanically induced TSV stress changes the nearby silicon characteristics. In addition to the TSV placement, floorplanners must not only understand the location of each IP block (on die level), but also its place vertically in the stack. Furthermore, the router should be thermal-aware to reduce

hot spots, for example by moving expected hot areas closer to heat sinks. During the routing phase, specific attention must be attributed to the back-side redistribution layer (RDL), and TSV sizes for optimal area placement. The routing algorithm should minimize wire length by taking connection points on adjacent dies into consideration. Other challenges include the distribution of the power grid and clock tree [26]. Dies that are further from the power source are likely to suffer more from voltage drop, clock skew and jitter, but are also be impacted by process variations between the dies.

Accurate tools must be developed to perform parasitic extraction and analysis after the routing phase [33]. In addition to traditional layout parasitic extraction, tools must recognize and integrate RLC parasitics of TSV and micro-bumps and perform thermal analysis for the whole stack. Thinned dies may lead to lower heat dissipation [17]. As the temperature might raise in the stack special care for heat flux must be taken into account. A challenging task is to remove the heat from the chips. Traditionally, packages remove heat from the chip by placing heat sinks on top and/or on the bottom of the chips. At this moment, the 3D-SIC packaging technology is under intensive development and roadmaps have yet to be defined for it [44]. TSVs could help removing the heat when they are used as heat conductors [45].

- **Manufacturing:** 3D-SIC manufacturing requires additional processing steps as compared to conventional ICs; these include for example the forming of TSVs, thinning wafers, and stacking and bonding wafers or dies as described in Section 1.1.3. Each of these additional steps may introduce new defects to the system. Figure 1.12 [17] shows examples of defects that may occur in the TSVs, micro-bumps, and thinned dies as a result of the 3D processing. Typical defects related to 3D processing may be summarized as follows.

  - Pinhole defects along TSV walls create shorts or low resistance paths between TSVs and the substrate; This causes degradation of the signal quality in terms of strength and speed [4, 46–48].
  - An incomplete fill of TSVs (voids) may originate from insufficient wetting during plating. Voids cause partial opens and increase resistance [4, 46–48].
  - Coefficient of thermal expansion (CTE) mismatch between TSV metal (e.g., copper) and substrate may lead to TSV cracks and sidewall delamination. Both lead to increased path resistance [47–51].
  - Pinch-off of TSVs during plating could lead to increased TSV resistance or partial opens [46].
  - Missing contacts between TSVs and transistors or metal layers cause opens [46, 52].
  - A misalignment of TSVs and μ-bumps increase the resistance and cause (partial) opens [46–48].
  - Crosstalk between different TSVs [48, 53].

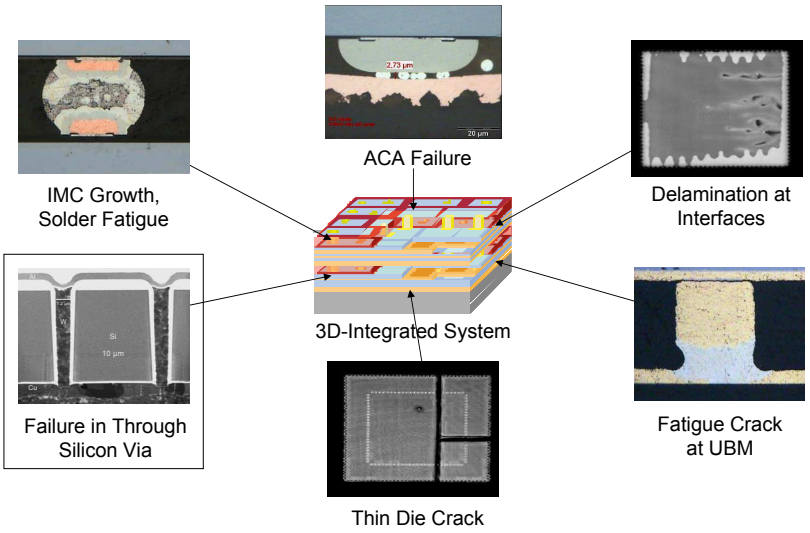


Figure 1.12: Examples of 3D failure mechanisms [17].

- Damage in underlying BEOL [54].
- Weak bonding due to buckled thinned Si chip [54].
- Variation in TSV heights may cause tin to be squeezed out from μ-bump causing shorts between μ-bumps [54, 55].
- Electromigration causes voids and cracks in the joints, resulting in higher resistive μ-bumps, or opens [56].
- Cracks in μ-bumps may be formed due to a CTE mismatch between copper, silicon, and silicon-oxide [46].

In order for 3D-SICs to be commercially viable, a high-yield manufacturing process is required. To achieve that, defects must be repaired or tolerated as 3D technology is currently in its infant stage. For example, several research publications already analyzed the impact of TSV redundancy schemes [57–59] to increase the TSV interconnect yield. In addition to a satisfactory yield, testing is required to keep defective ICs out from the market; this topic is described next.

- **Test:** Testing is one of the biggest challenges of 3D-SICs due to its number of potential test moments. Figure 1.13(a) shows the conventional 2D test flow for planar wafers [55, 60]; it consists of two test moments: a wafer test prior to packaging and a final test after packaging. 2.5D/3D-SICs, however, provide additional test moments. In general, four test phases can be distinguished for a 3D-SICs consisting of n dies as depicted in Figure 1.13(b): (1) n *pre-bond* wafer tests, (2) n-2 *mid-bond* tests, (3) one *post-bond* test prior packaging and (4) one *final* test; resulting into 2·n test mo-

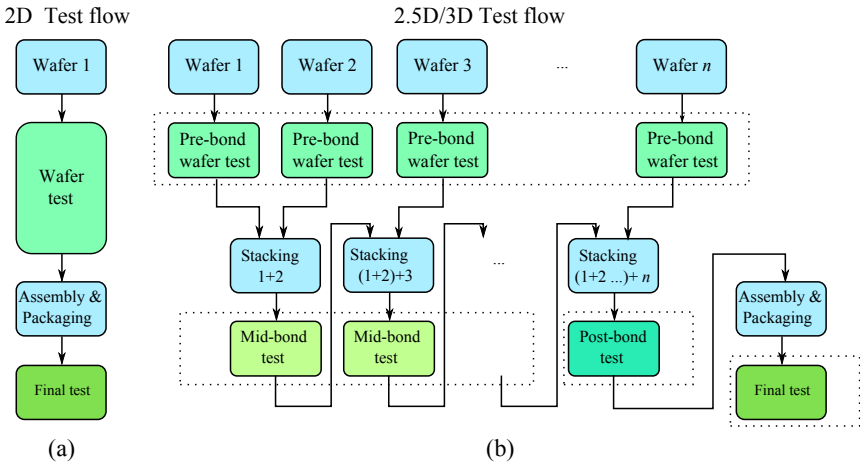


Figure 1.13: 2D versus 2.5D/3D D2W test flows.

ments [4].

The test challenge can be sub-divided into two main categories: (i) test access and (ii) test flow optimization.

- **Test Access:** The test access can be divided further into two subcategories: external and internal access (or DfT architecture). As non-bottom wafers are not designed with external I/O pins, pre-bond testing of such wafers comes with extra challenges. One option to access these wafers is by using dedicated test pads. The main disadvantages of these pads are their area overhead and undesired load capacitance in the final stack. Efforts are taking place to perform direct probing on the micro-bumps [61] which makes the probe pads superfluous. However, manufacturing a fine pitch probe card is challenging. Probing dies that are already thinned may lead to serious IC damage [62]. Testing the non-bottom dies in the other phases, i.e., during mid-bond, post-bond or the final phase requires proper DfT in the stack to forward test data to the specific die under test. Note that only the bottom die has external I/O pins. IEEE P1838 [63] is currently a DfT standard in development for digital stacked ICs; it is based on the presence of boundary scan cells in all dies.
- **Test Flow Optimization:** Test flow optimization can also be divided into two subcategories: test content and test order. Each test covers a set of faults (which are higher abstract presentations of defects). Generating test patterns for each die in the stack may follow a similar flow as used in traditional 2D. However, new type of defects may arise. For example, the mechanical stress induced by TSVs might impact (negatively or positively) the transistor speed [64]. Thinning of dies leads to shifts in transistor $I - V$, impacting both speed and power [65].

In addition, the new introduced components, the TSVs, should be tested. At-speed interconnect testing (for TSVs and micro-bumps) is challenging due to low latency interconnects.

Once the content is defined, the order in which interconnects (e.g., interconnect, die) and dies are tested might impact the overall cost. Testing first for defects that are likely to occur reduces average test time. Early testing might prevent further assembly costs such as the stacking of good dies on defective partial stacks, but may also impact the overall cost negatively. The total number of test moments, equal to twice the number of dies in the stack [4], further complicates finding optimal test flows.

- **Supply Chain:** There are some complex logistic issues that need to be solved for 3D-SIC. For example, responsibility should be taken for yield and inventory risk during the 3D manufacturing process. More precisely, responsibility should be taken for TSV manufacturing, FEOL, BEOL, thinning and bonding, testing (during the different phases), and packaging [66]. Other concerns may be delays between suppliers and transportation of (thinned) non-packaged wafers, as they may affect the yield.

1.3 Research Topics

The research that is carried out in this thesis can be divided mainly into three parts.

1. Yield improvement techniques.
2. Cost modeling, mainly focusing on test cost optimization.
3. Interconnect testing and diagnosis for memory stacked on logic.

Yield Improvement

As yield is one of the major concerns for 3D-SICs, yield improvement techniques should be developed from transistor level up to application level. In this dissertation, we focus on yield improvement in W2W stacking as the yield drops quickly with the increasing number of dies. This is a direct consequence of stacking good dies on bad dies and vice versa. Methods to improve this yield are required. Hence, efficient wafer matching algorithms are needed to maximize the compound yield.

Another interesting research topic is repair for yield improvement. Due to their regular structure, 3D stacked memories are good candidates for such schemes. In 2D memory, each die comes with its own redundant cells typically realized by spare rows and/or columns. In 3D memory, in addition to 2D repair, repair schemes of defective memory cells could utilize the third dimension. This gives defective cells more room for repair and therefore could improve the compound yield.

Cost Modeling

Each 3D-SIC must be tested as a consequence of many high-precision defect-prone steps. Testing identifies the defective chips and guarantees the end-of-line product quality. There are many possible test moments in the 3D manufacturing flow (see Figure 1.13); they are: pre-bond, mid-bond, post-bond, and final testing. Despite the cost of each test, it may filter defective components in an early stage, which prevents down-stream costs. In particular, each applied test has a particular *value* as it could (a) prevent faulty dies from entering good stacks (pre-bond test), (b) prevent stacking of good dies on faulty partial stacks (mid-bond test), (c) prevent packaging costs (post-bond test), and (d) prevent the shipment of defective parts to customers (final test).

Test flows, which consists of tests applied at some or all test moments, needs to be optimized based on yield and cost parameters of individual products; this is a complex optimization problem due to the various test moments. Once the test flow is determined, proper DfT must be added to the chip at design time. For example, this can be Memory Built-in Self Tests (MBISTs), Boundary Scan, scan chains, etc. This demands a sophisticated tool that is able to evaluate the trade-offs between test cost and test value of all possible test flows, during the early design stage.

Interconnect Test and Diagnosis

One of the challenges related to interconnect testing is to perform at-speed post-bond interconnect testing. Prior research publications focused on testing these interconnects using boundary scan. However, at this stage they typically fail to address dynamic faults and at-speed testing. Testing TSV interconnects between the two dies is challenging, as the dies in the stack might come from different manufacturers, which is typically the case for DRAM stacked on logic. In addition, memory vendors have not always been in favor of integrating IEEE 1149.1 on their devices [67].

1.4 Contributions

The contributions of this dissertation are directly related to the research topics presented in the previous section.

1.4.1 Yield Improvement

Several methods are deployed to boost the compound yield. We focus on the following methods: (i) wafer matching [68, 69], (ii) layer redundancy [38, 70], and (iii) inter-layer redundancy [71]. The first method, wafer matching, is applicable in the case where entire wafers are stacked using W2W bonding. By using wafer maps with faulty die locations, suitable wafer pairs can be selected for stacking to obtain higher compound yield as compared to blind stacking. Wafer matching does not come for free as it requires pre-bond testing.

Proper cost-trade off analysis are performed that show the added value of wafer matching, i.e., lower 3D-SIC cost. The second and third methods focus in particular on memories and utilize the third dimension to increase the compound yield. Both methods apply repair on different granularities. Layer redundancy focuses on the repair at the die level where complete faulty dies are replaced. However, inter-layer redundancy focuses on repair schemes within the memory array where spares can be accessed on neighbor dies. Both yield repair schemes are presented, analyzed and evaluated.

1.4.2 Cost Modeling

We present the tool 3D-COSTAR that is able to perform adequate cost prediction at the early design stage. To our knowledge, we are the first to introduce such a tool that is able to incorporate all test moments of the production cycle. 3D-COSTAR is able to evaluate test flows for 3D-SIC; the tool considers all costs involved in the 3D-SIC production (including design, manufacturing, testing, packaging and logistics) and attributes the cost to end-of-line passing products [72–76]. It is aware of the stack build-up (2.5D, 3D, multiple towers), stacking orientation (face-to-face, back-to-face, or face-to-back), and stacking process (die-to-die, die-to-wafer, or wafer-to-wafer). The tool allows us to evaluate several interesting case studies; some of them are listed next.

1. Trade-off between test quality and area overhead for passive interposers in 2.5D-SICs [77]. Interposers do not contain active logic and therefore, are difficult to test. To facilitate pre-bond testing, additional DfT structures must be embedded into the design. However, this impacts both the yield and die size.

2. Impact of the post-bond test quality for a given packaging cost [78]. The post-bond test is the last test opportunity for testing before the 3D-SIC is packaged.

3. Impact of the stacking order [79]. Changing the stacking order impacts the overall 3D-SIC cost in case mid-bond testing is performed.

4. Cost trade-off between testing by means of dedicated pads versus micro-bump probing [77]. Dedicated test pads increase the area (on non-bottom dies), while probing on micro-bumps requires fine-pitch low-force probe cards.

5. Analysis of (test) cost versus product quality (expressed in number of test escapes) [60].

1.4.3 Interconnect Testing

We present a methodology to test interconnects in memories-stacked-on-logic without the need for additional DfT [80, 81]. The assumption made here is that the logic die contains a memory controller or CPU such that TSV interconnects are tested by performing appropriate write and read instructions. These instructions function as test patterns that target specific faults. As defects in TSV are primarily timing related (see Section 1.2.2), it is required that dynamic faults have to be covered by the test. We have developed several test sets to detect

all targeted faults both for address and data line TSVs. Control lines have been assumed to be tested implicitly. In addition to testing, also diagnosis algorithms have been presented [82]. These algorithms are able to identify both the fault location and fault type of all targeted faults. We compared our proposed method with general interconnect DfT schemes such as IEEE 1581 and IEEE 1149.1, but also with dedicated Built-in-Self-Test (BIST) architectures.

1.5 Thesis Organization

The remainder structure of this dissertation is organized as follows.

Chapter 2 discusses the contributions of this dissertation with respect to yield improvement. It presents the proposed yield improvements techniques by describing their working principles. In addition, it summarizes the state-of-the-art in this field and presents our contributions. The publications accompanying this chapter can be found in Appendix A.

Chapter 3 discusses the contributions of this dissertation with respect to cost modeling. We first argue the need for such a tool followed by previous work in this area. Shortcomings clearly show the uniqueness of our tool. The publications accompanying this chapter can be found in Appendix B.

Chapter 4 presents our test and diagnosis approach for interconnects in memories stacked on logic. It first explains the need for such an approach and presents after that our contributions with respect to the start-of-the-art. The publications accompanying this chapter can be found in Appendix C.

Finally, the conclusions and future work are presented in Chapter 5.

Chapter 2

Yield Improvement

The content of this chapter is based on the following research articles:

1. **M. Taouil**, S. Hamdioui, J. Verbree, and E.J. Marinissen, "On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *International Test Conference (ITC)*, Austin, TX, USA, Nov. 2010, pp. 1-10.

2. **M. Taouil**, S. Hamdioui and E.J. Marinissen, "Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching," *submitted to ACM Transactions on Design Automation of Electronic Systems (TODAES), pp. 1–24, 2014*.

3. **M. Taouil** and S. Hamdioui, "Layer Redundancy Based Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *European Test Symposium (ETS)*, Trondheim, Norway, May 2011, pp. 45–50.

4. **M. Taouil** and S. Hamdioui, "Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 4, pp. 523-534, Aug. 2012.

5. M. Lefter, G.R. Voicu, **M. Taouil**, M. Enachescu, S. Hamdioui, and S.D. Cotofana, "Is TSV-based 3D Integration Suitable for Inter-die Memory Repair?" *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, March 2013, pp. 1251-1254.

2.1 Introduction

The compound yield of 3D SICs is one of the major challenges as the technology still needs to mature, especially for wafer-to-wafer (W2W) stacked ICs. W2W has several advantages over die-to-wafer (D2W) and die-to-die (D2D) stacking such as a high stacking throughput and the ability to handle thin wafers and small dies. However, it suffers from low compound yield as the stacking of good dies on bad dies and vice versa cannot be prevented. Several methods can deployed to boost this compound yield. In this chapter, the focus is on three of those methods: (i) *wafer matching*, (ii) *layer redundancy* and (iii) *inter-layer redundancy*.

The first method, *wafer matching*, can be generally practiced on all kind of wafers. In wafer matching, a software algorithm keeps track and matches wafer maps; each wafer map contains faulty die locations of a particular wafer. The algorithm matches wafers based on the similarity of fault maps. This increases the compound yield over randomly stacked wafers.

The other methods, *layer redundancy* and *inter-layer* redundancy, are applied in this work only to 3D memory. Nevertheless, they can be applied to any type of 3D-SIC. Traditionally, the memory yield improvement in 2D chips is realized by using spare rows and/or columns to repair defective ones. 3D stacked memories allow the exploration of new repair schemes that take advantage of the vertical dimension. In *layer redundancy*, repair takes place at the wafer level; additional redundant layer(s) are stacked to replace the faulty irreparable memory dies in the stack. In *inter-layer redundancy*, a non-repairable layer (i.e., the number of defective rows and/or columns is more than the available number of spares), borrows additional spares from the neighboring layers. A drawback of this approach, when compared to layer redundancy, is the additional required number of TSVs and the routing complexity to mutually share and access the spare resources among the layers in the stack. Nevertheless, it provides a more effective repair capability.

2.2 Main Contributions

This section describes the state-of-the-art and main contributions of the introduced yield improvement techniques.

2.2.1 Wafer Matching

The compound yield can be improved by wafer matching, initially introduced by Smith et al. [27]. In [83], Ferri et al. used wafer matching to increase the parametric yield of a two layered D2W stacked 3D-SIC. Only functional dies are considered in this case to produce an optimal binning; i.e., maximize the fastest speed bins and minimize the slowest ones. Wafer matching is then used to combine and improve the 3D parametric yield by including the process variation of both layers in a D2W stacking approach. The authors were able to increase the number of 3D-SICs in the fastest speed bins as well as simultaneously reducing the num-

ber of slow 3D-SICs. More elaborated studies of wafer matching regarding the functional yield are presented in [84, 85], e.g., by considering different die yields, stack and repository sizes, etc. In [86] the author presents wafer rotation; each wafer can be rotated with predefined angles before stacking. However, this imposes restrictions on the die orientations. Rotating wafers gives more freedom in stacking and therefore increases the compound yield. In [87,88] the same author presents a model that also considers radial defect clustering; these publications show that the compound yield is higher than the case where wafers are considered to have a random defect distribution.

All the related previous work considered static repositories (i.e., the repositories are not replenished unless they are empty) and used a single wafer matching criterion (matching of the good dies from the bottom layer with the good dies from the top layer). However, the matching could also be based on faulty dies instead of good dies. Our contributions [68, 69] are summarized as follows.

- The introduction of the concepts *matching process*, *matching criterion*, and *matching scenario*. The matching process defines how the repositories are traversed and how many wafers are selected from each repository visit at a time. The matching criterion specifies whether the matching of good or bad dies are maximized or whether the mismatch of good and bad dies are minimized. The matching scenario is defined by its matching process, matching criterion, and whether or not wafer rotation is applied, and its repository type (i.e., static or running repositories). Note that the matching process mainly determines the time and memory complexity of the matching scenario.

- The impact of several matching processes and matching criteria on the compound yield of 3D-SICs have been analyzed for running repositories. In running repositories, wafers are immediately replenished after matching.

- The optimal matching scenario for running repositories strongly depends on the yield of the stacked dies. We have created a Best Pair scenario that adaptively selects the optimal matching criterion based on given yields.

- Several comparisons are performed between static and running repositories using different matching processes both with and without wafer rotation.

- A new framework is constructed that covers all matching processes and wafer matching criteria for both static and running repositories. The framework does not only allow us to map prior work on it, but it also shows the space of uncovered matching scenarios.

2.2.2 Layer Redundancy

With respect to layer redundancy, the following contributions are made [38, 70].

- A classification of 3D memories and 3D memory redundancy repair schemes is provided. The partitioning of memories across multiple device layers can take place at different granularity resulting in different architectures. Our 3D memory classification shows the advantages and disadvantages of each partitioning scheme. The redundancy schemes for 3D stacked memories can be classified into three groups, i.e., intra-layer, inter-layer and layer redundancy. Intra-layer redundancy accesses uses local spares only (located on the same die), inter-layer redundancy may access spares on neighbor dies, and in layer redundancy faulty dies are completely replaced by spare dies.

- An analytical model is presented that formulates the compound yield improvement by using layer redundancy. This model takes into corporation the yields of the pre-bond dies, the die yield of the stacking operations, and the interconnect yield.

- A comparison of 3D W2W stacked memories with and without layer redundancy is presented in terms of yield and overall cost. The question rises whether it is cost-wise justified to increase the yield by adding more redundant layers. Therefore, the yield comparison is expanded to a cost comparison in which both yield and manufacturing cost are included. The results show huge yield and cost benefits.

- A memory layer replacement circuit that maps the addresses of faulty memory layer(s) to the spare layer(s) is developed. This circuit converts these addresses at run-time with a minimum timing penalty.

- In addition to the above, we have investigated the merged effect of applying simultaneously wafer matching and layer redundancy. First, a comparison is made between layer redundancy and wafer matching. Thereafter, both methods are merged into a single combined technique. The results typically show that layer redundancy outperforms wafer matching both from yield and cost viewpoint. When both methods are merged, further yield and cost improvements are obtained.

2.2.3 Inter-Layer Redundancy

Several authors presented inter-die memory repair as a means to increase the compound memory yield [16, 39, 89–92]. All these publications focused primarily on yield benefits typically evaluated through fault injection simulation. However, the obtained yield improvements form a theoretical upper bound and the challenges of actual silicon implementations have been simply overlooked; examples are the impact on area, layout and latency. Proper infrastructure must be embedded in the 3D memory to allow spares to be shared vertically in the stack. Our contributions are as follows [71].

- An overview of possible spare access scenarios in a 3D memory cube based on spare providers and spare consumers is provided. The spare providers have available spare resources and the spare consumers make use of externally available spares of the neighbor dies. Each provider-consumer pair satisfies one of three possible scenario's

[71]: (i) Idle provider - the two stacked arrays of the provider and consumer are part of different banks that are never concurrently accessed; (ii) Busy provider with different access pattern - the two arrays are part of different banks that are concurrently accessed with independent addresses (e.g., by having multiple memory ports); (iii) Busy provider with same access pattern - the two arrays are part of the same interleaved bank; therefore, they have the same address.

- Several implementation schemes are provided both for inter-die row and column repair with detailed circuit infrastructure. Advantages, disadvantages of the impact on memory area and latency are evaluated for each scheme. The results suggest that current state-of-the-art TSV dimensions make inter-die column repair schemes feasible at the expense of reasonable area overhead. However, most row-repair memory configurations require TSV dimensions to scale down at least with one order of magnitude for practical implementations.

- We performed theoretical analysis of the implications of the proposed 3D repair schemes on the memory access time.

2.3 Evaluation

Our results and analysis show that the compound yield can be improved by using wafer matching with running repositories which have a lower time and memory complexity. Compared to the state-of-the-art, running repositories outperform static repositories irrespective of the design and manufacturing parameter values (e.g. stack size, die yield), and by using a relative less complex matching process. The best matching criterion to be used for highest compound yield improvement is strongly stack size and die yield dependent; hence, using adaptive matching criterion selection is the optimal solution.

In addition, it is worth to mention several interesting aspects related to wafer matching.

- The absolute compound yield of W2W stacked 3D-SICs is typically low. Therefore, the applied redundancy schemes presented in this chapter impact the compound yield and reduce the cost significantly. Nevertheless, the absolute yield remains low. Hence, W2W stacking should be considered only out of necessity (like stacking small dies), or when the die yield is high.

- The down-side of the matching process in [85] for static repositories is the forcing of stacking bad wafers when the repositories become emptier. The authors presented a scenario with a greedy matching process; each time the two wafers with the highest yields are selected out of the repositories for stacking. Hence, bad wafers remain in the repositories till the end. To counteract this problem, we have proposed running repositories in which wafers in the repositories are directly replaced after being selected for stacking. This approach does not increase the run-time and memory-complexity of the

algorithm and more importantly, the proposed matching is performed each time using full repositories. However, the authors of [84] presented an optimal algorithm for static repositories based on interlinear programming, which quickly runs out of memory and its execution time is a major bottleneck even for limited number of stacked dies and reasonable repositories sizes.

- In [87] the author introduced wafer rotation where wafers can rotate with angles of 90, 180 or 270 degrees. Obviously, wafer mask designers have to take this into consideration in order to make it feasible, although it may impact the die yield. In addition, stacking equipment need to be modified to support the rotation of wafers. Therefore, rotating is an interesting concept but practically hard, if not impossible, to realize. Moreover, the additional yield benefit due to wafer rotation is marginal [69]. In [93], the authors generalize this concept further by cutting wafers into segments prior to stacking; this leads to a stacking approach between D2D and D2W stacking. This is even more complex to realize as it is very demanding in terms of processing and equipment.

- The impact of wafer matching reduces when the radial clustering defect is considered; the yield benefits in wafer matching are due to random defects. In case the wafers to be stacked are from the same manufacturing line, higher compounds yields are expected as both wafers will most probably suffer from the same systematic defect distribution (e.g., defects at the edge of the wafers); hence, wafer matching is less effective in such cases. In case wafers are coming from different manufacturing lines (e.g., in DRAM stacked on logic), lower compounds yields are expected when the location of the systematic defects on both wafers differs.

- Running repositories may also have practical implementations. One of its concerns is a polluted repository in which bad wafers would remain for a long period in the repository, thereby reducing the effective repository size. Proper filters that force such wafers to be removed from the repository could become necessary. In addition, prefilters could be set in place to prevent wafers with a very low yield from entering the repository; these wafers need to be processed separately. In particular, the process of replenishing wafers needs attention. One implementation is to consider a secondary repository in which wafers are only used to replace selected wafers from the main repository.

Layer redundancy improves the yield and reduces the cost significantly as the absolute compound yield is low. In inter-layer redundancy, the repair occurs on a much finer granularity (therefore, it is area-wise more effective) and its theoretical yield improvements are even better [16, 39, 89–92]. However, layer redundancy is from a practical point easier to implement. Our preliminary conclusion for inter-layer redundancy shows that only inter-die column redundancy is feasible with current TSV sizes. In addition, more research is required to conclude its practicality. For example, accurate timing analysis need to be performed for a memory layout which includes the redundancy repair logic, the required TSVs to access

the spare cells, and their KOZ. This is important as the timing is very critical in memories [37]. In addition, more research is required for low-cost inter-layer repair schemes for stacks containing more than two dies.

Chapter 3

Cost Modeling

The content of this chapter is based on the following research articles:

1. **M. Taouil** and S. Hamdioui, "On Optimizing Test Cost for Wafer-to-Wafer 3D-Stacked ICs," *7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Tunis, Tunisia, May 2012, pp. 1–6.
2. **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking," *19th IEEE Asian Test Symposium (ATS)*, Shanghai, China, Dec. 2010, pp. 435–441.
3. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "How Significant will be the Test Cost Share for 3D Die-to-Wafer Stacked-ICs?" *6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Athens, Greece, April 2011, pp. 1–6.
4. **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Impact on the Overall Die-to-Wafer 3D Stacked IC Cost," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 1, pp. 15-25, Feb. 2012.
5. **M. Taouil** and S. Hamdioui, "Stacking Order Impact on Overall 3D Die-to-Wafer Stacked-IC Cost," *14th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Cottbus, Germany, April 2011, pp. 335–340.
6. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," *6th IEEE International Design and Test Workshop (IDT)*, Beirut, Lebanon, Dec. 2011, pp. 24–29.
7. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Using 3D-COSTAR for 2.5D Test Cost Optimization," *IEEE International 3D Systems Integration Conference (3DIC)*, San Fransisco, CA, USA, Oct. 2013, pp. 1–8.
8. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Impact of Mid-Bond Testing in 3D Stacked ICs," *16th IEEE Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, New York, NY, USA, Oct. 2013, pp. 178–183.
9. **M.Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Quality versus Cost Analysis for 3D Stacked ICs," *32nd IEEE VLSI Test Symposium (VTS)*, Napa, CA, USA, April 2014, pp. 1–6.
10. E.J. Marinissen, B. de Wachter, K. Smith, J. Kiesewetter, **M.Taouil**, and S. Hamdioui, "Direct Probing on Large-Array Fine-Pitch Micro-Bumps of a Wide-I/O Logic-Memory Interface," *International Test Conference (ITC)*, Seattle, WA, Oct. 2014, pp. 1–10.

3.1 Introduction

ICs are manufactured in highly specialized fabs following a long sequence of defect-prone steps [94]. This makes testing an unavoidable expense as it must identify deficient ICs and prevent them from being shipped to customers. In return, testing adds value as it reduces the number of test escapes which satisfies the customers [95]. In particular markets, such as the automotive, medical, and aerospace industry, extremely low test escape rates are demanded [96]. Therefore, testing is subjected to cost versus quality trade-offs [97].

Defining optimal test strategies for 3D-SICs is more challenging as compared to planar ICs due to the significant increase in number of test moments. A planar IC has typically only two test moments; a wafer test prior to packaging and a final test after packaging [98]. The wafer test is cost-effective when the cost of faulty packaged ICs would exceed the test cost. The final test is used to guarantee the final quality of the packaged chips. However, 3D-SICs have many test moments denoted by a pre-bond, mid-bond, post-bond and final test. Pre-bond tests may prevent defective dies from entering the stack, while mid-bond (for partial stacks) and post-bond tests (for the complete stack prior to packaging) are used to verify the dies and interconnects after stacking. In addition, the post-bond test prevents faulty 3D-SICs from being packaged. The final test, applied after packaging, can be used to satisfy the product quality.

Testing dies in all the test phases might lead to test overhead. Therefore, careful analysis must identify in which stage to test. Whether or not to perform a particular test depends primarily on the yield. Even Known Good Dies (KGD) obtained through the pre-bond test are not guaranteed to survive the stacking processing; new defects from the stacking process are unavoidable. Typical sources of failures during stacking include thinning, bonding, as well as TSV failures such as misalignments and opens [62]. Depending on the quality or yield of such a stacking process, retesting of dies in the stack might be cost-wise favorable. If it is known beforehand that a particular stack is malfunctioning, additional silicon, stacking, and bonding costs can be prevented for the successive dies that have to be stacked. Therefore, early testing may prevent further down-stream processing costs.

Figure 3.1 shows the typical IC production cycle. There is a design, manufacturing, test, and packaging phase prior to the shipment to the customer. The customer demands products that work during the prescribed life time. This puts certain constraints on the quality of the test. The challenge is to single out proper test flows, i.e., to find out in which test phase to apply tests and to define what exactly to test for. This strongly depends on inputs from design, manufacturing and packaging as well. After the test flow is determined, proper DfT must be added to the chip to support the desired fault coverage, for example by using Memory Built-in Self Tests (MBISTs), Boundary Scan, scan chains etc. This DfT has to be inserted in the netlist at design time. Therefore, a sophisticated cost model must be developed that incorporates not only the test phases (pre-, mid-, post-bond and final testing), but also

manages cost inputs from other classes such as design, manufacturing, etc. In addition, it should be able to estimate the total 3D-SIC cost and its product quality.

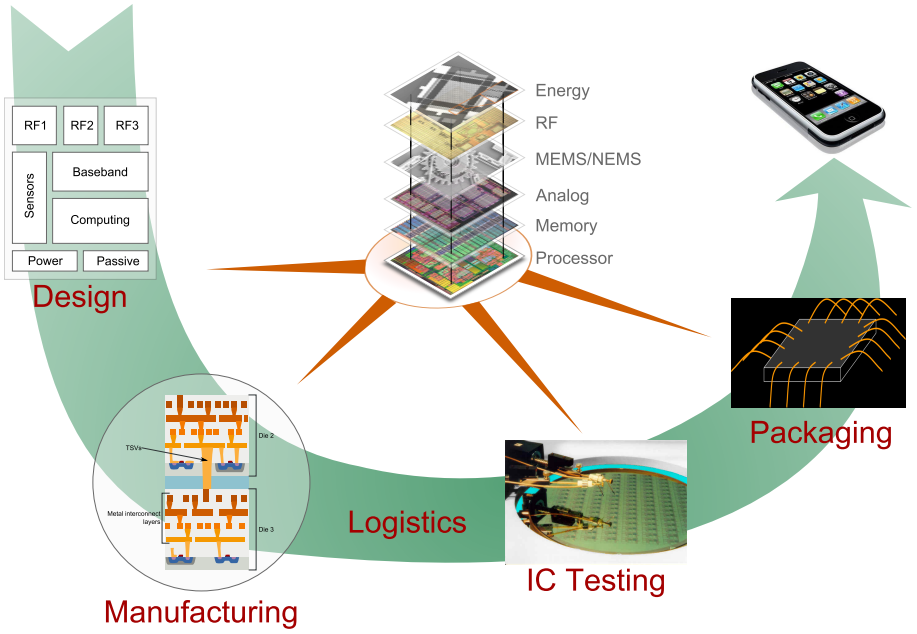


Figure 3.1: From design to 3D-SIC.

3.2 Main Contributions

Several cost models have been published in the 2.5D and 3D area. Most of them have focused on cost modeling for 3D manufacturing [99, 100], stacking and integration [101–106], TSV count and die area [107, 108]. However, limited work is published on test cost modeling and its impact on the overall chip quality and cost. In [109], the authors proposed a cost model that emphasizes on manufacturing and test cost; the authors investigated the impact of Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) stacking on overall cost and determined the lower bound of the yield of the final package level test given the number of stacked dies and the final yield. In [110], the author propose heuristics to find cost-wise optimal test flows that include mid-bond testing. They consider only manufacturing and test cost.

The state-of-the art described above shows that none of the published cost models incorporated the impact of partial stack tests while considering costs over the whole production chain. In addition, none of them is able to estimate the product quality. Our contributions [60, 72–79] address most of these shortcomings and are summarized next:

- Classification of test flows.

A systematic classification of test flows. The classification is based on the performed tests (i.e., no test, die test only or die and interconnect tests) during pre-bond, mid-bond, post-bond and final testing. For W2W stacking, mid-bond testing has not been considered as it presumably impacts the cost negatively; wafers have to be stacked on top of each other.

- Cost model for 2.5D-, 3D-SIC and 5.5D-SIC with its software implementation.

  We present 3D-COSTAR, a tool that considers the costs of the whole 2.5D/3D-SIC chain, including design, manufacturing, test, packaging and logistics and provides the estimated overall cost for 2.5D/3D-SICs for the end-of-line passing products, and the cost breakdown for a given input parameter set, e.g., test flows, die yield and stack yield. In addition, it estimates the number of test escapes measured in DPPM. Hereby, we used different fitting functions for yield, fault coverage and test escapes. The modularity of the tool supports any test flow for any of the 3D technologies (2.5D-SIC, 3D-SIC, or multiple tower).

- Analysis of several 2.5D- and 3D-SIC test flows using 3D-COSTAR.

  For D2W stacking, several case studies are evaluated for different 2.5D-SIC test cost optimization problems. Examples of such case studies show that for given parameters: (a) pre-bond testing of the passive silicon interposer is important for overall 2.5D-SIC cost reduction; the higher the fault coverage, the lower the overall cost, (b) using micro-bump probing results in much lower overall cost as compared to probe-pads, and (c) mid-bond testing can be avoided for high stacking yield.

  As mid-bond testing increases the amount of wafer transport, we investigate the impact of logistics as well. Two logistics models are compared. In the first model, referred to as the extensive model, non-zero values are assumed for wafer transports between all companies. Here, each company performs a certain step such as wafer manufacturing, pre-bond test, etc. For the second model, a reduced logistics model is assumed in which the manufacturing and pre-bond tests are performed by the wafer fab, and the remaining activities (such as, stacking, mid-, post-bond bond and final tests and packaging) by the outsourced semiconductor assembly and test (OSAT). The results show for the most relevant test flows that, as long as the transport cost per single wafer is low, the overall impact of the logistics is for both logistic models minor.

  Many test flows have been investigated for 3D-SICs based on variable fault coverages during the different test phases. The results show that choosing an appropriate test flow the overall 3D-SIC cost can reduce the overall cost up to 20% for a five-layer 3D-SIC with die yields of 90%.

  For W2W stacking, analysis are performed for all the test flows in the classification framework. Objective here was to maximize the compound yield and minimize overall cost. The results show that the pre-bond testing has the most impact on cost. The benefit of the pre-bond test is that wafer matching can be applied. In most of the cases,

this proved to be beneficial except for certain conditions such as high die yields. In the post-bond test phase, primarily interconnect tests are of relevance. In the final test, we assumed full tests for dies and interconnects to guarantee the final product quality.

- Out-of-order stacking to reduce cost.

  In-order stacking restricts the stacking of the dies in a bottom-up sequential order, while out-of-order stacking poses no restrictions as long as the stacking order is realistic. The comparison and the analysis of the two stacking approaches have been performed while varying several parameters; these parameters include different stack sizes, die yields, stack yields for all the different test flows. The results show that out-of-order stacking results in equal or lower overall cost as compared to in-order stacking; this is because testing during out-of-order stacking reduces the number of wasted faulty dies (or partial stacks). This cost reduction depends on the selected test flow. The reduction becomes more significant as the stack size increases or when the stacking yield decreases.

3.3 Evaluation

3D-COSTAR has been presented as a tool to evaluate test flows for D2D and D2W stacking. The tool is able to evaluate all test flows and attributes the costs only to the end-of-line passing products and reports the estimated test escape rate and cost break down.

From our W2W cost model, we conclude that pre-bond testing impacts the cost the most as it enables wafer matching. In the post-bond test, typically only interconnect testing pays off. Therefore, designers must consider to integrate DfT that allows these tests to be performed during pre-bond and post-bond testing. From our D2W cost model, we conclude that pre-bond testing is extremely important as it impacts the overall cost significantly. Testing on the other moments, is heavily dependent on the inputs such as test cost, packaging cost, packaging yield, logistics cost etc. We encourage designers to evaluate test flows at the early stage of the design in order to conclude which test infrastructure is required; this to satisfy the product quality and cost.

At its current implementation, 3D-COSTAR does not include solutions to search for optimal test flows. In [110], the authors present a heuristic to find such test flows. However, they assume fixed inputs for test cost and fault coverage in case a test is applied. New heuristics must be proposed that do not put such constraints on the input fault coverage and its associated test cost. The objective is to find (sub-)optimal input values for the test class for given design, manufacturing, packaging, and logistics inputs that minimize the overall 3D-SIC cost, while the test inputs satisfy the product quality. This is a complex problem as for each test moment, the test content (fault coverage and test cost included) and test order must be defined.

Furthermore, depending on the number of companies that are involved in the manufacturing of 3D-SICS, different constraints can be applied to the pre-bond wafer test quality [55]. Wafers that are manufactured by a company different from the stacking one often require a high quality pre-bond wafer test (KGD test). The pre-bond test quality is subject to optimization in case both wafer manufacturing and stacking are performed by the same company. Faulty undetected dies that escape the pre-bond test can be detected in a later stadium, e.g., in the final test. Similarly, a high quality pre-packaging test (Known Good Stacks test) can be applicable if the packaging is performed by another company.

In the previous chapter, we already concluded that the compound yield in W2W stacked 3D-SICs is typically low. This results in a higher cost for the end-of-line passing products in W2W stacking as compared to D2D/D2W stacking.

Chapter 4

Interconnect Testing and Diagnosis

The content of this chapter is based on the following research articles:

1. **M. Taouil**, M. Lefter, and S. Hamdioui, "Exploring Test Opportunities for Memory and Interconnects in 3D ICs," *International Design and Test Symposium (IDT)*, Marrakesh, Morocco, Dec. 2013, pp. 1–6.

2. **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Interconnect Test for 3D Stacked Memory-on-Logic,"*Design, Automation & Test in Europe (DATE)*, Dresden, Germany, March 2014, pp. 1–6.

3. **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Post-Bond Interconnect Test and Diagnosis for 3D Memory Stacked on Logic," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1–12, 2014*.

4.1 Introduction

The previous chapter focused on test flows and cost modeling for 3D-SIC. In this chapter, the focus shifts towards the required DfT for 3D-SICs. Several DfT techniques have been developed to test planar ICs, such as scan chain insertion, MBIST, and boundary scan [111]. Many of these techniques can be reused for 3D-SICs to test individual layers. However, specific DfT must be added to route data vertically through the stack and/or deal with defects related to 3D stacking and bonding [112].

One of the main applications that utilizes the mentioned benefits of 3D-SICs is the stacking of memory (DRAM) on logic (CPU). After stacking, a post-bond interconnect test is required to test interconnects (TSVs + micro-bumps) between the memory and logic dies. This is not straightforward as (1) stacked dies may come from different providers that want to keep their IPs confidential, (2) memory providers have been in the past reluctant to integrate DfT such as IEEE 1149.1 for interconnect testing, and (3) even with DfT support, obtaining high coverage for dynamic faults is still challenging.

Several interconnect test strategies are under development. IEEE P1838 [63] is standardizing DfT for general stacked ICs; it is based on the presence of Boundary Scan cells in all dies. Lewis and Lee [113] considered pre-bond die testing in order to obtain a satisfactory compound yield. The authors proposed an approach with scan islands based on the IEEE 1149.1 [114] and IEEE 1500 [115]. Marinissen et al. [112] addressed many limitations of previous work by proposing a structured and scalable test access architecture using *TestTurns* and *TestElevators* to route test data through the stack, for pre-, mid- and post-bond tests. The architecture is further extended to support Multiple Tower (MT) stacking [116] and $2\frac{1}{2}$-D stacking [117, 118]. Wide I/O [119, 120] also supports interconnect testing using Boundary Scan. However, (DRAM) memory vendors are not always in favor of integrating IEEE 1149.1 on their devices [67]. Other approaches such as IEEE 1581 [67], originally for 2D ICs, can be extended in the third dimension. In test mode, the memory is bypassed and interconnects are tested by creating a direct logic function between the inputs and outputs of the memory. IEEE 1581 prefers the accompanying logic chip to be IEEE 1149.1 compliant, i.e., the test infrastructure on the memory chip can function with a logic chip that supports IEEE 1149.1. This approach, referred by us as Test Logic (TL) based interconnect testing, also requires additional DfT test logic on the memory die. The test logic consists typically of several XOR gates and a test mode activation circuit. This standard can be mapped to 3D-SICs by having the bottom logic die IEEE 1149.1 compliant and the test logic residing on the top memory dies. On top of the undesired DfT on the memory die, both the BS and TL based test methods are unable to apply at speed tests which are required to target dynamic faults. Testing for dynamic faults is crucial, as 3D interconnects are expected to suffer from speed and timing-related faults [4, 46, 48, 53, 121, 122].

In addition to lack of standards, limited dedicated test solutions with at-speed testing ca-

pability for TSV faults have been published. In [123, 124], authors present hardware BIST approaches to test the TSV interconnects using the Maximum Aggressor Fault (MAF) model. Both methods apply fixed test patterns and have an DfT area overhead; this overhead is approximately 7% for [124] and 9.8% for [123] for 15 μm TSV diameters in 90 nm technology. In [125], the authors test the memory interconnects using the embedded CPU. They target crosstalk faults in planar dies. However, the authors did not address diagnosis. Moreover, the layout of a TSV array in 3D-SICs differs from wire connections in planar ICs.

4.2 Main Contributions

Our contributions with regards to interconnect testing are:

- A framework of interconnect test approaches such as BS and TL for memories stacked on logic. The benefits and drawbacks of each possible solution is extensively discussed for stacked memories both with and without MBISTs, placed on the memory dies or on a separate logic die. The location of the MBIST on the stack impacts the test flow as it may or may not support pre-bond testing. Thereafter, we discuss how they affect quality and memory repair. For interconnect tests, the three test approaches (Boundary Scan, TL and MBIT) are explored.

- A Memory Based Interconnect Test (MBIT) methodology to test interconnects between memory and logic dies by performing read and write operations from the logic die to the memory dies. The logic die must contain a processor that can execute some basic instructions. We first provide a classification of interconnect defects and compiled them into fault models. In addition, we develop the test pattern generation for these faults and used the proposed MBIST to implement them. In addition, several algorithms are presented to perform maximum fault diagnosis (i.e., both fault location and type are diagnosed) without the need for additional DfT as it reuses existing components in the stack. The proposed MBIT supports at-speed testing and detects all static and dynamic faults. Moreover, it is very flexible in altering test patterns simply by modifying software instructions and has a extreme short test execution time. We verify and analyze the test time of our test patterns using a MIPS64 simulator. We compare the MBIT approach with previous hardwired BISTs [123, 124], Boundary Scan, and TL. MBIT results in zero area overhead and allows flexible patterns to be applied, in contrast to the hardwired approaches. The required test time is lower than traditional based solutions such as Boundary Scan, but is slower than the hardwired BIST solutions. However, BIST solutions have a large area overhead and cannot apply flexible patterns.

4.3 Evaluation

In this chapter, we have demonstrated the feasibility of testing and diagnose interconnects for memories stacked on DRAM. The read and write patterns are assumed to be executed from a processor that is embedded in the logic die. Besides processors, also ROMs can be used on the logic die to execute the developed patterns. However, it requires a small control circuit that is responsible for loading these patterns from the ROM, apply them to the memory, and subsequently evaluate the memory responses.

In our current algorithms, we assumed the control signals to be tested implicitly. This may not be applicable for complex memories such as DRAMs as they have sophisticated control signals such as a burst read. The testability of the control signals for such memories needs to be researched additionally. A loss in fault coverage may be unavoidable.

A requirement to perform post-bond interconnect diagnosis is to have both the memory and logic die fault-free. For example, both dies can be tested already during a pre-bond test. In case the memory contains defects, the diagnosis algorithms could report wrong results as they are not able to distinguish between memory and interconnect faults.

The MBIT approach tests not only the TSV interconnects, but the end-to-end path from logic to memory. It can detect all timing problems on these path. The methods that use boundary scan only test the paths between the boundary scan cells of two dies, which might not be the end-to-end path from logic to memory.

Chapter 5

Conclusion and Future Work

5.1 Summary

5.2 Future Research Directions

This chapter summarizes the overall achievements of this dissertation and highlights some future research directions. Section 5.1 presents a summary of the main conclusions presented in this dissertation. Section 5.2 provides the future research directions.

5.1 Summary

Chapter 1, "Introduction", briefly introduced 3D-Stacked ICs. It described the past and future trends of 3D-stacked ICs and provided a 3D stacking classification. In addition, it briefly explained the key manufacturing steps of manufacturing 3D-SICs. 3D-SICS have many benefits over planar ICs. They may lead to improvements in cost, form factor, electrical performance, functionality, repair capabilities and system reliability. However, several challenges still exist in design, manufacturing yield, test, etc. This thesis focused on three of such challenges; they are yield improvement, test cost modeling and interconnect testing.

Chapter 2, "Yield Improvement", discussed several techniques to improve the compound yield of 3D-SICs. The first method investigated various wafer matching scenarios. A framework has been established that covers all different matching processes and wafer matching criteria for both replenished and non-replenished wafer repositories. An adaptive matching scenario has been created that provides the best solution at run-time for running repositories. It selects the best matching criterion as function of the yield. This adaptive approach marginally outperforms yield-wise wafer matching scenarios based on static repositories, but at a much lower memory and time complexity.

The second proposed method is layer redundancy. First, an analytical model is provided to prove the added value of layer redundancy. Second, the impact of such a scheme on the manufacturing cost is evaluated. Finally, these two parts are integrated together to analyze the trade-off between yield improvement and its associated cost; the realized yield improvement is also compared to yield gain obtained when using wafer matching. The results showed that for a typical stack size layer redundancy realizes a significant yield improvement as compared to wafer matching. Next, we combined both methods, i.e., wafer matching and layer redundancy to obtain even better improvements.

Finally, analysis were made to perform inter-die row and column repair by evaluating area overhead and delay. The analysis showed that current state-of-the-art TSV dimensions allow inter-die column repair schemes at the expense of reasonable area overhead. For row repair, however, most memory configurations require TSV dimensions to scale down at least with one order of magnitude in order to make this approach a possible candidate for 3D memory repair. In addition, the implications of the proposed 3D inter-die repair schemes on memory access time were analyzed; the results indicate that no substantial delay overhead is expected.

Chapter 3, "Cost Modeling", discussed mainly cost models for 3D-SICs. A framework covering different test flows for 3D W2W ICs has been compiled. Test flows that include pre-bond tests can benefit from wafer matching. Subsequently, a cost model for W2W is used to evaluate and estimate the impact of test flows on the overall 3D-SIC cost.

Secondly, a cost model has been developed for D2D and D2W stacking referred to as 3D-

COSTAR. The tool considers cost numbers for design, manufacturing, testing, packaging, and logistics and attributes the cost to end-of-line passing product. In addition, it provides the estimated overall cost for 2.5D/3D-SICs and its cost breakdown for a given input parameter set, e.g., test flows, die yield and stack yield. More important, 3D-COSTAR supports cost versus product quality predictions at an early stage of the design. The tool is used to evaluate several test optimization problems. Examples are (i) the impact of the fault coverage of the pre-bond silicon interposer test, (ii) the impact of pre-bond testing of active dies using either dedicated probe-pads or micro-bumps, and (iii) the impact of mid-bond testing and logistics on the overall cost.

Chapter 4, "Interconnect Testing and Diagnosis", focused on interconnect testing for memory stacked on logic, which is one of the attractive 3D applications. System integrators have to provide an appropriate test strategy for such applications. However, they have to deal with black box IPs as IP providers usually refuse to share the IP content. Therefore, developing a low cost and high quality test approaches, while taking these constraints into consideration, is of great importance. A framework of interconnect test approaches for memories stacked on logic have been presented looking further than the only proposed JTAG solutions. The benefits and drawbacks of each possible solution is extensively discusses for stacked memories both with and without MBISTs, placed on the memory dies or on a separate logic die. Furthermore, a new Memory Based Interconnect Test (MBIT) approach for 3D stacked memories is proposed. MBIT can be used both for detection and diagnosis tests. Our test patterns are applied by read and write instructions to the memory and are validated by a case study where a 3D memory is assumed to be stacked on a MIPS64 processor.

5.2 Future Research Directions

Several recommendations are suggested to further research some aspects of topics addressed in this dissertation. They are given next.

- Yield improvement

    1. The matching framework shows us all possible matching scenarios. Several of these matching scenarios have not been implemented yet. Analyzing the whole framework gives a complete insight in the trade-offs between yield improvement and the cost in time and memory complexity of the algorithms.
    2. Several previous publications reported yield improvement schemes for inter-layer redundancy. However, these provide only theoretical limits as physical implementations were not considered. Actual yield improvements must be computed for physical implementations. Moreover, some schemes may degrade the performance; hence, the impact on the layout needs to be analyzed.
    3. The yield improvements gained with layer-redundancy and inter-die redundancy and their associated costs are not compared. In addition, the combination of the

two schemes into a single hybrid scheme can be explored.

- Cost

  1. The proposed cost models primarily focus on test. More extensive analysis of other cost classes such as manufacturing and packaging might be considered to answer other domain-specific questions like whether to manufacture 2.5D-SICs based on glass or silicon interposers.
  2. Currently, 3D-COSTAR assumes independent test inputs. However, in reality DfT, fault coverage, and test cost are strongly correlated. Therefore, developing an appropriate method based on real designs that correlates these inputs and assign them to their associated cost is worth exploring.
  3. 3D-COSTAR cannot recommend an optimal test flow as it only evaluates provided input. It needs additional heuristics to support this. These heuristics should put constraints on important tests such as the pre-bond and final test.
  4. The cost models can be used to evaluate more optimization problems. For example, to perform cost evaluations for test flows of $5\frac{1}{2}$D-SICs. In addition, new experiments could be considered such as analyzing the impact of the test order for both dies and interconnects.

- Interconnect testing

  1. In this work, we presented a low-cost solution for memory stacked on logic assuming a typical SRAM interface. In addition, we assumed that control signals were implicitly tested. Analysis must be performed to investigate the test coverage loss (if any) for these control signals. Moreover, more complex memory interfaces such as DRAM need to be considered.
  2. Our Memory Interconnect BIST requires memories to be stacked on logic. Several approaches have been proposed to address testing digital stacked ICs in general. However, more research is required to develop interconnect standards for non-digital dies in the stack such as sensors, analog circuits, RF etc.

Beyond the challenges related to the discussed topics in this dissertation, there are many other challenges related to 3D-SICs. Several of them have been discussed in Section 1.2.2 such as system-level exploration, floorplanning, general test access and content, reliability, thermal stress, etc.

Bibliography

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, April 1965.

[2] The International Technology Roadmap for Semiconductors, "2007 Edition," ITRS, Tech. Rep., 2007. [Online]. Available: http://www.itrs.net

[3] W. Arden *et al.*, "Towards a More-than-Moore Roadmap," CATRENE Scientific Committee, Tech. Rep., Nov. 2011.

[4] E.J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias," in *International Test Conference*, Nov. 2009, pp. 1–11.

[5] Toshiba, "SiP (System in Package)," March 2004. [Online]. Available: http://www.toshiba-components.com/ASIC/data/System_in_Package_sce0010a.pdf

[6] V. F. Pavlidis and E. G. Friedman, *Three-dimensional Integrated Circuit Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009.

[7] D. Lu and C.P. Wong, *Materials for Advanced Packaging*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[8] E. Beyne and B. Swinnen, "3D System Integration Technologies," in *IEEE International Conference on Integrated Circuit Design and Technology*, May 2007, pp. 1–3.

[9] B. Swinnen, "3D Technologies: Requiring more than 3 Dimensions from Concept to Product," in *IEEE International Interconnect Technology Conference*, June 2009, pp. 59–62.

[10] S. Mick *et al.*, "Buried Bump and AC Coupled Interconnection Technology," *IEEE Transactions on Advanced Packaging*, vol. 27, no. 1, pp. 121–125, Feb. 2004.

[11] K. Kanda *et al.*, "1.27Gb/s/pin 3mW/pin Wireless Superconnect (WSC) Interface Scheme," in *IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, vol. 46, no. 1, Feb. 2003, pp. 186–487.

[12] N. Miura and T. Kuroda, "A 1TB/s 3W Inductive-Coupling Transceiver Chip," in *Asia and South Pacific Design Automation Conference*, Jan. 2007, pp. 92–93.

[13] J. Xu *et al.*, "AC Coupled Interconnect for Dense 3-D ICs," *IEEE Transactions on Nuclear Science*, vol. 51, no. 5, pp. 2156–2160, Oct. 2004.

[14] W. Davis *et al.*, "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Design Test of Computers*, vol. 22, no. 6, pp. 498–510, Nov. 2005.

[15] T. Jiang and S. Luo, "3D Integration-Present and Future," in *10th Electronics Packaging Technology Conference*, Dec. 2008, pp. 373 –378.

[16] R. Anigundi *et al.*, "Architecture Design Exploration of Three-Dimensional (3D) Integrated DRAM," in *Quality of Electronic Design*, March 2009, pp. 86–90.

[17] P. Garrou, C. Bower, and P. Ramm, *Handbook of 3D Integration: Volumes 1 and 2 - Technology and Applications of 3D Integrated Circuits*. Weinheim, Germany: John Wiley & Sons, 2008.

[18] R. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1214–1224, June 2006.

[19] J. Knickerbocker *et al.*, "2.5D and 3D Technology Challenges and Test Vehicle Demonstrations," in *IEEE 62nd Electronic Components and Technology Conference*, May 2012, pp. 1068–1076.

[20] C. Tan, R. Gutmann, and L. Reif, *Wafer Level 3-D ICs Process Technology*, ser. Integrated Circuits and Systems. Dordrecht, the Netherlands: Springer, 2009.

[21] S. Yoshimi *et al.*, "Development of TSV Interposer with 300 mm Wafer for 3D Packaging," in *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS*, April 2013, pp. 1–5.

[22] J. Fan and C. S. Tan, *Low Temperature Wafer-Level Metal Thermo-Compression Bonding Technology for 3D Integration, Metallurgy - Advances in Materials and Processes*, Dr. Yogiraj Pardhi, Ed., 2012.

[23] G. H. Loh, Y. Xie, and B. Black, "Processor Design in 3D Die-Stacking Technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, May 2007.

[24] K. Puttaswamy and G. Loh, "3D-Integrated SRAM Components for High-Performance Microprocessors," *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369 –1381, Oct. 2009.

[25] Y.-F. Tsai *et al.*, "Design Space Exploration for 3-D Cache," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 444 –455, April 2008.

[26] T. Thorolfsson *et al.*, "Comparative Analysis of Two 3D Integration Implementations of a SAR Processor," in *IEEE International Conference on 3D System Integration*, Sept. 2009, pp. 1–4.

[27] G. Smith *et al.*, "Yield Considerations in the Choice of 3D Technology," in *International Symposium on Semiconductor Manufacturing*, Oct. 2007, pp. 1–3.

[28] A.J. van Roosmalen, "ETP Nanoelectronics," 2007.

[29] M. Ooishi, "Vertical Stacking to Redefine Chip Design," March 2007. [Online]. Available: http://archive.today/jLeDT#selection-193.0-193.14

[30] W. A. Wulf and S. A. McKee, "Hitting the Memory Wall: Implications of the Obvious," *SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, March 1995.

[31] G. H. Loh, "3D-Stacked Memory Architectures for Multi-core Processors," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, June 2008, pp. 453–464.

[32] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, Jan. 2010.

[33] Cadence, "3D ICs with TSVs Design Challenges and Requirements," 2011. [Online]. Available: https://www.cadence.com/rl/resources/white_papers/3dic_wp.pdf

[34] J. Davis *et al.*, "Interconnect Limits on Gigascale Integration (GSI) in the 21st Century," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 305–324, March 2001.

[35] W. K. Huang, Y.-N. Shen, and F. Lombardi, "New Approaches for the Repairs of Memories with Redundancy by Row/Column Deletion for Yield Enhancement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 3, pp. 323–328, March 1990.

[36] I. Kim *et al.*, "Built in Self Repair for Embedded High Density SRAM," in *International Test Conference*, Oct. 1998, pp. 1112–1119.

[37] R. Adams, *High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test*, ser. Frontiers in Electronic Testing. Springer, 2003.

[38] M. Taouil and S. Hamdioui, "Layer Redundancy Based Yield Improvement for 3D Wafer-to-Wafer Stacked Memories." IEEE Computer Society, May 2011, pp. 45–50.

[39] L. Jiang, R. Ye, and Q. Xu, "Yield Enhancement for 3D-Stacked Memory by Redundancy Sharing Across Dies," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2010, pp. 230–234.

[40] Tezzaron, "3D-ICs and Integrated Circuit Security," Feb. 2008. [Online]. Available: http://www.tezzaron.com/media/3D-ICs_and_Integrated_Circuit_Security.pdf

[41] D. Milojevic *et al.*, "Pathfinding: A Design Methodology for Fast Exploration and Optimisation of 3D-stacked Integrated Circuits," in *International Symposium on System-on-Chip*, Oct. 2009, pp. 118–123.

[42] K. Athikulwongse *et al.*, "Stress-Driven 3D-IC Placement with TSV Keep-Out Zone and Regularity Study," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2010, pp. 669–674.

[43] Y. Yang *et al.*, "Through-Si-via (TSV) Keep-Out-Zone (KOZ) in SOI Photonics Interposer: A Study of the Impact of TSV-Induced Stress on Si Ring Resonators," *IEEE Photonics Journal*, vol. 5, no. 6, pp. 1–12, Dec. 2013.

[44] K. Tu and T. Tian, "Metallurgical Challenges in Microelectronic 3D IC Packaging Technology for Future Consumer Electronic Products," *Science China Technological Sciences*, vol. 56, no. 7, pp. 1740–1748, May 2013.

[45] L. Zhang *et al.*, "Thermal Characterization of TSV Array as Heat Removal Element in 3D IC Stacking," in *IEEE 14th Electronics Packaging Technology Conference*, Dec. 2012, pp. 153–156.

[46] K. Chakrabarty *et al.*, "TSV Defects and TSV-Induced Circuit Failures: The Third Dimension in Test and Design-for-Test," in *Reliability Physics Symposium (IRPS), 2012 IEEE International*, April 2012, pp. 5F.1.1–5F.1.12.

[47] S. Kannan, B. Kim, and B. Ahn, "Fault Modeling and Multi-Tone Dither Scheme for Testing 3D TSV Defects," *Journal of Electronic Testing*, vol. 28, no. 1, pp. 39–51, Feb. 2012.

[48] F. Ye and K. Chakrabarty, "TSV Open Defects in 3D Integrated Circuits: Characterization, Test, and Optimal Spare Allocation," in *49th ACM/EDAC/IEEE Design Automation Conference*, June 2012, pp. 1024–1030.

[49] C.-W. Kuo and H.-Y. Tsai, "Thermal Stress Analysis and Failure Mechanisms for Through Silicon Via Array," in *13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, May 2012, pp. 202–206.

[50] Xi Liu and Chen, Q. and Dixit, Pradeep and Chatterjee, R. and Tummala, R.R. and Sitaraman, S.K., "Failure Mechanisms and Optimum Design for Electroplated Copper Through-Silicon Vias (TSV)," in *59th Electronic Components and Technology Conference*, May 2009, pp. 624–629.

[51] M. Jung *et al.*, "Full-Chip Through-Silicon-Via Interfacial Crack Analysis and Optimization for 3D IC," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2011, pp. 563–570.

[52] A. Papanikolaou, D. Soudris, and R. Radojcic, *Three Dimensional System Integration*. Springer US, 2011.

[53] A. Engin and S. Narasimhan, "Modeling of Crosstalk in Through Silicon Vias," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 1, pp. 149–158, Feb. 2013.

[54] E. Beyne and I. de Wolf. (2013, July) Failure Analysis for 3D TSV Systems. [Online]. Available: http://www.sematech.org/meetings/archives/3d/10124/pres/Beyne.pdf

[55] E.J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs," in *Design, Automation Test in Europe Conference Exhibition*, March 2010, pp. 1689–1694.

[56] D. Jung *et al.*, "Disconnection failure model and analysis of TSV-based 3D ICs," in *IEEE Electrical Design of Advanced Packaging and Systems Symposium*, Dec. 2012, pp. 164–167.

[57] A.-C. Hsieh and T. Hwang, "TSV Redundancy: Architecture and Design Issues in 3-D IC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 4, pp. 711–722, 2012.

[58] J. Jung *et al.*, "Cost-Effective TSV Redundancy Configuration," in *IEEE/IFIP 20th International Conference on VLSI and System-on-Chip*, Oct. 2012, pp. 263–266.

[59] P. C. Chew *et al.*, "Through Silicon Via (TSV) Redundancy - a High Reliability, Networking Product Perspective," in *14th International Conference on Electronic Materials and Packaging*, Dec. 2012, pp. 1–5.

[60] M. Taouil, S. Hamdioui, and E. Marinissen, "Quality versus Cost Analysis for 3D Stacked ICs," in *IEEE 32nd VLSI Test Symposium*, April 2014, pp. 1–6.

[61] K. Smith *et al.*, "Evaluation of TSV and Micro-Bump Probing for Wide I/O Testing," in *IEEE International Test Conference*, Sept. 2011, pp. 1–10.

[62] H.-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits," *IEEE Design & Test of Computers*, vol. 26, no. 5, pp. 26–35, Sept. 2009.

[63] IEEE Computer Society and Test Technology Technical Council. (2014) IEEE 3D-Test Working Group (3DT-WG). [Online]. Available: http://grouper.ieee.org/groups/3Dtest/

[64] S. Deutsch *et al.*, "TSV Stress-Aware ATPG for 3D Stacked ICs," in *IEEE 21st Asian Test Symposium*, Nov. 2012, pp. 31–36.

[65] A. Ikeda *et al.*, "Design and Measurements of Test Element Group Wafer Thinned to 10 μm for 3D System in Package," in *The International Conference on Microelectronic Test Structures*, March 2004, pp. 161–164.

[66] J. H. Lau, "Supply Chains for High-Volume Manufacturing of 3D IC Integration," July 2013. [Online]. Available: http://www.chipscalereview.com/tech_monthly/csrtm-1112-supply-chains.php

[67] H. Ehrenberg and B. Russell, "IEEE Std 1581- A standardized Test Access Methodology for Memory Devices," in *IEEE International Test Conference*, Sept. 2011, pp. 1–9.

[68] M. Taouil *et al.*, "On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *International Test Conference*, Nov. 2010, pp. 1–10.

[69] M. Taouil *et al.*, "Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching," *Submitted to ACM Transactions on Design Automation of Electronic Systems*, 2014.

[70] M. Taouil and S. Hamdioui, "Yield Improvement for 3D Wafer-to-Wafer Stacked Memories." *Journal of Electronic Testing*, vol. 28, no. 4, pp. 523–534, 2012.

[71] M. Lefter *et al.*, "Is TSV-based 3D Integration Suitable for Inter-Die Memory Repair?" in *Design, Automation Test in Europe Conference Exhibition*, March 2013, pp. 1251–1254.

[72] M. Taouil and S. Hamdioui, "On Optimizing Test Cost for Wafer-to-Wafer 3D-Stacked ICs," in *7th International Conference on Design Technology of Integrated Systems in Nanoscale Era*, May 2012, pp. 1–6.

[73] M. Taouil *et al.*, "Test Cost Analysis for 3D Die-to-Wafer Stacking," in *19th IEEE Asian Test Symposium*, Dec. 2010, pp. 435–441.

[74] M. Taouil, S. Hamdioui, and E. Marinissen, "How Significant will be the Test Cost Share for 3D Die-to-Wafer Stacked-ICs?" in *6th International Conference on Design Technology of Integrated Systems in Nanoscale Era*, April 2011, pp. 1–6.

[75] M. Taouil *et al.*, "Test Impact on the Overall Die-to-Wafer 3D Stacked IC Cost," *Journal of Electronic Testing*, vol. 28, no. 1, pp. 15–25, Feb. 2012.

[76] M. Taouil, S. Hamdioui, and E. J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," in *6th International Design and Test Workshop*, Dec. 2011, pp. 24–29.

[77] M. Taouil *et al.*, "Using 3D-COSTAR for 2.5D Test Cost Optimization," in *IEEE International 3D Systems Integration Conference*, Oct. 2013, pp. 1–8.

[78] M. Taouil *et al.*, "Impact of Mid-Bond Testing in 3D stacked ICs," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, Oct. 2013, pp. 178–183.

[79] M. Taouil and S. Hamdioui, "Stacking Order Impact on Overall 3D Die-to-Wafer Stacked-IC Cost," in *14th International Symposium on Design and Diagnostics of Electronic Circuits Systems*, April 2011, pp. 335–340.

[80] M. Taouil, M. Lefter, and S. Hamdioui, "Exploring Test Opportunities for Memory and Interconnects in 3D ICs," in *8th International Design and Test Symposium*, Dec. 2013, pp. 1–6.

[81] M. Taouil *et al.*, "Interconnect Test for 3D Stacked Memory-on-Logic," in *Design, Automation and Test in Europe Conference and Exhibition*, March 2014, pp. 1–6.

[82] M. Taouil *et al.*, "Post-Bond Interconnect Test and Diagnosis for 3D Memory Stacked on Logic," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2014.

[83] C. Ferri, S. Reda, and R. I. Bahar, "Parametric Yield Management for 3D ICs: Models and Strategies for Improvement," *Journal on Emerging Technologies in Computing Systems*, vol. 4, no. 4, pp. 19:1–19:22, Nov. 2008.

[84] S. Reda, G. Smith, and L. Smith, "Maximizing the Functional Yield of Wafer-to-Wafer 3-D Integration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 9, pp. 1357–1362, 2009.

[85] J. Verbree *et al.*, "On The Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking," in *15th IEEE European Test Symposium*, May 2010, pp. 36–41.

[86] E. Singh, "Exploiting Rotational Symmetries for Improved Stacked Yields in W2W 3D-SICs," in *IEEE 29th VLSI Test Symposium*, May 2011, pp. 32–37.

[87] E. Singh, "Impact of Radial Defect Clustering on 3D Stacked IC Yield from Wafer to Wafer Stacking," in *IEEE International Test Conference*, Nov. 2012, pp. 1–7.

[88] E. Singh, "Analytical Modeling of 3D Stacked IC Yield from Wafer to Wafer Stacking with Radial Defect Clustering," in *27th International Conference on VLSI Design and 13th International Conference on Embedded Systems*, Jan. 2014, pp. 26–31.

[89] C.-W. Chou, Y.-J. Huang, and J.-F. Li, "Yield-Enhancement Techniques for 3D Random Access Memories," in *International Symposium on VLSI Design Automation and Test*, April 2010, pp. 104–107.

[90] Y.-F. Chou, D.-M. Kwai, and C.-W. Wu, "Yield Enhancement by Bad-Die Recycling and Stacking with Though-Silicon Vias," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1346–1356, Aug. 2011.

[91] C.-W. Wu, S.-K. Lu, and J.-F. Li, "On Test and Repair of 3D Random Access Memory," in *17th Asia and South Pacific Design Automation Conference*, Jan. 2012, pp. 744–749.

[92] S.-K. Lu, T.-W. Chang, and H.-Y. Hsu, "Yield Enhancement Techniques for 3-Dimensional Random Access Memories," *Microelectronics Reliability*, vol. 52, no. 6, pp. 1065 – 1070, 2012.

[93] B. Zhang, B. Li and V.D. Agrawal, "Yield Analysis of a Novel Wafer Manipulation Method in 3D stacking," in *IEEE International 3D Systems Integration Conference*, Oct. 2013, pp. 1–8.

[94] C. Hawkins *et al.*, "Defect Classes - An Overdue Paradigm for CMOS IC Testing," in *International Test Conference*, Oct. 1994, pp. 413–425.

[95] E. Marinissen *et al.*, "Creating Value through Test," in *Design, Automation and Test in Europe Conference and Exhibition*, 2003, pp. 402–407.

[96] N. Mukherjee, "Targeting Zero DPPM - Can we ever get there?" in *IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, Oct. 2008, pp. 163–163.

[97] J. Sousa *et al.*, "Fault Modeling and Defect Level Projections in Digital ICs," in *European Design and Test Conference, The European Conference on Design Automation, European Test Conference, The European Event in ASIC Design*, Feb. 1994, pp. 436–442.

[98] J. Shumaker, M. Lauderdale, and K. Shults, "A Process for Optimizing Probe and Final Test through Parameter Matching," in *IEEE International Symposium on Semiconductor Manufacturing*, Sept. 2003, pp. 431–434.

[99] A. Walker, "A Manufacturing Cost Model for 3-D Monolithic Memory Integrated Circuits," *IEEE Transaction on Semiconductor Manufacturing*, vol. 22, no. 2, pp. 268–275, May 2009.

[100] P. Mercier *et al.*, "Yield and Cost Modeling for 3D Chip Stack Technologies," in *Custom Integrated Circuits Conference*, Sept. 2006, pp. 357–360.

[101] Y. Chen *et al.*, "Cost-Effective Integration of Three-Dimensional (3D) ICs Emphasizing Testing Cost Analysis," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2010, pp. 471–476.

[102] D. Velenis *et al.*, "Impact of 3D Design Choices on Manufacturing Cost," in *IEEE International Conference on 3D System Integration*, Sept. 2009, pp. 1–5.

[103] X. Dong and Y. Xie, "System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs)," in *Asia and South Pacific Design Automation Conference*, Jan. 2009, pp. 234–241.

[104] D. Velenis, E. J. Marinissen, and E. Beyne, "Cost Effectiveness of 3D Integration Options," in *IEEE International Conference on 3D System Integration*, Nov. 2010.

[105] D. Velenis *et al.*, "Cost Comparison between 3D and 2.5D Integration," in *Electronic System-Integration Technology Conference*, Sept. 2012, pp. 1–4.

[106] D. Velenis *et al.*, "Si Interposer Build-Up Options and Impact on 3D System Cost," in *IEEE International 3D Systems Integration Conference*, Oct. 2013, pp. 1–5.

[107] C.-C. Chan, Y.-T. Yu, and I.-R. Jiang, "3DICE: 3D IC Cost Evaluation Based on Fast Tier Number Estimation," in *12th International Symposium on Quality Electronic Design*, March 2011, pp. 1–6.

[108] C. Zhang and G. Sun, "Fabrication Cost Analysis for 2D, 2.5D, and 3D IC Designs," in *IEEE International 3D Systems Integration Conference*, Jan. 2012, pp. 1–4.

[109] Y.-W. Chou *et al.*, "Cost Modeling and Analysis for Interposer-Based Three-Dimensional IC," in *IEEE 30th VLSI Test Symposium*, April 2012, pp. 108–113.

[110] M. Agrawal and K. Chakrabarty, "Test-Cost Optimization and Test-Flow Selection for 3D-stacked ICs," in *IEEE 31st VLSI Test Symposium*, April 2013, pp. 1–6.

[111] H. Butt, "ASIC DFT Techniques and Benefits," in *Sixth Annual IEEE International ASIC Conference and Exhibit*, Sept. 1993, pp. 46–53.

[112] E. Marinissen, J. Verbree, and M. Konijnenburg, "A Structured and Scalable Test Access Architecture for TSV-based 3D Stacked ICs," in *28th VLSI Test Symposium*, April 2014, pp. 269–274.

[113] D. Lewis and H. Lee, "A scan-island based design enabling pre-bond testability in die-stacked microprocessors," in *IEEE International Test Conference*, Oct. 2007, pp. 1–8.

[114] IEEE, "IEEE Standard Test Access Port and Boundary Scan Architecture," *IEEE Std 1149.1-2001*, pp. 1–212, July 2001.

[115] IEEE, "IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits," *IEEE Std 1500-2005*, pp. 1–117, 2005.

[116] C.-C. Chi *et al.*, "Post-bond Testing of 2.5D-SICs and 3D-SICs Containing a Passive Silicon Interposer Base," in *International Test Conference*, Sept. 2011, pp. 1–10.

[117] K. Saban. (2012, Dec.) Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency. [Online]. Available: http://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf

[118] C.-C. Chi *et al.*, "DfT Architecture for 3D-SICs with Multiple Towers," in *16th IEEE European Test Symposium*, May 2011, pp. 51–56.

[119] JEDEC, Global Standards for the Microelectronic Industry. (2011) JEDEC, Wide I/O Single Data Rate (Wide I/O SDR). [Online]. Available: http://www.jedec.org/standards-documents/results/jesd229

[120] S. Deutsch *et al.*, "DfT Architecture and ATPG for Interconnect Tests of JEDEC Wide-I/O Memory-on-Logic Die Stacks," in *IEEE International Test Conference*, Nov. 2012, pp. 1–10.

[121] A. Papanikolaou, D. Soudris, and R. Radojcic, *Three Dimensional System Integration: IC Stacking Process and Design*, ser. SpringerLink: Bücher. Springer, 2010.

[122] C. Liu *et al.*, "Full-chip TSV-to-TSV Coupling Analysis and Optimization in 3D IC," in *48th ACM/EDAC/IEEE Design Automation Conference*, June 2011, pp. 783–788.

[123] V. Pasca, L. Anghel, and M. Benabdenbi, "Configurable Thru-Silicon-Via interconnect Built-In Self-Test and diagnosis," in *12th IEEE Latin-American Test Workshop*, March 2011, pp. 1–6.

[124] Yu-Jen Huang and Jin-Fu Li and Che-Wei Chou, "Post-Bond Test Techniques for TSVs with Crosstalk Faults in 3D ICs," in *VLSI Design, Automation, and Test*, April 2012, pp. 1–4.

[125] L. Chen, X. Bai, and S. Dey, "Testing for Interconnect Crosstalk Defects Using On-Chip Embedded Processor Cores," in *Design Automation Conference*, 2001, pp. 317–322.

Appendix A

Publications - Yield Improvement

This chapter presents the publications on yield improvement. The following papers are included:

A1: **M. Taouil**, S. Hamdioui, J. Verbree, and E.J. Marinissen, "On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *International Test Conference (ITC)*, Austin, TX, USA, Nov. 2010, pp. 1-10.

A2: **M. Taouil**, S. Hamdioui and E.J. Marinissen, "Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching," *submitted to ACM Transactions on Design Automation of Electronic Systems (TODAES), pp. 1–24, 2014*.

A3: **M. Taouil** and S. Hamdioui, "Layer Redundancy Based Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *European Test Symposium (ETS)*, Trondheim, Norway, May 2011, pp. 45–50.

A4: **M. Taouil** and S. Hamdioui, "Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 4, pp. 523-534, Aug. 2012.

A5: M. Lefter, G.R. Voicu, **M. Taouil**, M. Enachescu, S. Hamdioui, and S.D. Cotofana, "Is TSV-based 3D Integration Suitable for Inter-die Memory Repair?" *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, March 2013, pp. 1251-1254.

On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs

Mottaqiallah Taouil[1] Said Hamdioui[1]
[1]Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, S.Hamdioui}@tudelft.nl

Jouke Verbree[1,2] Erik Jan Marinissen[2]
[2]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
{jouke.verbree, erik.jan.marinissen}@imec.be

Abstract

Three-Dimensional Stacked IC (3D-SIC) is an emerging technology that provides heterogeneous integration, higher performance, and lower power consumption compared to planar ICs. Fabricating these 3D-SICs using Wafer-to-Wafer (W2W) stacking has several advantages including: high throughput, thin wafer and small die handling, and high TSV density. However, W2W stacking suffers from low compound yield. This paper investigates various matching processes by using different wafer matching criteria in order to maximize the compound yield. It first establishes a framework covering different matching processes and wafer matching criteria for both replenished and non-replenished wafer repositories. Thereafter, a subset of the framework is analyzed. The simulation results show that the compound yield not only depends on the number of stacked dies, die yield, and repository size, but it also strongly depends on the used matching process and the wafer matching criteria. Moreover, by choosing an appropriate wafer matching scenario (e.g., wafer matching process, criterion etc.), the compound yield can be improved up to 13.4% relative to random W2W stacking.

Keywords: *3D integration, wafer matching, matching criteria, compound yield, wafer-to-wafer stacking*

I. Introduction

The ability to create *Three-Dimensional Stacked Integrated Circuits (3D-SICs)* alleviates or even eliminates various existing problems in planar ICs. A 3D-SIC consists of multiple stacked planar dies, fabricated in a conventional process augmented by new Through Silicon Via (TSV) process steps, which electrically connect the planar wafers in the vertical direction. An efficient partitioning of IP cores among the stacked dies reduces the need for long wires and is thus able to reduce the wire delay, as well as the power dissipation [1], [2]. Heterogeneous integration is a promising concept for 3D-SICs, since each layer can be manufactured with different technology and optimized for speed or area. This affects the yield, performance, and lithography cost positively. Furthermore, miniaturization of the physical sizes of stacked dies reduces the footprint size and volume area, which in turn increases the package density. Examples of 3D-SICs include 3D CMOS sensors [3], 3D FPGAs [3], 3D processors [4], 3D cache and memory [5], [6], and combined stacks of memories and processors [3], [7].

Tiers are stacked at the die or the wafer level and can be stacked based on Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) or Die-to-Die (D2D) bonding. In W2W bonding, complete wafers are stacked and bonded together. One of the benefits of W2W stacking is the high manufacturing throughput due to single wafer alignment [8]. High alignment accuracy can also be applied to D2W and D2D, but it negatively affects the throughput due to many dies that have to be aligned [8]. However, the yield loss for 3D-SICs is one of the major bottlenecks that must be overcome for 3D technology to make it a lucrative business [9]. The major limitations of W2W stacking is the rapid compound yield decrease, as the number of layers in the stack increases. The compound yield can be improved by *wafer matching*, initially introduced by Smith et al. [10]. In wafer matching, a software algorithm keeps track of the fault map of each wafer. The algorithm matches wafer pairs that contain the same or similar fault maps. This increases the 3D compound yield over randomly stacked wafers. More elaborated studies of wafer matching are presented in [11], [12]. Nevertheless, all the published work considered wafer matching with *static* repositories, i.e., after wafer selection, the repositories are not replenished unless they are empty. In addition, these papers focused *only* on matching of the *good dies* from the bottom layer with the *good dies* from the top layer. However, this could also be the matching of the *faulty* dies instead of the good dies.

978-1-4244-7207-9/10/$26.00

In this paper, the impact of *replenished* repositories on the compound yield by using different wafer matching criteria is investigated. In this case, when a wafer is selected from a repository, its empty spot is directly replenished with a new one. This keeps the size of the *running* repository constant over time. The main contributions of this paper are:

- A new framework that covers all matching processes and wafer matching criteria for both *static* and *running* repositories.
- The illustration of the impact of several matching processes and wafer matching criteria on the compound yield of 3D-SICs.
- The demonstration of the impact of running repositories on the 3D-SIC compound yield.
- A comparison between the yield benefits gained from static and running repositories over random stacking.

The remainder of the paper is organized as follows. Section II provides an overview of the prior work in the area of wafer matching. Section III introduces the framework for wafer matching and defines the focus of this paper. Section IV describes the wafer matching scenarios to be experimented with in this work. Section V presents the simulation results and the comparison to the related work. Section VI concludes the paper.

II. Related Prior Work

Improving the yield for 3D circuits based on wafer matching was initially introduced by Smith et al. [10], where the authors compared the yield improvement of a single die SoC, by mapping it into a 3D-SIC with two equal sized layers. The yield improvement is both simulated for D2W and W2W stacking. In the W2W stacking case, a software matching algorithm is used to select pair-wise the best wafers from two repositories with a size of 25 each. The wafer fault map is based on a random generation.

The concept of W2W matching introduced by Smith [10] is further generalized by Reda et al. [11]. The paper formulates the W2W matching problem and proves it to be $\mathcal{NP}$-hard. Several matching processes and wafer matching algorithms are investigated, including the optimal hard one. In [12], Verbree et al. define a mathematical model for wafer matching; the model has some practical limitations, but nevertheless it gives a good indication of the yield improvements. The authors include wafer matching simulations for a greedy algorithm that address the limitations. In addition, the authors justify their pre-bond test cost required for wafer matching.

In [13], Ferri et al. used wafer matching to increase the *parametric yield* of a two layered D2W stacked 3D-SIC. Only functional dies are considered in this case to produce an optimal binning; i.e., maximize the fastest speed bins and minimize the slowest ones. Wafer matching is then used to combine and improve the 3D parametric yield by including the process variation of both layers in a D2W stacking approach. The authors were able to increase the number of 3D-SICs in the fastest speed bins as well as simultaneously reducing the number of slow 3D-SICs.

All the related previous work considered *static* repositories and used a *single* wafer matching criterion.

III. Wafer Matching Framework

As it has already been mentioned, W2W stacking provides the highest manufacturing throughput and is suitable for wafers with identical die sizes and/or small die sizes. However, it suffers from lower compound yield, as the stacking of bad dies on good dies and vice versa can not be avoided. *Wafer matching* can be performed on repositories of wafers in order to find out the best wafer combinations that would result in higher yield, given that the wafers were tested before the bonding. This section defines a framework for all possible *wafer matching scenarios* for 3D W2W stacked ICs; a wafer matching scenario combines different aspects at a time: (a) *Static or running repositories*, (b) *Wafer matching process*; e.g., how many wafer and/or layers are considered at each step, and (c) *Wafer matching criterion*; e.g., select the matching based on the good matched dies.

In the rest of the section, first the problem of W2W 3D-SICs is defined. Then, the aspects of wafer matching scenarios are addressed. Thereafter, the wafer matching framework is given.

A. W2W 3D-SIC Problem

The problem of W2W 3D-SICs can be defined as follows: Given, (a) n number of repositories each with k wafers, (b) fault maps for all the wafers (based on pre-bond testing), and (c) a production size of m 3D-SICs, the purpose is to maximize the overall compound yield for all m 3D-SICs, by selecting appropriate wafers for the n-layer 3D-SICs from the repositories. Figure 1 shows two freedom degrees to create 3D stacks. The vertical direction considers the wafers and the selection freedom here is the number of wafers that are selected to be stacked simultaneously; this can be either one wafer at a time (*Wafer-by-Wafer*) or k wafers at a time (*All-Wafers*). The horizontal direction shows the freedom selection from the number of layers that are considered simultaneously for stacking; this can be either two layers at a time (*Layer-by-Layer*) or n layers at a time (*All-Layers*).

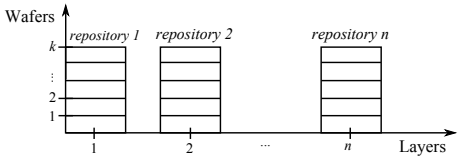


Fig. 1. Wafer vs layers in 3D stacking

B. Static Versus Running Repositories

Wafer matching can be considered as a time consuming process when the objective is to obtain the global compound yield for a production size m; m can be in the order of thousands or millions. To split up and divide the problem, a fixed number of k (usually $k << m$) wafers per repository can be considered and matched at a time. Depending on either a repository is replenished immediately (after a wafer is removed from it for matching and stacking) or not, two classes can be defined:

- *Static repositories*: From each repository k wafers are selected and processed before considering the next group of k wafers. The procedure stops after m/k steps.
- *Running repositories*. Each repository is immediately replenished with a new wafer each time a wafer is selected. The procedure stops after m wafers are processed.

The freedom to select wafers from static repositories reduces over time, since the repositories become more and more empty. Running repositories, however, provide always the full repository (of size k) to select from; this improves the effectiveness of wafer matching as compared with static repositories. The downside of running repositories is that unattractive wafers may remain in the repository for many iterations, occupying space, and in effect reducing the size of the repository in the long run. We call this effect, the repository *pollution*.

Another difference between static and running repositories is the actual implementation. Static repositories map fairly well onto a production line, where basically the repositories are the wafer containers that move from one machine in the production line to the next. With running repositories, a container would need to go back and forth between the bonding machine and the wafer production line to be replenished, before a new selection is made. Clearly, this is impractical, and therefore we suggest using two containers. One to select from, and one acts as a wafer source to replenish the first one at the bonding machine. This, however, reduces the effective capacity of the bonding machine as both containers are in the machine, yet only one is used to select a wafer from.

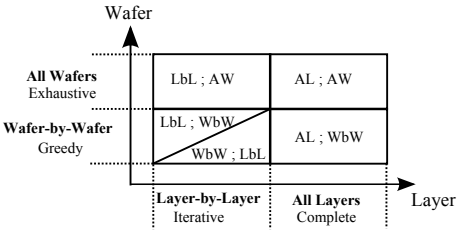


Fig. 2. Framework of matching processes

C. Matching Process

The matching process defines the step-by-step process to be followed in order to realize wafer matching. The matching process, therefore, determines the *number of repositories* and the *number of wafers* that are considered at a time.

Depending on the number of involved repositories (see also Figure 1), two cases are distinguished:

- *Layer-by-Layer (LbL)*: Initially, the first two repositories are selected for wafer matching. In each additional step, only one additional repository is used during matching. Hence, this is an *iterative* process in terms of the number of involved layers.
- *All-Layers (AL)*: In each step of the wafer matching process, all repositories are used at once. As every wafer in every repository is taken into account, this process is labeled *complete*.

In a similar way, depending on the number of wafers involved in each step of the matching process, two cases can be distinguished:

- *Wafer-by-Wafer (WbW)*: In each step of the wafer matching process, the best wafers contributing to the possible match are selected. Only one wafer from each repository is involved in the matching process, with no regard to the remaining wafers in those repositories. Thus, this process is regarded as *greedy*.
- *All-Wafers (AW)*: In each step of the wafer matching process, all wafers from all involved repositories are matched. As the process considers all possible outcomes for all k wafers to be matched, this process is considered to be *exhaustive*.

The above combinations result into five possible wafer matching processes, as shown in Figure 2.

- LbL;WbW: The matching process steps are iterative over the repositories. In each iteration step, only two repositories are considered. First, the best wafer pair for the first two repositories (each with k wafers) is selected. Then, the step is repeated $(k-1)$ times on these two repositories. Thereafter, the process is repeated on the rest of the repositories one by one.

Note that in each step, the size of the repositories are reduced by one.

- WbW;LbL: The matching process steps are iterative over the wafers. In each step, a single wafer is selected iteratively from each repository to form the 3D-SIC. The difference between LbL;WbW and WbW;LbL is the reversed loop order of visiting the repositories and the wafer selections within a repository.
- LbL;AW: Similar to LbL;WbW, the matching process iteratively considers two repositories at a time, but in this case, all wafers from the two repositories under consideration are matched. Note that, this matching process is only applicable to static repositories, since running repositories are replenished, each time a wafer is selected from them. The difference between LbL;WbW and LbL;AW is that LbL;AW provides an *exhaustive* solution within the LbL process, while LbL;WbW selects the wafers one by one in a *greedy* way.
- AL;WbW: The matching process considers all repositories simultaneously in each matching step, and selects the best matching combination of n wafers along the repositories. The same step is repeated over time. In the case of static repositories, the matching of n wafers along the repositories is performed, first with k wafers and in the second step with $k-1$ wafers, etc. In case of running repositories, the matching considers always k wafers from each repository.
- AL;AW: This is similar to AL;WbW, but here, all k wafers from each repository are matched simultaneously. Note that, this matching process is only applicable to static repositories.

It is worth noting that for the LbL processes, an additional freedom can be defined for the traversal order for the repositories. The number of freedom possibilities to step over the repositories equals to $\binom{n}{2} \cdot (n-2)! = \frac{n!}{2}$; the first term of the equation represents the number of possibilities to select the first two repositories out of n, while the second term $(n-2)!$ presents the number of combinations of the remaining repositories.

D. Matching Criteria

The matching processes select wafers based on certain criteria; e.g., best good dies. Each criterion is orthogonal with respect to the process. Based on the fact that each wafer consists of both good and bad dies and that the purpose of the wafer matching is to maximize the compound yield, one can define three possible criteria: (a) maximize the matching good dies, (b) maximize the matching faulty dies, and (c) minimize the matching between good and bad dies. The criteria are defined as follows:

- *Max(MG)*. This criterion selects the best wafer pair combinations based on the maximum *Matched Good (MG)* dies. All the published work so far regarding wafer matching considers only this criterion.
- *Max(MF)*. The best wafer pair combinations is selected based on maximum *Matched Faulty (MF)* dies.
- *Min(UF)*. This criterion selects the best wafer pair combinations based on minimum *Unmatched Faulty (UF)* dies. The objective is to increase the compound yield by minimizing faulty dies that land on good dies and vice versa.

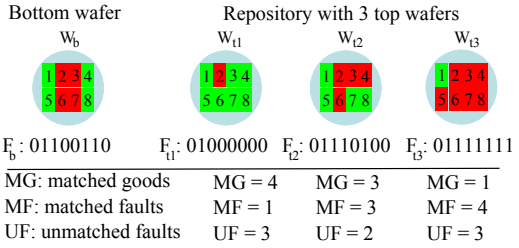


Fig. 3. Wafer matching criteria

All the above criteria produce the same result in terms of compound yield, in case the wafer matching process is exhaustive (AW process) for static repositories. For the greedy wafer matching processes (WbW), it is evident that different criteria lead to different results due to the greediness of the algorithm. For running repositories, the criteria lead to different compound yields, as will be explained next.

In order to provide more insight into the impact of the above criteria on wafer selection, refer to the example shown in Figure 3, which considers a bottom wafer W_b and three potential top wafers (W_{t1}, W_{t2}, W_{t3}), each with its own fault map. The fault map of each wafer is denoted by F and contains a sequence of 0s (good dies) and 1s (bad dies) ordered according to the indices of the dies on the wafer; e.g., the bottom wafer has $F_b = 01100110$, since the dies 2, 3, 6 and 7 are faulty. The bottom table in the figure lists the value of the different criteria for the three matching possibilities; e.g., for matching W_b-W_{t1}, the number of matched good dies is MG = 4 (which are dies 1, 4, 5, 8). The figure clearly shows that depending on the criterion, different top wafers will be selected; e.g., if max(MG) is considered, then W_{t1} will be selected. However, if the max(MF) is the criterion, then W_{t3} is the best match. The criteria can be mathematically formulated. Let the function $G(F_i)$ be the number of faulty dies in the wafer with fault map F_i. Then,

$$Max(MG) = max(\forall_{i,j}, G(\bar{F_i}\&\bar{F_j})) \quad (1)$$

$$Max(UF) = min(\forall_{i,j}, G(F_i \oplus F_j)) \quad (2)$$

$$Max(MF) = max(\forall_{i,j}, G(F_i\&F_j)) \quad (3)$$

Here, $0 \leq i, j \leq k$, where k the repository size.

TABLE I. W2W Matching Framework

Matching process	Static repository	Running repository
LbL;WbW	yes (Greedy [12])	n.a.
WbW;LbL	yes	yes
LbL;AW	yes (IMH [11])	n.a.
AL;WbW	yes (Greedy [11])	yes
AL;AW	yes (ILP/UB [11])	n.a.

n.a. denotes not applicable

E. The W2W matching framework

The wafer matching scenario aspects discussed in the previous section can be integrated into a complete framework that covers all wafer matching scenarios as shown in Table I. The table shows the possible combinations of matching processes and repository types (e.g., static and running repositories). Each combination results in a wafer matching scenario when combined with a matching criteria. The matching scenarios considered in the previous published work are represented by their references in the table. The criteria are left out, since they are independent of the matching processes. The table shows whether for each combination between the processes and the repository types, a valid combination exists ("yes" in the table) or not ("n.a"). Going vertically down in the entries of the table, more advanced algorithms are used which in general lead to a higher compound yield at the cost of higher computational effort. Putting the previous work in the context of the framework defined in Section III, the following can be concluded. The greedy algorithm in [12] is a LbL;WbW process. It creates a sorted list based on the compound yield of all wafer combinations between two repositories. From this list, valid pairs are selected starting from the highest yield. A combination is considered invalid when at least one of the wafers of the current compound has already been taken in a previous selection. After the repositories are empty, the repository of the next layer is matched with the current temporary stacks. In [11], three matching scenarios are described. In the first scenario, a greedy algorithm is used to create a sorted list of all k^n wafer combinations; this is in fact AL;WbW process. The difference with the greedy algorithm in [12] is that in this scenario all layers are considered at the same time. The second scenario, referred to as the Iterative Heuristic Matching (IHM) algorithm, considers two repositories at a time and optimally matches them by the Hungarian algorithm. These steps are iteratively repeated by including one additional repository in each iteration. The IHM algorithm is an LbL;AW process. In the third scenario, a global optimal algorithm based on Inter Linear Programming (ILP) is used to explore the exhaustive search space and obtain the global maximum yield. The execution time reduction of ILP scenario is realized by relaxing the ILP and allowing the program variables to take fractional values; this resulted into Upper Bound (UB) scenario. The ILP and UB scenarios are both AL;AW processes.

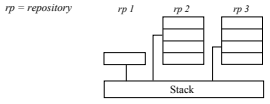


Fig. 4. Matching scenario FIFO1

From Table I we conclude that several scenarios are not explored yet, mainly the ones for running repositories. This paper explores part of this space as will be explained in the next section.

IV. Scenarios for Running Repositories

The paper focuses on the impact of running repositories on the compound yield. Different wafer matching scenarios are considered based on the WbW;LbL matching process and different matching criteria. Due to space limitation and its low time and memory complexity, WbW;LbL is the only wafer matching processes considered in this paper.

As already explained, WbW;LbL process considers only two repositories at a time; in addition, only a single wafer pair selection is performed. Based on the wafer pairs selection order, three LbL;WbW matching processes can be defined:

- FIFO1-based WbW;LbL matching process.
- FIFOn-based WbW;LbL matching process.
- Best Pair-based WbW;LbL matching process.

Note that there are 9 matching scenarios, where 9 = 1 (running repository) · 3 (matching processes) · 3 (matching criteria). These are explained next.

A. FIFO1

In the FIFO1-based matching process the wafers from the first repository are selected based on a FIFO approach, as depicted in Figure 4 for $n = 3$. The wafers from repository 1 (rp1) are selected without any freedom and matched with the best wafer from the second repository. The process iterates over all the repositories. The size of the first repository is actually irrelevant, and can be changed to one. The order in which the repositories are traversed is linear starting at repository 1 and ending at repository n. Note that for FIFO1, the pollution is not critical for the first repository, since wafers are forced to get out. The runtime complexity of FIFO1 is $O(m \cdot k \cdot (n-1)) = O(m \cdot k \cdot n)$. The worst case memory complexity is $O(n)$; this is the memory required to store the list of indices holding the positions of the selected wafers from each repository.

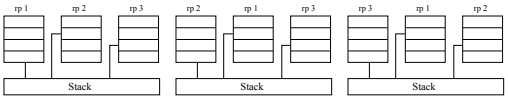


Fig. 5. Matching scenario FIFOn

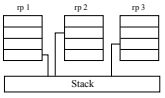


Fig. 6. Matching scenario BP

B. FIFOn

In the FIFOn-based matching process, we generalize the concept of FIFO1. This is performed by moving the FIFO-repository in a round robin fashion among all repositories as shown in Figure 5 for $n = 3$. At the left side of the figure, repository one (rp1) is used as FIFO. After an n-compound stack is created, the repository belonging to the next layer is considered to be the FIFO as shown in the middle of the figure. Here, the algorithm starts from rp2 and proceeds next with rp1 and rp3; the traversal order is written in the top part of Figure 5. For the next compound, rp3 is used as FIFO. These steps are repeated until the production size is reached. The first traversed repository is the repository that is considered as FIFO, the remaining repositories are traversed in monolithic increasing order starting at repository 1 and ending at repository n. FIFOn is able to control the pollution since it forces wafers to stay maximally $n \cdot k$ cycles in a repository. In this way, the repositories are not contaminated with bad wafers that stay for a long time in the repositories without being selected. The memory and runtime complexity for this scenario are the same as in the case for FIFO1, since it only changes the position of the FIFO-repository.

C. Best Pair (BP)

In the BP-based matching process, the wafers from the first two repositories are matched in pairs without any selection restrictions; see Figure 6 for $n = 3$. The process iteratively proceeds along the repositories until a single n-compound match is determined. Then, this process is repeated until the production size m is met. The BP matching process has more freedom in wafer selection, as compared to FIFOn, but it lacks controlling the repository pollution. The runtime complexity equals to $O(m \cdot k \cdot n + k^2) = O(mkn)$. Initially, k^2 comparisons are performed on the initial set of the first two repositories. The best pair is selected and used to search for the best matching with the rest of repositories (one by one); this requires $(n-2)*k$ comparisons. Note that after replenishing, the process will be repeated; however, now the first two repositories require only $2*k-1$ comparisons rather than k^2 since the results of the previous comparison can be reused. The memory complexity is $O(k^2+n)$, required to store all k^2 compound yield combinations between the first repositories, and to hold a list of n numbers identifying the indices of the selected wafers of each repository.

V. Simulation Results

This section presents the simulation results and analyzes the impact of the 9 wafer matching scenarios discussed in the previous section of the compound yield (i.e., FIFO1-based, FIFOn-based and BP-Based scenarios). Section V-A describes the experimental setup. Section V-B provides the impact of the running repositories, while V-C presents the impact on repository 'pollution'. Finally, the best wafer matching scenario will be selected and compared to related work in Section V-D.

A. Experimental Setup

The experiments are based on the reference process in [12]. A standard 300 mm diameter wafer is selected with an edge clearance of 3 mm. The defect density is considered to be d_0 = 0.5 defects/cm^2 and the defect clustering parameter $\alpha = 0.5$. For the reference design, the die area is assumed to be A=$50mm^2$. For this die area and wafer size, the number of Gross Dies per Wafer (GDW) approximately equals to 1278 [15]. The expected yield of the wafers can be estimated by the negative binomial formula as: $y = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$= 81.65% [16].

In our experiments, we simulate a production size $m = 25000$. Here, m is the number of produced 3D-SICs. Initially, each repository is filled up with k wafers and after selecting and stacking m-compound wafers, the wafers that are left in the repository are discarded and not included in the simulation results for two reasons.

1) First, we want to observe the impact of the running repository only.
2) Second, even if the wafers would be thrown away, their impact on the compound yield is minimal (k/m), due to a high production volume m. Actually, the matching scenarios presented in [11] and [12] for static repositories could be used to match these last k unconsidered wafers.

To measure the impact of running repositories on the compound yield (while considering repository size, wafer yield and matching criteria) as well as repository pollution, three experiments are performed:

- In experiment 1, the impact on the compound yield for different stacked number of layers n ($2 \leq n \leq 6$) for various repository sizes is examined. The reference

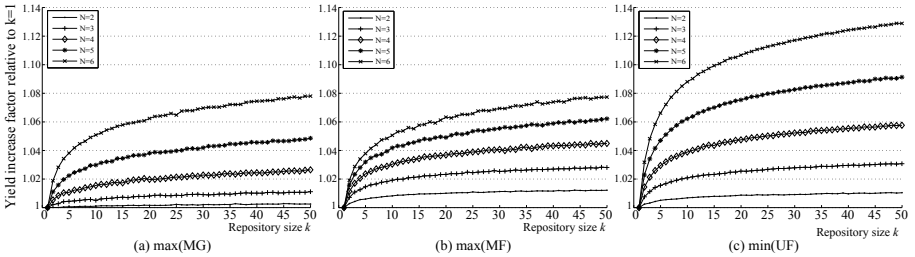


Fig. 7. Impact of n and k on compound yield for FIFO1 using the reference process

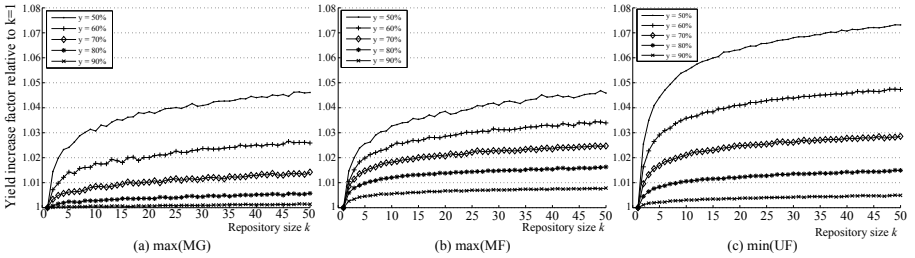


Fig. 8. Impact of wafer yield and k on the compound yield for FIFO1

process is considered and all criteria are simulated for each scenario.

- In experiment 2, we adjust the wafer yield of the reference process over a wide range to simulate the impact of the compound yield on stacked 3D-SICs. We consider a stack of two layers and vary the repository size.
- The last experiment consists of indirect measurement of the repository pollution. By plotting the compound yield for different stack sizes versus different production sizes m, we can indirectly measure the pollution that takes place and observe the effect on the compound yield. Moreover, we look at the compound yield differences between FIFO1 and FIFOn.

B. Impact of Running Repositories

Figure 7 plots the relative compound yield increase with respect to random stacking (i.e., $k = 1$) for different stacked layers n and repository sizes k for each criteria, while Figure 8 plots the relative compound yield increase with respect to wafer yield. Due to space limitation, only the simulation results for FIFO1-based matching processes are presented here. The figures clearly show that the relative compound yield increases with larger repositories and lower wafer yield, but the obtained gain stabilizes as the size of k becomes larger; the trends are similar for all criteria. It is worth noting that FIFOn-based and BP-based show similar trends as FIFO1-based wafer matching processes.

Let's now examine the impact of the matching criteria on the compound yield. Figure 7 shows that the criteria min(UF) outperforms the other two criteria for $n \geq 3$. On the other hand, Figure 8 indicates that min(UF) performs the best for wafer yields in the range of 50%-70%, while the criteria max(MF) performs the best for higher wafer yield (80% and above).

To obtain a better picture of the impact of the criteria on the compound yield, simulation for all scenarios (FIFO1, FIFOn and BP) and different criteria for a fixed repository size of $k = 50$ has been performed. Figure 9 and 10 show the results. One can conclude the following:

- In general, a higher improvement can be gained for larger stack sizes and lower wafer yield. Note that, when the stack size increases, the compound yield decreases.
- FIFOn always performs better than FIFO1 for the same conditions, especially for the criteria that relatively perform poor. This difference in performance is minimal for min(UF) criterion; this means that in this case a small pollution is taking place (see next section).
- Overall, BP scenario scores the best in terms of compound yield. Depending on the value of the wafer yield y, BP has to be combined with appropriate criterion. For wafer yield $50 \leq y \leq 70$, BP combined with Min(UF) scores the best, while for $y \geq 80$ BP combined with Max(MF) scores the best.

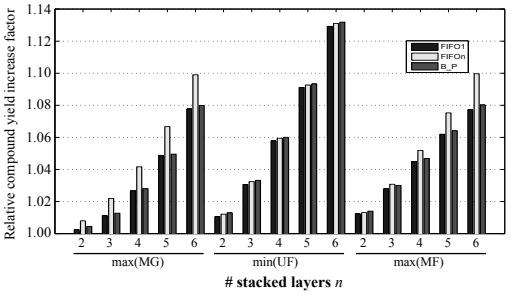


Fig. 9. Yield increase with variable n

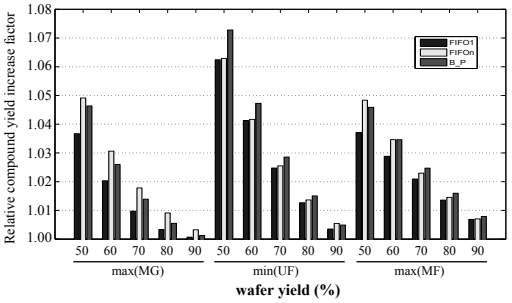


Fig. 10. Yield versus wafer yield

The above results clearly show that BP scenario outperforms both FIFO1 and FIFOn. The question is now which matching criterion has to be combined with -for a certain process- to maximize the compound yield. Table II answers this question. The table shows the criteria for different top wafer yield y_t and bottom wafer yield y_b that have be selected to achieve the highest compound yield. From the table one can conclude the following:

- When the wafer yield is low, the Max(MG) criterion should be selected. Max(MG) tries to match the good dies only and since these are in minority, the choice to select the best matching is relatively easy.
- For wafer yield in midrange values, the Min(UF) criterion performs the best. In this case, the probability of the presence of good and bad dies is similar.
- For very high wafer yield, it is most advantageous to select the Max(MF) criterion. In this case, the matching is based on faulty dies. As the faulty dies are in minority due to a high wafer yield, an overall highest compound yield is obtained if the matching of the minority dies is maximized.

The above clearly shows that an *adaptive* BP-based wafer matching is the best approach to realize the maximal overall compound yield. Table II can be used as a decision rule for the matching criterion selection. Each time a new wafer has to be selected for stacking, the table determines

TABLE II. Yield Based Criterion Selection

$y_t \backslash y_b$	10	20	30	40	50	60	70	80	90
10	MG	MG	MG	UF	UF	UF	MG	MG	MG
20	MG	MG	UF	UF	UF	UF	UF	MF	MF
30	MG	UF	UF	UF	UF	UF	UF	UF	MF
40	UF	UF	UF	UF	UF	UF	UF	UF	UF
50	UF	UF	UF	UF	UF	UF	UF	UF	UF
60	UF	UF	UF	UF	UF	UF	UF	UF	UF
70	MG	UF	UF	UF	UF	UF	UF	UF	MF
80	MG	MG	UF	UF	UF	UF	UF	MF	MF
90	MF	MF	MF	UF	UF	UF	MF	MF	MF

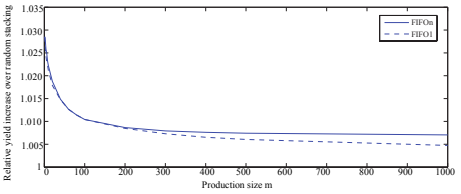


Fig. 11. Yield versus production sizes.

the matching criterion to be used. As an example, consider a three layered stack with equal wafer yield of 80%. According to Table II, the matching of the bottom and middle wafers is performed best using the Max(MF) criterion. If we assume now that the compound yield of this two-stacked IC is 70%, then the matching with the third layer can be best performed based on the min(UF) criterion. This adaptive BP scenario always results in the highest yield for all simulation parameters. From now on, we refer to adaptive BP as the matching scenario that adapts itself with respect to the criterion selection. In the next section, this adaptive scenario is used for comparison with the related work.

C. Repository pollution

In order to estimate the repository pollution, the compound yield for different production sizes is simulated. FIFO1 and FIFOn scenarios are considered for this experiment because: (a) they have the same complexity and (b) FIFOn forces the wafers to leave the repositories while FIFO1 does this only for one repository. Comparing these two scenarios will provide us with an idea about the impact of repository pollution on the overall compound yield.

Figure 11 plots the relative compound yield for the FIFO1- and FIFOn-based matching processes over random stacking for different production wafer sizes m. Here, the reference process is used with $n = 2$, $k = 25$ and the matching criterion max(MG). Three observations can be made from the graph:

- The relative yield for both FIFO1 as well as FIFOn decreases with increasing production size. For low m (typically below 200), the yield for both scenarios are

almost the same.

- As the size of m increases (hence probability of having bad wafers increases), the difference in yield between FIFOn and FIFO1 becomes more visible. The compound yield of FIFO1 decreases faster than that of FIFOn; FIFOn forces wafers to leave the repository at most after $k \cdot n$ cycles and this has a positive affect on the yield.
- The yield degradation due to pollution is stabilizing for larger m.

It can be concluded that in order to minimize the pollution and improve the overall compound yield, it is important to implement a mechanism to force the wafers to leave the repositories after a certain time period.

D. Comparison of Matching Scenarios

In this section, we compare our adaptive BP matching scenario with the scenarios of static repositories published in [11]. We reproduce the same experiments as in [11]; we compare the optimal UB scenario and when this scenario is inapplicable due to memory limitations, the IMH scenario is used [11]. It is worth noting that in case of the optimal UB, different wafer matching criteria will lead to the same compound yield and thus they are not able to enhance the compound yield further.

Table III and IV show these differences for n=3, k=25 with 590 dies per wafer. In each table, the first column provides the varied parameter of the simulation (i.e., stacked number of layers n and the wafer yield y); the second column reports the compound yield of the related work; the third column presents the compound yield of the adaptive BP scenario; the fourth column shows the relative improvement of the BP algorithm versus the obtained yield of the related work; finally, the last column shows the improvement of the BP scenario relative to random stacking. From the tables, we can clearly conclude that running repositories lead to a higher compound yield than static repositories. Although the yield improvement is small, the time complexity difference is huge as summarized in Table V; the table also gives an overview over the memory and runtime complexity cost for each wafer matching scenario. For example, the optimal static algorithm in [11] implemented in C++, requires 0.392 seconds to solve an instance for $n = 3$ and 40.64 seconds for $n = 4$ and runs out of memory for larger number of stacked layers [11]. For the same parameters, our adaptive BP scenario implemented in Matlab required only 0.0028 seconds while using a negligible amount of memory to match a single compound for $n = 7$.

It is important noting that using of wafer matching requires *pre-bond testing*. Hence, it is worth to examine the additional costs required for pre-bond testing. We compare

TABLE III. Yield Comparison with [11] for $n = 3$, $k = 25$, $d = 590$

yield	UB [11] (%)	BP (%)	$\frac{\text{BP}}{\text{UB [11]}}$ (%)	$\frac{\text{BP(k=25)}}{\text{random}}$ (%)
0.3	04.24	04.30	1.42	59.26
0.5	15.08	15.24	1.06	21.92
0.7	37.29	37.46	0.46	9.21
0.9	74.41	74.46	0.07	2.14

TABLE IV. Yield Comparison with [11] for $y = 80\%$, $k = 25$, $d = 590$

n	Alg. [11]	3D Yield [11] (%)	BP (%)	$\frac{\text{BP}}{\text{Alg. [11]}}$ (%)	$\frac{\text{BP(k=25)}}{\text{random}}$ (%)
2	UB	65.25	65.32	0.11	2.06
3	UB	53.56	53.76	0.37	5.00
4	UB	44.58	44.63	0.11	8.96
5	IMH	36.61	37.28	1.83	13.77
6	IMH	30.68	31.29	1.99	19.36
7	IMH	25.76	26.35	2.29	25.65

TABLE V. Wafer Matching Complexity

Ref	Scenario	Memory complexity	Runtime complexity
[11]	Greedy	$O((n+1) \cdot k^n)$	$O(m \cdot k^{n-1} \cdot log(k))$
[11]	IMH	$O(k^2)$	$O(m \cdot n^2 \cdot k^2)$
[11]	ILP/UB	$O((n+1) \cdot k^n)$	$O(\frac{m}{k} \cdot (k!)^{n-1})^*$
[12]	Greedy	$O(k^2)$	$O(m \cdot k^2 \cdot n)$
Ours	Fifo1	$O(n)$	$O(m \cdot k \cdot n)$
Ours	Fifon	$O(n)$	$O(m \cdot k \cdot n)$
Ours	Best Pair	$O(k^2 + n)$	$O(m \cdot k \cdot n)$

* denotes the complexity of the search space

three test flows depicted in Figure 12(a) [12]; they consist of three test moments: Die Test (i.e., pre-bond testing) on the wafers, Stack Tests to verify the stacked wafers before they are packaged and bonded; and Final Tests to ensure overall chip functionality.

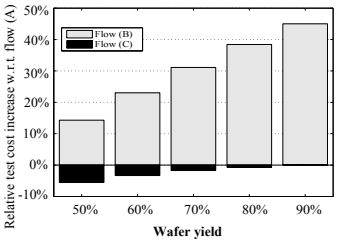
The work in [12] assumes a stack pass yield of 99% and an interconnect yield of 97%. Further, the wafer yield is assumed to be 81.65%, as for our reference process. The three test flows are:

- Flow (A) includes a stacking test and a final test, but has *no pre-bond* die tests. This flow is applicable for random W2W stacking.
- Flow (B) consists of pre-bond die tests (required for wafer matching), a stacking test that tests both dies and interconnects, and a final test.
- Flow (C) consists of pre-bond die tests, a stacking test for the interconnects only and a final test. The idea behind this flow is to optimize the wafer test flow (B) by not replicating the die test in the stacking test. As a consequence of faults introduced into the dies during stacking, a small percentage of faulty dies is still packed.

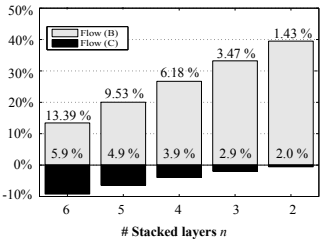
The test cost per functional good stack in terms of test time for the test flows (B) and (C) relative to flow(A) are shown in Figure 12(b) and 12(c); the heights of the bars present this relative cost. The absolute number variation from 2.0-5.9% within the gray bar in Figure 12(c) presents the percentage of faulty packaged 3D-SICs and

Test moment	Flow (A)	(B)	(C)
Die Test	—	✓	✓
Stack Test			
- TSV	✓	✓	✓
- Dies	✓	✓	—
Final Test	✓	✓	✓

(a) Test moments



(b) Test impact on yield



(c) Test impact on # layers

Fig. 12. Normalized cost of test flows (B) and (C) relative to the random W2W stack flow (A)

depends on the stack size n. For Figure 12(b), this package waste is equal to 2.0% for all different yields [12]. The numbers on top of the bars in Figure 12(c) present the yield gain relative to random stacking. Note that these are independent of the test flow.

Relatively to test cost of Test Flow (A), which has no pre-bond die tests, Test Flow (B) negatively effects the test cost, while test flow (C) is able to reduce the test cost per functional good stack. For example, in Figure 12(c), for a two-stacked 3D-SIC, the test time reduction is 0.55 %, the yield is increased with 1.43%, while the packaging cost is increased with 2%. For a six-layered stack, a test cost reduction of 9.23% is expected with a yield increase of 13.39%, but with a package cost increase of only 5.9%. Since the compound yield for running repositories is higher than that of static repositories (while the test costs are the same), we can conclude that the test time per functional working die is lower than in the case of static repositories.

VI. Conclusion

This paper investigates the impact of running repositories on the compound yield for 3D-ICs based on wafer-to-wafer (W2W) stacking. It first introduces a framework for 3D W2W matching, which consists of several wafer matching scenarios. Each scenario is a combination of a matching process, wafer matching criterion, and a repository type (e.g., running or static repositories). The framework shows several scenarios that are not explored yet and a subset of it was selected for further investigation.

Nine wafer matching scenarios have been analyzed based on running repositories. The simulation results showed that the compound yield not only depends on the wafer yield and the number of stacked layers, but also strongly depends on the selected wafer matching scenario. By merging the best performing criteria into the best wafer matching process, an adaptive matching scenario is created that provides the best solution at runtime. By using the adaptive wafer matching scenario, we were able to improve the compound yield up to 13.39% relative to random stacking for realistic wafer yield. Moreover, the adaptive approach outperforms the compound yield of all wafer matching scenarios based on static repositories at a lower cost in terms of the test time, the required memory and time complexity.

References

[1] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Design Test on Computers*, Vol 22, Issue 8, pp. 498-510, Nov 2005.
[2] J. A. Davis et al. "Interconnect Limits on Gigascale Integration (GSI) in the 21st Century", *Proc. IEEE*, Vol 89, Issue 3, pp. 305-224, 2001.
[3] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, Vol 94, Issue 6, June 2006.
[4] G. Loh et al. "Processor Design in 3D Die-Stacking Technologies", *IEEE Micro*, Vol 27, Issue 3, pp. 31-48, Aug. 2007.
[5] K. Puttaswamy et al. "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, Vol 58, Issue 10, pp. 1369-1381, Aug. 2009.
[6] Y-F Tsai et al. "Design Space Exploration for 3-D Cache", *IEEE Transactions on Very Large Scale Integration Systems*, Vol 16, Issue 4, pp. 444-455, 2008.
[7] F. Li et al. "Design and Management of 3D Chip Multiprocessors Using Network-in-Memory", *International Symposium on Computer Architecture*, pp. 130-141, July 2006.
[8] P. Garrou, C. Bower and P. Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.
[9] J. Baliga, "Chips Go Vertical", *IEEE Spectrum*, Vol 41, Issue 3, pp. 43-47, March 2004.
[10] L. Smith, G. Smith, S. Hosali, and S. Arkalgud, "Yield Considerations in the Choice of 3D Technology", *In Proc. IEEE Int. Symp. Semiconductor Manufacturing*, pp. 535-537, 2007.
[11] S. Reda et al. "Maximizing the Functional Yield of Wafer-to-Wafer 3-D Integration", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol 17 , Issue: 9, pp. 1357 - 1362, 2010.
[12] J. Verbree et al. "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-wafer 3D Chip Stacking", Paper accepted in: *IEEE European Test Symposyum*, May 2010.
[13] C. Ferri et al. "Parametric Yield Management for 3D ICs: Models and Strategies for Improvement", *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, Vol 4, Issue 4, Oct. 2008.
[14] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference, 2009*, Nov. 2009.
[15] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas", *IEEE Transactions on Semiconductor Manufacturing*, Vol 18, Issue 1, pp. 136-139, Feb. 2005.
[16] M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Wiley-VCH, Weinheim, Germany, Aug. 2000.

A

Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching

MOTTAQIALLAH TAOUIL, Delft University of Technology, the Netherlands
SAID HAMDIOUI, Delft University of Technology, the Netherlands
ERIK JAN MARINISSEN, IMEC vzw, Belgium

Three-Dimensional Stacked IC (3D-SIC) using Trough-Silicion Vias (TSV) is an emerging technology that provides heterogeneous integration, higher performance and lower power consumption compared to traditional ICs. Stacking 3D-SICs using Wafer-to-Wafer (W2W) has several advantages such as high stacking throughput, high TSV density and the ability to handle thin wafers and small dies. However, it suffers from low compound yield as the stacking of good dies on bad dies and vice versa cannot be prevented. This paper investigates wafer matching as a mean for yield improvement. It first defines a complete wafer matching framework consisting of different scenarios; each scenario is a combination of a *matching process* (defines the order of wafer selection), a *matching criterion* (defines whether good or bad dies are matched), *wafer rotation* (defines either wafers are rotated or not) and a *repository type*. The *repository type* specifies whether either the repository is filled immediately after each wafer selection (i.e., running repository) or after *all wafers* are matched (i.e., static repository). A mapping of prior work on the framework shows that existing work has mainly explored scenarios based on static repositories. Therefore, the paper analyzes scenarios based on running repositories. Simulation results show that scenarios based on running repositories improve the compound yield with up to 13.4% relative to random W2W stacking; the improvement strongly depends on the number of stacked dies, die yield, repository size, as well as on the used matching process. Moreover, the results reveal that scenarios based on running repositories outperform those of static repositories in terms of yield improvement at significant run-time reduction (three orders of magnitude) and lower memory complexity (from exponential to linear in terms of stack size).

Additional Key Words and Phrases: 3D integration, wafer matching, matching criterion, compound yield, wafer-to-wafer stacking

ACM Reference Format:
Mottaqiallah Taouil, Said Hamdioui and Erik Jan Marinissen, 2014. Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching. *ACM Trans. Embedd. Comput. Syst.* V, N, Article A (January YYYY), 24 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Three-Dimensional Stacked Integrated Circuits (3D-SICs) consist of multiple stacked planar dies fabricated in a conventional process augmented by new Through Silicon Via (TSV) process steps, which electrically connect the dies in the vertical direction. The ability to create these vertical stacked ICs alleviates or even eliminates various existing problems in planar ICs. An efficient partitioning of IP cores among the stacked dies reduces the need for long wires and is thus able to reduce the wire delay, as well as the power dissipation [Davis et al. 2005; Davis et al. 2001]. Heterogeneous integra-

Author's addresses: M. Taouil and S. Hamdioui, Software and Computer Technology, Delft University of Technology; E.J. Marinissen, Imec vzw.

DOI: http://dx.doi.org/10.1145/0000000.0000000

tion is a promising concept for 3D-SICs, since each layer can be manufactured with dedicated technology and optimized for speed or area. This affects the yield, performance, and lithography cost positively. Furthermore, miniaturization of the physical sizes of stacked dies reduces the footprint and volume area, which in turn increases the package density. Examples of 3D-SICs include 3D CMOS sensors [Patti 2006], 3D FPGAs [Patti 2006], 3D processors [Loh et al. 2007], 3D cache and memory [Puttaswamy and Loh 2009; Tsai et al. 2008] and memories on top of processors [Patti 2006; Li et al. 2006].

Tiers are stacked at the die or wafer level and can be stacked based on Wafer-to-Wafer (W2W), Die-to-Wafer (D2W), or Die-to-Die (D2D) bonding [Garrou et al. 2008]. High alignment accuracy is feasible in D2D and D2W bonding, but it negatively impacts the throughput. In D2D and D2W bonding, Known Good Die (KGD) stacking can be applied to prevent faulty dies from being stacked [Garrou et al. 2008]. W2W stacking allows for (a) high manufacturing throughput due to single wafer alignment, (b) thinned wafers and small die handling, and (c) the ability to create a higher TSV density, but requires dies to be of equal size and therefore probably fitting more for FPGA and memory application. In general, the compound yield is one of its major bottlenecks that must be overcome to make W2W stacking a lucrative business [Baliga 2004].

The compound yield can be improved by *wafer matching*, initially introduced by Smith et al. [Smith et al. 2007]. In wafer matching, a software algorithm keeps track of the test result of each die of the wafer stored in a fault map. The algorithm matches wafer pairs that contain the same or similar fault maps. This increases the compound yield over randomly stacked wafers. More elaborate studies of wafer matching are presented in [Reda et al. 2009; Verbree et al. 2010], e.g., by considering different die yields, stack and repository sizes etc. However, they only considered wafer matching with *static* repositories, i.e., the repositories are not replenished unless they are empty. In [Singh 2011], the authors present wafer rotation; wafer rotation gives more freedom during stacking and therefore it increases the matching combination yielding in higher compound yield. All these papers have focused *only* on matching of the *good dies* from the bottom layer with the *good dies* from the top layer as a matching criterion. However, this could also be the matching of the *faulty* dies instead of the good dies.

In this paper we examine the impact of running repositories (i.e., replenished repositories) on the compound yield by considering different wafer matching criteria [Taouil et al. 2010] and compare the obtained results with those of static repositories. For running repositories, each time a wafer is selected from the repository its empty spot is directly replenished with a new wafer. This keeps the size of the *running* repository constant over time.

The main contributions of this paper are the following.

— A framework that covers all matching scenarios.
— A mapping of prior work on the framework.
— Impact and analysis of the most important uncovered scenarios and their comparison with prior work.

The remainder of the paper is organized as follows. Section 2 introduces the framework for wafer matching and defines the focus of this paper. Section 3 provides an overview of related prior work. Section 4 describes two matching scenarios used in this work. Section 5 describes the experimental setup. Section 6 presents the simulation results. Section 7 compares the obtained results with related prior work. Finally, Section 8 concludes the paper.

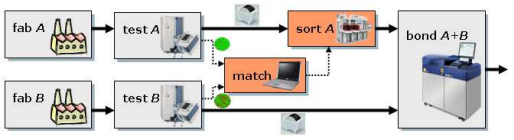


Fig. 1. W2W stacking flow [Verbree et al. 2010].

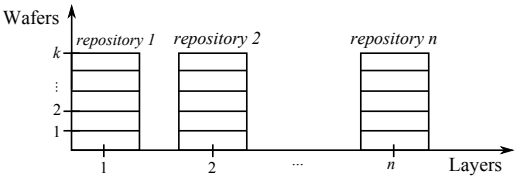


Fig. 2. Wafer vs layers in 3D stacking.

2. WAFER MATCHING FRAMEWORK

This section defines a framework for all possible *wafer matching scenarios* for 3D W2W stacked ICs; a wafer matching scenario combines four parameters:

(1) *matching process* [Taouil et al. 2010]: it defines how the repositories are traversed and and how many wafers are selected from each repository visit at a time.
(2) *matching criterion* [Taouil et al. 2010]: it defines the criterion based on which a pair of wafers are matched; for example good dies or faulty dies.
(3) *wafer rotation* [Singh 2011]: it defines the angle at which one of the two wafers can be rotated before stacking in order to realize a better compound yield. For example, the top wafer can be rotated with 180^0 before it is stacked on the bottom wafer.
(4) *repository type* [Taouil et al. 2010]: it defines the nature of the replenishing of wafers. For running repositories, wafers are immediately replenished after each wafer selection. For static repositories, replenishing takes place only when the repositories are empty.

In the rest of the section, first the problem of W2W 3D-SICs is defined. Thereafter, the four parameters of the wafer matching scenarios are addressed.

2.1. W2W Matching Problem

Figure 1 [Verbree et al. 2010] shows the W2W flow using wafer matching. Instead of stacking wafers randomly, wafer matching can be applied. This requires pre-bond tests on wafers to obtain wafer maps with per-die pass/fail results. Using a software tool to analyze the pass/fail results allows to match wafers with similar or same fault distributions. Based on the outcome of the matching, wafers need to be sorted before stacking. This wafer sorting can be performed with wafer sorter machines.

To manufacture n-wafer stacks using W2W stacking, n different wafers have to be stacked. These steps are repeated until the production size m (the total of produced wafer stacks) is reached. Therefore, for each layer in the stack m wafers are manufactured. To obtain the maximum overall yield, all wafers have to be matched appropriately. It is therefore a challenging task, both from computational complexity and a logistics point of view [Reda et al. 2009]. It is from a manufacturing point of view more practical to match only a small subset at a time; this subset is grouped in repositories each of size k. In total there are n of these repositories as depicted in Figure 2.

The problem of W2W 3D-SICs can now be defined as follows: Given, (a) n repositories each with k wafers, (b) fault maps for all the wafers (based on pre-bond testing), and (c)

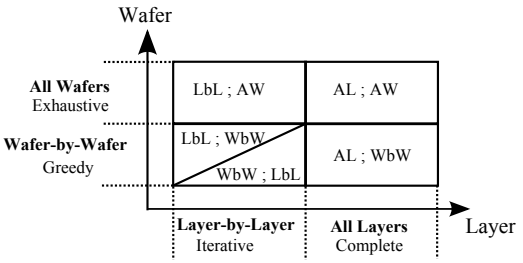


Fig. 3. Framework of matching processes.

a production size of m n-wafer stacks, the purpose is to maximize the overall compound yield for all 3D-SICs, by selecting appropriate wafers for the n-layer 3D-SICs from the repositories. In [Reda et al. 2009], the W2W matching problem is proved to be $\mathcal{NP}$-hard.

2.2. Matching Process

The matching process defines the step-by-step process to be followed in order to realize wafer matching. The problem can be formulated as a 2-dimensional problem where the dimensions are k (vertical dimension in Figure 2) and n (horizontal direction in the figure). The dimensions are independent from each other. The horizontal direction determines which layers are active in each matching step, while the vertical dimension consists of the selection of wafers from the repositories.

Therefore, the matching process determines the *number of repositories* for the horizontal dimension and the *number of wafers* for the vertical dimension that are considered at a time. Depending on the number of involved *repositories* (see also Figure 2), two cases are distinguished:

— *Layer-by-Layer (LbL)*: Initially, the first two repositories are selected for wafer matching. In each successive step, only one additional repository is considered during matching. Hence, this is an *iterative* process in terms of the number of involved layers.
— *All-Layers (AL)*: In each step of the wafer matching process, all repositories are considered at once. As every wafer in every repository is taken into account, this process is labeled *complete*.

In a similar way, depending on the number of *wafers* involved in each step of the matching process, two cases can be distinguished for the vertical dimension:

— *Wafer-by-Wafer (WbW)*: In each step of the wafer matching process, the best wafer combination is selected (e.g., based on highest yield). Only one wafer from each repository is selected after matching, with no regard to the remaining wafers in those repositories. Thus, this process is regarded as *greedy*.
— *All-Wafers (AW)*: In each step of the wafer matching process, all wafers from all involved repositories are simultaneously matched. As the process considers all possible outcomes for all k wafers to be matched, this process is considered to be *exhaustive*, and providing a global solution for this dimension.

A matching process is formed when both dimensions are combined, e.g., when AL is used in combination with WbW. The order of these loops affects the matching process. We use the notation A;B to denote that subprocess A is part of the outer loop and subprocess B of the inner loop. The case with LbL and WbW is special, as in both di-

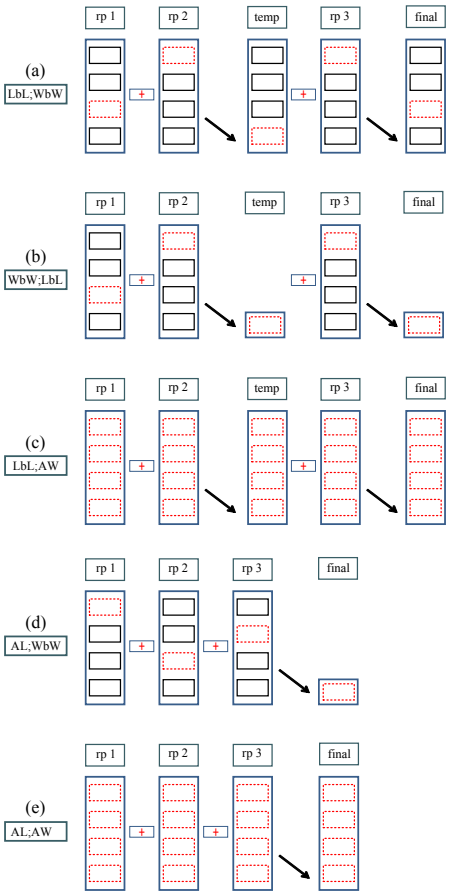


Fig. 4. Wafer matching processes.

mensions a loop is used to traverse over the repositories and to select wafers from each repositories. Therefore, the above combinations result into five possible wafer matching processes, as shown in Figure 3 and illustrated in Figure 4; they are explained next.

— LbL;WbW: This process consist of two loops in which the outer loop traverses the repositories and the inner loop selects wafers from the traversed repositories. Figure 4(a) shows this for n=3 and k=4. In the first step of the outer loop, all wafers of repositories *rp1* and *rp2* are considered for matching; the inner loop selects the best wafer-pair at a a time for further processing. The inner loop is repeated untill the repositories are empty; the resulted wafer pairs are stored in a temporary repository *temp*. The outer loop continues now by considering repositories *temp* and *rp3*. The boxes with dashed lines denote which wafers are selected for stacking and the arrow shows the result of the stacking operation.
— WbW;LbL: In this process (Figure 4(b)), the outer loop (WbW) considers all wafers of repositories *rp1* and *rp2* to select the best wafer-pair. Because the inner loop is

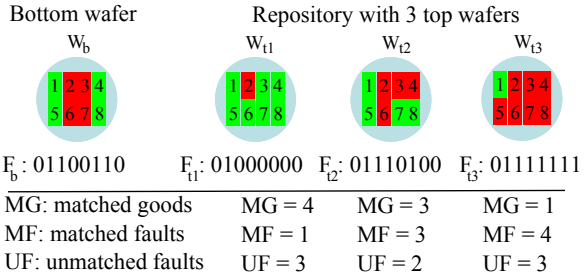


Fig. 5. Wafer matching criteria.

LbL, only this pair (stored in *temp*) is used for further processing. Once the complete stack is created, the entire process repeats again.

— LbL;AW: Similar to LbL;WbW, the matching process iteratively considers two repositories at a time, but in this case, all wafers from the two repositories under consideration are being matched simultaneously (see Figure 4(c)). The difference between LbL;WbW and LbL;AW is that LbL;AW provides an *exhaustive* solution within the LbL process, while LbL;WbW selects the wafers one by one in a *greedy* way.

— AL;WbW: In this process, n wafers are simultaneously selected from n repositories (one wafer from each repository) in order to realize the best matching combination as depicted in Figure 4(d). Note that in each step of this process, only the best combination of n wafers is considered.

— AL;AW: This process is similar to AL;WbW, but here, all k wafers from each repository are matched simultaneously (see Figure 4(e)). It does not only consider the best match of n wafers at each step, but in fact it considers the overall compound yield by finding the best k n-wafer combinations.

It is worth noting that for the LbL processes, the traversal order in which the layers are selected can be freely chosen; e.g., in the order 1, 2, ... n, or n, ..., 2,1. In total the are $n!$ possible sequences. However, only $\binom{n}{2} \cdot (n-2)! = \frac{n!}{2}$ are relevant; the first term of the equation represents the number of possibilities to select the first two repositories out of n without considering the order as it will have no impact, while the second term $(n-2)!$ represents the number of combinations of the remaining repositories.

2.3. Matching Criteria

A matching process select wafers based on a certain criterion; e.g., best good dies. Each criterion is independent from the process. Based on the fact that each wafer consists of both good and bad dies and that the purpose of the wafer matching is to maximize the compound yield, one can define three possible criteria:

(1) *Max(MG)*. This criterion selects the best wafer pair combinations based on the maximum number of *Matched Good (MG)* dies. All the published work so far regarding wafer matching considers only this criterion.
(2) *Max(MF)*. The best wafer pair combinations is selected based on maximum number of *Matched Faulty (MF)* dies.
(3) *Min(UF)*. This criterion selects the best wafer pair combinations based on minimum number of *Unmatched Faulty (UF)* dies. The objective is to increase the compound yield by minimizing faulty dies that land on good dies and vice versa.

All the above criteria produce the same results in terms of compound yield in case the wafer matching process is exhaustive (AW process) for static repositories. For

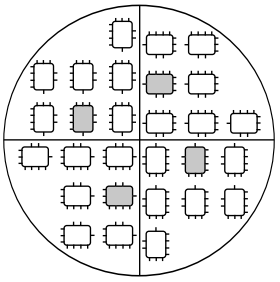


Fig. 6. Wafer rotation: A wafer with four quarters

the greedy wafer matching processes (WbW), it is evident that different criteria lead to different results. For running repositories, the criteria lead to different compound yields, as will be shown later in this paper.

In order to provide more insight into the impact of the above criteria on wafer selection, refer to the example shown in Figure 5, which considers a bottom wafer W_b and three potential top wafers (W_{t1}, W_{t2}, W_{t3}), each with its own fault map. The fault map of each wafer is denoted by F and contains a sequence of 0s (good dies) and 1s (bad dies) ordered according to the indexes of the dies on the wafer; e.g., the bottom wafer has $F_b = 01100110$, since the dies 2, 3, 6, and 7 are faulty. The bottom table in the figure lists the value of the different criteria for the three matching possibilities; e.g., for matching W_b-W_{t1}, the number of matched good dies is MG = 4 (which are dies 1, 4, 5, 8). The figure clearly shows that depending on the criterion, different top wafers have to be selected; e.g., if max(MG) is considered, then W_{t1} will be selected. However, if the max(MF) is the criterion, then W_{t3} is the best match. The criteria can be mathematically formulated. Let the function $G(F_i)$ be the number of 1's that the fault map F_i contains. Then,

$$Max(MG) = max(\forall_{i,j}, G(\bar{F}_i \& \bar{F}_j)) \quad (1)$$

$$Max(MF) = max(\forall_{i,j}, G(F_i \& F_j)) \quad (2)$$

$$Min(UF) = min(\forall_{i,j}, G(F_i \oplus F_j)) \quad (3)$$

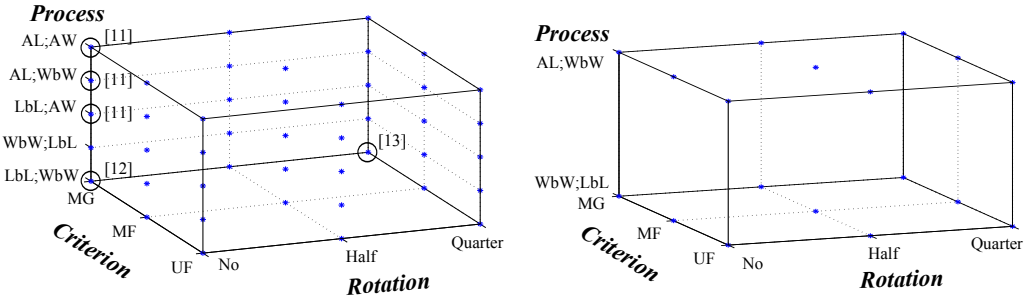
Here, $0 \leq i, j \leq k$, where k the repository size.

2.4. Wafer rotation

By providing the wafer rotation freedom before stacking, compound yield can be improved [Singh 2011]. Obviously, designers of wafer masks have to take this into consideration in order to make it feasible. This might results in lower die yield. Figure 6 shows a general example of a wafer which contains 28 dies (each quadrant contains 7 dies) that can be used for rotation. The wafer can be rotated 90^0, 180^0, and 270^0 before stacking. Note that after each rotation, the position and orientation of the dies remain exactly the same.

In general, the following wafer rotation schemes are applicable:

— no-rotation: it allows only one possible wafer orientation (the default one).
— half-rotation: it allows a wafer rotation of 180^0, resulting into two possible wafer orientations for two wafers to be stacked.
— quarter-rotation: it allows a wafer rotation of 90^0, 180^0 or 270^0 as Figure 6 shows. This results into four possible wafer orientations for two wafers to be stacked.



(a) Static repository scenarios (b) Running repository scnearios

Fig. 7. Wafer Matching Scenarios.

It is obvious that the higher the rotation freedom, the higher the number of possible wafer orientations; hence, the higher the impact on the compound yield.

2.5. Repository Type

Wafer matching can be considered as a time consuming process when the objective is to maximize the compound yield for a production size m; m can be in the order of thousands or millions. To split up and divide the problem, a fixed number of k (usually $k \ll m$) wafers per repository can be considered and matched at a time. Depending on whether a repository is replenished immediately (after a wafer is removed from it for matching and stacking) or not, two classes can be defined:

— *Static repositories*: From each repository, k wafers are selected and processed before considering the next group of k wafers. The procedure stops after m/k steps.
— *Running repositories*. Each repository is immediately replenished with a new wafer each time a wafer is selected. The procedure stops after m wafers are processed.

The freedom to select wafers from static repositories reduces over time, since the repositories become more and more empty. Running repositories, however, provide always the full repository (of size k) to select from; this improves the effectiveness of wafer matching as compared with static repositories. The downside of running repositories is that unattractive wafers may remain in the repository for many iterations, occupying space, and in effect reducing the size of the repository in the long run. We call this effect, the repository *pollution*.

Another difference between static and running repositories is the actual implementation. Static repositories map fairly well onto a production line, where basically the repositories are the wafer containers that move from one machine in the production line to the next. With running repositories, a container would need to go back and forth between the bonding machine and the wafer production line to be replenished, before a new selection is made. Clearly, this is impractical, and therefore we suggest using two containers. One to select from, and one acts as a wafer source to replenish the first one at the bonding machine. This, however, reduces the effective capacity of the bonding machine as both containers are in the machine, yet only one is used to select a wafer from.

Not all matching processes (see Section 2.2) are applicable for running repositories. For example, the processes LbL;WbW and LbL;AW depicted in Figure 4(a) cannot be performed for running repositories, as the matching between the first two repositories will stay in an infinite loop. Further, note that the matching process AL;AW is equal

for both the static and running repositories, as all wafers are selected at the same time from all repositories.

2.6. Wafer Matching Scenarios

Figure 7(a) and 7(b) depict possible matching scenarios for static and running repositories respectively; each * in the figure presents a scenario. A wafer matching scenario is formed by combining (a) a matching process (z-axis), (b) a matching criterion (y-axis), (c) support for wafer rotation (x-axis), and (d) a repository type (Figure 7(a) vs Figure 7(b)). Going vertically up in the figure (the z-axis) leads in general to a higher compound yield due to the use of more advanced matching processes, but at the cost of increased computational effort. The circled items contain previous work in the area of wafer matching and are discussed in the next section.

3. RELATED PRIOR WORK CLASSIFICATION

In this section, the related prior work is considered. Section 3.1 discusses the contributions of previously published wafer matching scenarios. Section 3.2 maps these scenarios on the framework.

3.1. Contributions of Prior Work

Improving the yield for 3D circuits based on wafer matching was initially introduced by Smith et al. [Smith et al. 2007], where the authors compared the yield improvement of a single-die SoC, by mapping it onto a 3D-SIC with two equally sized layers. The yield improvement is simulated both for D2W and W2W stacking. In the W2W stacking case, a software matching algorithm is used to select pair-wise the best wafers from two repositories with a size of 25 each.

The concept of W2W matching introduced by Smith [Smith et al. 2007] is further generalized by Reda et al. [Reda et al. 2009]; the paper formulates the W2W matching problem and proves it to be $\mathcal{NP}$-hard. Several wafer matching algorithms are investigated, including the optimal solution. In [Verbree et al. 2010], Verbree et al. define a simplified mathematical model for wafer matching and presented its simulation results. The model has some practical limitations such as a fixed number of faulty dies per layer, but nevertheless it gives a good indication of the yield improvements. The simulation results were based on 10000 experiments. In [Singh 2011], the author extended the work of [Verbree et al. 2010] and added wafer rotation to investigate further yield improvements. The same author further improves the yield numbers by considering a radial defect model in [Singh 2012].

In [Ferri et al. 2008], Ferri et al. used wafer matching to increase the *parametric* yield of a two layered D2W stacked 3D-SIC. Only functional dies are considered in this case to produce an optimal binning; i.e., maximize the fastest speed bins and minimize the slowest ones. Wafer matching is then used to combine and improve the 3D parametric yield by including the process variation of both layers in a D2W stacking approach.

All related prior work considered *static* repositories and used a *single* wafer matching criterion, i.e., the matching of *good dies* from the bottom layer with *good dies* from the top layer. However, running repositories in combination with different matching criteria can be used instead. In [Taouil et al. 2010], we have introduced the concept of running repositories.

3.2. Mapping of Prior Work

The wafer matching scenarios of prior work can be mapped on the solution space of wafer matching scenarios depicted in Figure 7; in the figure, the scenarios covered by prior work are marked with circles and references. They are explained next. The

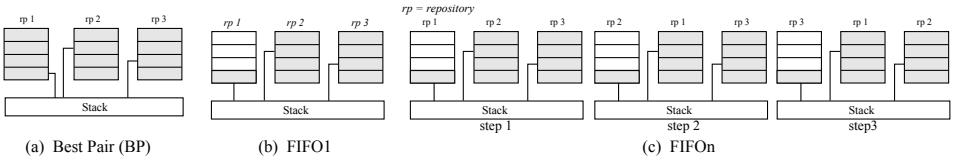


Fig. 8. Matching scenarios for LbL;WbW.

greedy algorithm in [Verbree et al. 2010] is a LbL;WbW process. It creates a sorted list based on the compound yield of all wafer combinations between two repositories. From this list, valid pairs are selected starting from the highest yield. A combination is considered invalid when at least one of the wafers of the current compound has already been taken in a previous selection. After the repositories are empty, the repository of the next layer is matched with the current temporary stacks. In [Reda et al. 2009], three matching scenarios are described. In the first scenario, a greedy algorithm is used to create a sorted list of all k^n wafer combinations; this is in fact AL;WbW process. The difference with the greedy algorithm in [Verbree et al. 2010] is that in this scenario all layers are considered at the same time. The second scenario, referred to as the Iterative Heuristic Matching (IHM) algorithm, considers two repositories at a time and optimally matches them by the Hungarian algorithm. These steps are iteratively repeated by including one additional repository in each iteration. The IHM algorithm is an LbL;AW process. In the third scenario, a global optimal algorithm based on Inter Linear Programming (ILP) is used to explore the exhaustive search space and obtain the global maximum yield. The execution time reduction of ILP scenario is realized by relaxing the ILP and allowing the program variables to take fractional values; this resulted into Upper Bound (UB) scenario. The ILP and UB scenarios are both AL;AW processes. In [Singh 2011], the authors use the same greedy algorithm as in [Verbree et al. 2010] but in addition use wafer rotation. All these previous work focused on static repositories using the Max(MG) criteria only. From Figure 7 we conclude that several scenarios are not explored yet, mainly the ones for running repositories (i.e., WbW;LbL using rotation and AL;WbW); these are discussed in the next section.

4. SCENARIOS FOR RUNNING REPOSITORIES

Recently, in our work [Taouil et al. 2010], we have used the WbW;LbL matching process while considering all matching criteria for running repositories, but without rotation. In the rest of this section we describe the two processes WbW;LbL and Al;WbW which are further investigated in this paper. Section 4.1 considers implementation schemes for the WbW;LbL matching process, while Section 4.2 focuses on the AL;WbW matching process.

4.1. Scenarios based on the WbW;LbL process

As already explained, WbW;LbL process considers only two repositories at a time; in addition, only a single wafer pair selection is performed. Based on the wafer pairs selection order, three WbW;LbL matching processes can be defined:

— Best Pair-based matching process.
— FIFO1-based matching process.
— FIFOn-based matching process.

These are explained in the next subsections.

4.1.1. Best Pair (BP). In this matching process, wafers from the first two repositories are matched in pairs without any selection restrictions; see Figure 8(a) for $n = 3$.

The process iteratively proceeds along the repositories until a single n-wafer match is determined. Then, this process is repeated until the production size m is met. The run-time complexity equals to $O(m \cdot k \cdot n + k^2) = O(mkn)$. Initially, k^2 comparisons are performed on the initial set of the first two repositories. The best pair is selected and used to search for the best matching with the rest of the $n-2$ repositories (one by one); this requires $(n-2)*k$ comparisons. Note that after replenishing, the process will be repeated; however, now the first two repositories require only $2*k-1$ comparisons rather than k^2 since the results of the previous comparison can be reused. The memory complexity is $O(k^2+n)$, required to store all k^2 compound yield combinations between the first repositories, and to hold a list of n numbers identifying the indices of the selected wafers of each repository. Both the run-time and memory complexity are the same for all rotation schemes, i.e., for no-rotation, half-rotation and quarter-rotation.

4.1.2. FIFO1. In the FIFO1-based matching process the wafers from the first repository are selected based on a FIFO approach, as depicted in Figure 8(b) for n=3. The wafers from repository rp1 are selected in pre-defined order and without any freedom to match them with the best wafers from the second repository. The process iterates over all the repositories. The size of the first repository is actually irrelevant, and can be changed to one. The order in which the repositories are traversed is linear starting at repository 1 and ending at repository n. Note that for FIFO1, the pollution is not an issue for the first repository, since wafers are forced to get out. The run-time complexity of FIFO1 is $O(m \cdot k \cdot (n-1)) = O(m \cdot k \cdot n)$, where m the production size, k the repository size and n the stack size. The worst case memory complexity is $O(n)$; this is the memory required to store the list of indices holding the positions of the selected wafers from each repository.

4.1.3. FIFOn. In the FIFOn-based matching process, we generalize the concept of FIFO1. This is done by moving the FIFO-repository in a round robin fashion among all repositories as shown in Figure 8(c) for n=3. At the left side of the figure, repository rp1 is used as FIFO. After the first n-wafer stack is created, the repository belonging to the next layer is considered to be the FIFO as shown in the middle of Figure 8(b). Here, the process starts from rp2 and proceeds next with rp1 and rp3; the traversal order is written in the top part of Figure 8(c). For the next compound, rp3 is used as FIFO. These steps are repeated until the production size is reached. The first traversed repository is the repository that is considered as FIFO, the remaining repositories are traversed in monotonic increasing order starting at repository 1 and ending at repository n. FIFOn is able to control the pollution since it forces wafers to stay maximally $n \cdot k$ cycles in a repository. In this way, the repositories are not contaminated with bad wafers that stay for a long time in the repositories without being selected. The memory and run-time complexity for this scenario are the same as in the case for FIFO1, since it only changes the position of the FIFO-repository.

Note that the given three WbW;LbL processes, in total 27 scenarios can be derived: 3 (matching processes) · 3 (matching criteria) · 3 (wafer rotation schemes) · 1 (running repository).

4.2. Scenarios based on the AL;WbW process

In AL;WbW process, all repositories are simultaneously considered in the matching process. From each repository a single wafer is selected; only those wafers are selected that combined result in highest compound yield.
In order to determine this best match, we have to create a list of k^n of n-wafer combinations. From this list the best combination is selected. The complexity of this task is $O(k^n)$. After the best match is determined, the selected wafers must be replaced by new

wafers; i.e., repositories are replenished. Next, the new possible k^n n-wafer combinations have to be recalculated. However, the n-wafer combinations for the replenished wafers should be determined; the other combinations were already computed in the first step. Hence, the total number of computations in the second step is $k^n-(k-1)^n$; k^n is the total number of n-wafer combinations, while $(k-1)^n$ is the number of n-wafer combinations based on repositories with $k-1$ wafers. Hence, the complexity of the second step is $O(k^n-(k-1)^n) = O(k^{n-1})$. This second step is repeated until m stacks are created. Therefore, the total complexity is $O(k^n + m \cdot k^{n-1})$ = $O(m \cdot k^{n-1})$. The memory complexity of the scheme is $O(k^n)$, which indicates the amount of memory required to store the list of all possible wafer combinations. As in this scenario all n-wafers are considered simultaneously in each step, only max(MG) criterion is practical; we are interested in maximizing the compound yield, which is exactly given by max(MG). The other two criteria do no make sense for the AL;WbW matching process.

5. EXPERIMENTAL SETUP

This section presents the experimental setup of this paper. First, the reference process is described in Section 5.1; it describes the default parameters for each experiment. Thereafter, the performed experiments are described in Section 5.2.

5.1. Reference Process

We categorize the default parameters of the reference process in two types of classes:

— design/manufacturing parameters, such as stack size and die yield, etc.
— scenario parameters, such as matching process, criteria etc.

Each class is described next.

Design/manufacturing parameters
The performance of the matching scenario is heavily influenced by several design and manufacturing parameters. Their values are based on the reference process in [Verbree et al. 2010]. A standard $300\ mm$ diameter wafer is selected with an edge clearance of 3 mm. The defect density is considered to be d_0 = 0.5 defects/cm^2 and the defect clustering parameter $\alpha = 0.5$. For the reference design, the die area is assumed to be A=$50mm^2$. For this die area and wafer size, the number of Gross Dies per Wafer (GDW) approximately equals to 1278 [De Vries 2005]. However, since we apply also quarter-rotation we always select the number of dies to be a multitude of 4, therefore we use 1276 dies per wafer. The expected die yield can be estimated by the negative binomial formula as: $y = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$= 81.65% [Agrawal 2000]. The compound yield improvements are heavily affected by the stack size and the number of wafers per repository. We select a default stack size of n=2 and a repository size of k=50.

In our experiments, we simulate a default production size of $m = 25000$. Here, m is the number of stacked wafer sets. Initially, each repository is filled up with k wafers and after selecting and stacking m-compound wafers, the wafers that are left in the repository are discarded and not included in the simulation results for two reasons.

(1) We want to observe the impact of the running repository only.
(2) Even if the wafers would be thrown away, their impact on the compound yield is minimal, due to a high production volume m. Actually, the matching scenarios for static repositories presented in [Reda et al. 2009; Verbree et al. 2010; Singh 2011] could be used to match these last k unconsidered wafers.

Each simulated wafer is generated by the default Matlab pseudo-random function and is represented by a sequence of 1's and 0's for good and bad dies respectively.

Scenario parameters
Each scenario consists of 4 parameters as described in Section 2. The default values for the reference process consist of the following: (a) the LbL;WbW matching process implemented using BP, (b) the max(MG) matching criterion (c) no wafer rotation, and (d) running repositories.

5.2. Performed Experiments

Two types of experiments are performed. The first type investigates the impact of design/manufacturing parameters, while the second type investigates that of scenario parameters.

Design/manufacturing parameters
For the experiments we explore the impact of three design/manufacturing parameters, while considering the repository size $1 \leq k \leq 50$; they are:

— stack size n: where $2 \leq n \leq 6$.
— die yield Y_D: where $50\% \leq Y_D \leq 90\%$.
— die area A: where $25mm^2 \leq A \leq 125mm^2$.
— production size m: where $1 \leq m \leq 25000$.

Scenario parameters For the experiments, we explore the impact of three parameters for running repositories; they are:

— matching process: we compare the performance of both the WbW;LbL and AW;LbL process.
— matching criterion: we investigate the impact of all the criteria max(MG), min(UF), max(MF) on the compound yield for the WbW;LbL processes.
— wafer rotation: we investigate no, half- and quarter-rotation on the compound yield for the best matching process.

We compare the compound yield using the different wafer matching scenarios with respect to random stacking, unless otherwise stated. The compound yield Y_{rand} for random W2W stacking can be calculated by:

$$Y_{rand}(Y_D, n) = Y_D^n \tag{4}$$

6. SIMULATION RESULTS

In this section, we present the impact of the parameters due to design/manufacturing considerations and the impact of scenario parameters in Sections 6.1 and 6.2 respectively.

6.1. Impact of design and manufacturing parameters

In this subsection, we investigate the impact of the stack size, die yield and die area on the compound yield of the 3D-SICs for different repository sizes.

Impact of stack size

Figure 9 plots for the reference process the relative compound yield with respect to random stacking (i.e., $k = 1$, Y_{rand}=66.67%) for different stack sizes n and repository sizes k. The figure clearly shows two trends. The first trend is the higher improvement of the compound yield for larger repository sizes; however, the obtained gain stabilizes with larger k. The second trend is the increased improvement for a larger stack size.

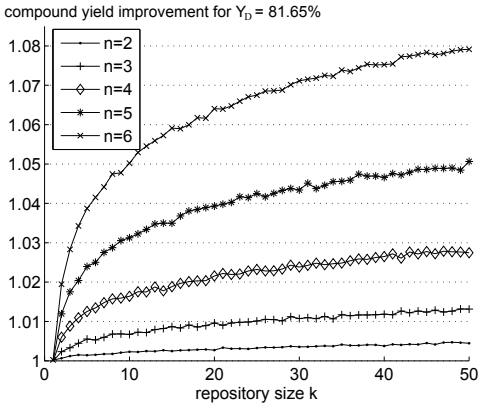


Fig. 9. Compound yield for different stack size n.

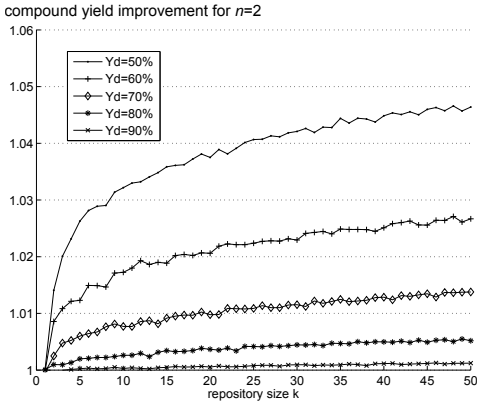


Fig. 10. Compound yield for different die yield Y_D.

For example, the compound yield improvement for n=6 reaches up 8% for k=50, while this improvement is only 0.5% for n=2.

Impact of die yield

Figure 10 shows the impact of the die yield on the compound yield using the reference process for different repository sizes. The figure shows also two trends. First, the larger the repository size k, the higher the compound yield improvement similarly as the trend observed in Figure 9. Second, the lower the die yield Y_D, the higher the compound yield improvement, for example, for Y_D=50% this improvement reaches up to 4.6% while for Y_D=90% a negligible improvement is realized.

Impact of die area

Figure 11 shows the impact of variable die areas (between $25mm^2$ and $125mm^2$) on the compound yield. For these areas, we used the same yield values and number of dies per wafer as in [Verbree et al. 2010], i.e., 89.44% yield and 2596 dies per wafer for the area of $25mm^2$ (Y_{rand}=80.00%), 81.65% yield and 1276 dies per wafer for the die

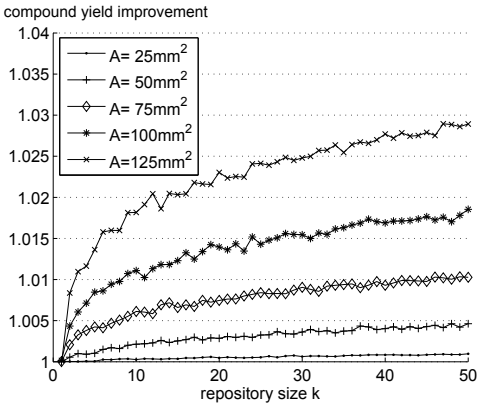


Fig. 11. Compound yield for different die area A.

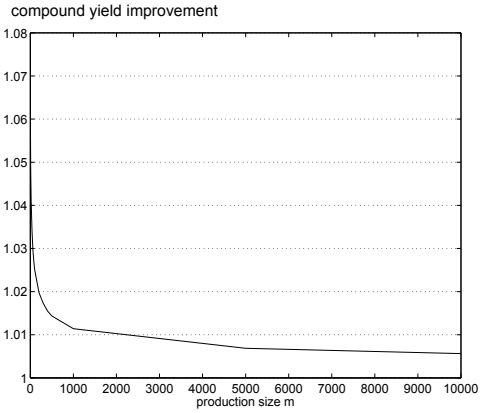


Fig. 12. Compound yield for different production size m.

area of $50mm^2$ (Y_{rand}=66.67%), 75.59% yield and 836 dies per wafer for the die area of $75mm^2$ (Y_{rand}=57.14%), 70.71% yield and 620 dies per wafer for the die area of $100mm^2$ (Y_{rand}=50.00%), and 66.60% yield and 492 dies per wafer for the die area of $125mm^2$ (Y_{rand}=44.35%). The figure shows a similar trend with respect to the repository size as in the previous cases; i.e., the compound yield improvement increases with the repository size and stabilizes for larger repository sizes. With respect to the die area, we see that the compound yield can be improved much better for larger die areas; this is due to fewer dies per wafer and lower die yield.

Impact of production size

Figure 12 plots the compound yield improvement for a variable production size for the reference process (Y_{rand}=66.67%). The figure reveals that the improvement decreases as a function of the production size. In the beginning of the matching process, fresh wafers reside in the repository and therefore the gained yield improvement is relative higher. As the good wafers are selected first, and more bad wafers reside in the repository, the compound yield slowly reduces and reaches a stable value. In our ex-

Table I. Compound Yield Improvement for AL;WbW Scenario.

Relative yield improvement		Comparison	
die yield (%)	$\frac{\text{AL;WbW}}{\text{random}}$ (%)	$\frac{\text{AL;WbW}}{\text{Deafult BP}}$ (%)	$\frac{\text{AL;WbW}}{\text{Adaptive BP}}$ (%)
30	49.36	7.60	06.03
50	15.38	3.03	00.19
70	04.52	1.24	-1.64
90	00.40	0.18	-1.04

periments, we consider a production size of 25000 stacked wafers. For this size, the compound yield reaches stable values. Note that if the production size is very low a higher compound yield can be realized.

6.2. Impact of scenario parameters

In this section, we investigate the impact of the matching process, matching criterion and wafer rotation on the compound yield. Thereafter, we investigate the wafer pollution that could take place in running repositories.

Impact of matching process

We compare the performance of the matching process AL;WbW and WbW;LbL by considering one scenario for each process. For the AL;WbW based scenario, the max(MG) criterion will be used as it is the only one that can be combined with such a process (see Section 4.2). On the other hand, there are three possible WbW;LbL based scenarios; i.e., BP, FIFO1 and FIFOn (see Section 4.1). However, only BP based scenario will be considered for this comparison; this is because BP based scenario overall scores the best as will be discussed in the next subsection. Moreover, BP will be used first with max(MG) criterion (referred to as *Default BP*) and thereafter with the best matching criterion that results in the best yield improvement (referred to as *Adaptive BP*). Note that adaptive BP automatically selects the best matching criterion to be combined with the BP process and which results in the best yield improvement.

The simulations performed for the comparison are restricted to only a stack size of n=3 only and a repository size of k=25. This is because of the time and memory complexity of AL;WbW based scenario. It is worth noting that for n=2, AL;WbW based scenario and WbW;LbL default BP based scenario perform exactly the same steps, resulting into the same compound yield improvement.

The simulation results for AL;WbW based scenario for different die yield values are shown in the left block of Table I; the second column in the table give the relative yield improvement of AL;WbW over random stacking ($\frac{\text{AL;WbW}}{\text{random}}$). As expected, the lower the die yield, the higher the yield improvement.

The right block of Table I shows the relative yield improvement of AL;WbW with respect to default BP scenario ($\frac{\text{AL;WbW}}{\text{Default BP}}$) and with respect to adaptive BP ($\frac{\text{AL;WbW}}{\text{Adaptive BP}}$). The table clearly reveals that for lower die yield, AL;WbW outperforms default BP. However, for higher die yield adaptive BP scenario performs better. The reason behind this is that AL;WbW works only with the max(MG) criterion (the remaining criteria are not applicable when more than two layers are considered simultaneously), while adaptive BP select the best criterion resulting in the highest yield improvement.

The differences in compound yield improvement, for realistic die yields ($Y_D \geq 50$), between the WbW;LBL and AL;WbW matching processes are very minor. In fact, adaptive BP outperforms AL;WbW in some cases. In addition, the time and memory complexity of the WbW;LbL process is significantly lower and therefore, only this matching process is considered onwards.

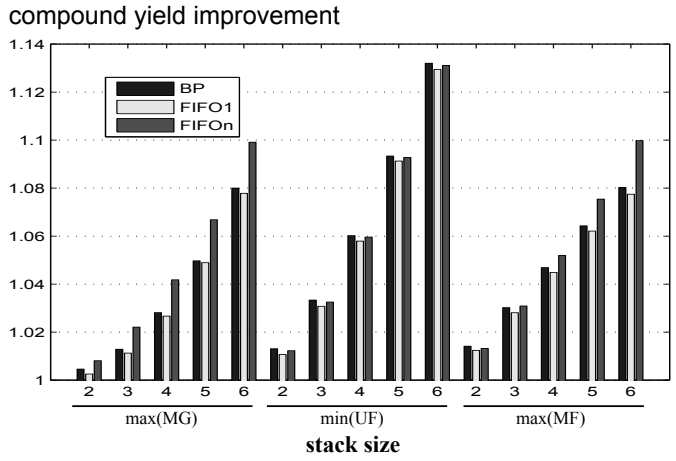


Fig. 13. Compound yield for different stack size n and matching criterion.

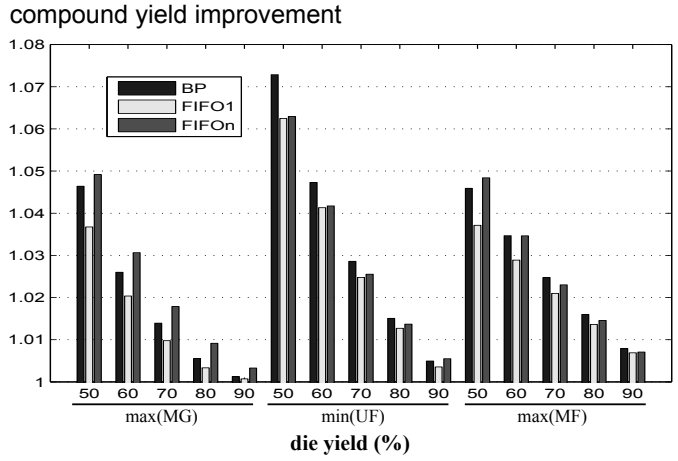


Fig. 14. Compound yield for different die yield Y_D and matching criterion.

Impact of matching criteria

In order to investigate the impact of matching criteria we simulate the three scenarios based on the WbW;LbL process (i.e., BP, FIFO1 and FIFOn) combined with the three matching criteria max(MG), min(UF), max(MF); see Section 2.3. The simulation is performed first for different stack size n and thereafter for different die yield Y_D.

Figure 13 show the simulation results for variable stack size using reference process. One can conclude the following:

— The higher the stack size, the higher the yield improvement. Note that when the number of stack size increases, the compound yield decreases.

Table II. Yield Based Criterion Selection.

$Y_{D,t} \backslash Y_{D,b}$	10	20	30	40	50	60	70	80	90
10	MG	MG	MG	UF	UF	UF	MG	MG	MG
20	MG	MG	UF	UF	UF	UF	UF	MF	MF
30	MG	UF	UF	UF	UF	UF	UF	UF	MF
40	UF	UF	UF	UF	UF	UF	UF	UF	UF
50	UF	UF	UF	UF	UF	UF	UF	UF	UF
60	UF	UF	UF	UF	UF	UF	UF	UF	UF
70	MG	UF	UF	UF	UF	UF	UF	UF	MF
80	MG	MG	UF	UF	UF	UF	UF	MF	MF
90	MF	MF	MF	UF	UF	UF	MF	MF	MF

— FIFOn always performs better than FIFO1. This difference in performance is minimal for min(UF) criterion; this means that in this case a small pollution is taking place (see next section).
— The best criterion resulting in maximal yield improvement is stack size dependent. For the simulated reference process (where die yield is 80%), for $n{=}2$ the max(MF) criterion performs the best, for $n{\geq}3$ min(UF) performs the best. However, in both cases, the criterion scores the best when it is combined with BP scenario.

Figure 14 shows the simulation results for die yield using reference process with a fixed repository size of $k = 50$. One can conclude the following:

— The lower die yield, the higher the yield improvement. Obviously, the compound yield increases when die yield increases.
— FIFOn always performs better than FIFO1. This difference in performance is minimal for min(UF) criterion; this means that in this case a small pollution is taking place (see next section).
— The best criterion resulting in maximal yield improvement is die yield dependent. For die yield $50{\leq}Y_D{\leq}70$, min(UF) combined with BP scores the best, while for $Y_D{\geq}80$ max(MF) combined with BP scores the best. Note that also in this case, BP scenario combined with appropriate criterion is the one that results in the best yield improvement.

The above results clearly show that BP scenario outperforms both FIFO1 and FIFOn when combined with appropriate criterion. The question now rises which matching criterion has to be used - given certain die yield (and stack size)- to maximize the compound yield. Table II answers this question when assuming that a top die with a yield $Y_{D,t}$ and bottom die with a yield $Y_{D,b}$ have to be stacked together to realize the highest compound yield; the table shows which criterion has to be selected with BP for given $Y_{D,t}$ and $Y_{D,b}$. From the table one can conclude the following:

— When the die yield is low, the max(MG) criterion should be selected. This criterion targets the matching of good dies only. Since these are in minority, the choice to select the best matching is relatively easy.
— For die yield in midrange values, the min(UF) criterion performs the best. In this case, the probability of the presence of good and bad dies is similar.
— For very high die yield, it is most advantageous to use the max(MF) criterion. In this case, the matching is based on faulty dies. As the faulty dies are in minority due to a high die yield, an overall highest compound yield is obtained if the matching of the minority dies is maximized.

The above clearly shows that an *adaptive* BP-based wafer matching is the best approach to realize the maximal overall compound yield. Table II can be used as a deci-

Table III. Impact of wafer rotation on adaptive BP for variable n.

stack size n	$\frac{\text{half-rotation}}{\text{no-rotation}}$ (%)	$\frac{\text{quarter-rotation}}{\text{no-rotation}}$ (%)
2	0.15	0.27
3	0.37	0.67
4	0.68	1.23
5	0.97	1.88
6	1.31	2.55

Table IV. Impact of wafer rotation on adaptive BP for variable Y_d.

die yield Y_D	$\frac{\text{half-rotation}}{\text{no-rotation}}$ (%)	$\frac{\text{quarter-rotation}}{\text{no-rotation}}$ (%)
50	0.56	1.18
60	0.34	0.75
70	0.29	0.48
80	0.16	0.31
90	0.08	0.14

sion rule for the matching criterion selection. Each time a new wafer has to be selected for stacking, the table determines the matching criterion to be used. As an example, consider a three layered stack with equal die yield of 80%. According to Table II, matching of the bottom with middle wafers is performed best by using the Max(MF) criterion. If we assume now that the compound yield of this two-stacked IC is 70%, then the matching with the third layer can be best performed based on the min(UF) criterion. This adaptive BP scenario always results in the highest yield for all simulation parameters. From now on, we refer to adaptive BP as the matching scenario that adapts itself with respect to the criterion selection.

Impact of wafer rotation

We measure the impact of wafer rotation for the reference process, but using the *adaptive* BP scenario for different stack sizes and die yield. The simulation is performed for no-rotation, half-rotation and quarter-rotation; the improvement of using wafer rotation with respect to no-rotation is measured. It worth noting that in case of quarter-rotation (rotation of 90^0, 180^0 or 270^0; see Section 2.4) all rotation combinations are tried and each time the best combination is selected.

Table III shows the compound yield improvement in percentage for half-rotation and quarter-rotation with respect to no-rotation, for a stack size $2 \leq n \leq 6$. The table shows that the added value of wafer rotation is marginal (1.31% and 2.55% for a stack size of 6 layers for half- and quarter-rotation respectively).

Table IV shows the results for a die yield between 50 and 90%. Again, only marginal improvement is obtained using wafer rotation (0.56% and 1.18% for a die yield of 50% for half- and quarter-rotation respectively). For higher die yields, the yield improvement even reduces.

6.3. Repository pollution

In order to estimate the repository pollution, the compound yield for different production sizes is simulated. It is hard to directly measure the impact on pollution when using adaptive BP. Nevertheless, we can indirectly measure such an impact by investigating the pollution issue for FIFO1 and FIFOn scenarios. Obviously, the impact of FIFO1 on repository pollution is the worst because it will never force the bad wafers to get out from the repositories except for the first one. On the other hand, the impact of FIFOn on repository pollution is minimum because it will always force all wafers to

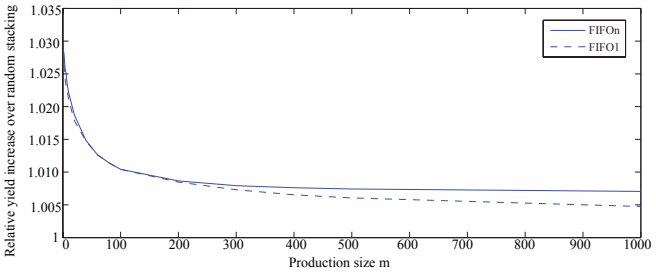


Fig. 15. Yield versus production sizes.

get out from the repositories. Hence, the impact of BP on repository pollution will be somewhere between.

Figure 15 plots the relative compound yield for the FIFO1- and FIFOn-based matching processes over random stacking for different production size m. Here, the reference process is used with n=2, k=25 and the matching criterion max(MG). The die yield Y_D is assumed to have a Gaussian distribution with a mean of 80%. Three observations can be made from the graph:

— The relative yield for both FIFO1 as well as FIFOn decreases with increasing production size. For low m (typically below 200), the yield for both scenarios is almost the same.
— As the size of m increases (hence probability of having bad wafers increases), the difference in yield between FIFOn and FIFO1 becomes more visible but still marginal for this case (about 0.5% difference). The compound yield of FIFO1 decreases faster than that of FIFOn; FIFOn forces wafers to leave the repository at most after $k \cdot n$ cycles and this has a positive affect on the yield.
— The yield degradation due to pollution is stabilizing for larger m.

Obviously, implementing a mechanism to force the wafers to leave the repositories after a certain time period is useful to increase the overall compound yield. However, for a mature process -with a typically die yield above 80%- the impact of repository pollution is minimal.

7. COMPARISON OF MATCHING SCENARIOS

In this section, we evaluate the analyzed matching scenarios in this work with the related prior work that use static repositories. Due to the high run-time complexity of the AL;WbW matching scenario and the very marginal improvements with respect to WbW;LbL (only for very low die yield), we only consider the adaptive Best Pair (BP) scenario for comparison. First, we compare our yield improvement results with the Greedy scenario [Verbree et al. 2010] and UB scenario [Reda et al. 2009] that both do not consider wafer rotation. Thereafter, we compare our results with the Greedy scenario [Singh 2011] that utilizes quarter-rotation. Finally, the time and memory complexity of the proposed scheme as well as prior work will be analyzed.

Table V and Table VI show the comparison results between BP matching scenario and Greedy scenario [Verbree et al. 2010] for variable stack size and die yield, respectively, for the number of dies per wafer d=1278 and k=50. The first column in each table provides the variation parameter of the simulation (i.e., stacked number of layers n and the die yield y); the second column reports the compound yield of the related work; the third column reports the compound yield of [Verbree et al. 2010] using adaptive

Table V. Yield comparison with [Verbree et al. 2010] for Y_D=81.61%.

n	Greedy [Verbree et al. 2010] (%)	Mod. Greedy (%)	BP (%)	$\frac{\text{BP}}{\text{Greedy } [Verbreeet\ al.2010]}$ (%)	$\frac{\text{BP}}{\text{random}}$ (%)
2	67.3	67.3	67.56	0.4	1.43
3	55.7	55.8	56.24	1.0	3.47
4	46.3	46.5	47.10	1.7	6.18
5	38.7	38.9	39.65	2.5	9.53
6	32.5	32.7	33.50	3.1	13.39

Table VI. Yield comparison with [Verbree et al. 2010] for n=2.

die yield (%)	Greedy [Verbree et al. 2010] (%)	Mod. Greedy (%)	BP (%)	$\frac{\text{BP}}{\text{Greedy } [Verbreeet\ al.2010]}$ (%)	$\frac{\text{BP}}{\text{random}}$ (%)
50	26.2	26.3	26.82	2.4	7.28
60	37.2	37.2	37.71	1.4	4.75
70	50.0	50.0	50.40	0.8	2.86
80	64.7	64.8	64.96	0.4	1.50
90	81.3	81.5	81.64	0.4	0.79

Table VII. Yield Comparison with [Reda et al. 2009] for Y_D=80%.

n	Scenario [Reda et al. 2009]	Compound Yield [Reda et al. 2009] (%)	BP (%)	$\frac{\text{BP}}{\text{Scenario } [Redaet\ al.2009]}$ (%)	$\frac{\text{BP}}{\text{random}}$ (%)
2	UB	65.25	65.32	0.11	2.06
3	UB	53.56	53.76	0.37	5.00
4	UB	44.58	44.63	0.11	8.96
5	IMH	36.61	37.28	1.83	13.77
6	IMH	30.68	31.29	1.99	19.36
7	IMH	25.76	26.35	2.29	25.65

Table VIII. Yield Comparison with [Reda et al. 2009] for $n = 3$.

die yield (%)	UB [Reda et al. 2009] (%)	BP (%)	$\frac{\text{BP}}{\text{UB } [Redaet\ al.2009]}$ (%)	$\frac{\text{BP}}{\text{random}}$ (%)
30	04.24	04.30	1.42	59.26
50	15.08	15.24	1.06	21.92
70	37.29	37.46	0.46	9.21
90	74.41	74.46	0.07	2.14

matching criteria; the fourth column presents the compound yield of the adaptive BP scenario; the fifth column shows the relative improvement of the BP scenario versus the obtained yield of the related work; the last column shows the improvement of the BP scenario relative to random stacking. The table clearly shows that BP scenario (slightly) outperforms the greedy approach in [Verbree et al. 2010] and significantly outperforms the random stacking approach; the higher the stack size and/or the lower the die yield, the higher the improvement.

Next, we compare the optimal UB scenario [Reda et al. 2009] with our adaptive BP scenario. In case the UB scenario is inapplicable due to memory limitations, the IMH scenario is used [Reda et al. 2009]. Table VII and VIII show the results for variable stack size and die yield, where k=25 and d=590. In each table, the first column provides the varied parameter of the simulation; the second column reports the compound yield of the related work; the third column presents the compound yield of the adaptive BP scenario; the fourth column shows the relative improvement of the BP scenario versus the obtained yield of the related work; the last column shows the improvement of the BP scenario relative to random stacking. From the tables, we can clearly conclude that BP scenario (slightly) outperforms UB and

Table IX. Yield Comparison with [Singh 2011] for $Y_D = 81.67\%$.

stack size (%)	Greedy [Singh 2011] (%)	BP (%)	$\frac{\text{BP}}{\text{Greedy } [Singh2011]}$ (%)	$\frac{\text{BP}}{\text{random}}$ (%)
2	67.7	67.84	0.02	01.70
4	47.4	47.81	0.85	07.45
6	33.9	34.46	1.66	16.14

Table X. Time and Memory Complexity of the Considered Scenarios.

Ref	**Scenario**	**Run-time complexity**	**Memory complexity**
[Reda et al. 2009]	Greedy	$O(m \cdot k^{n-1} \cdot log(k))$	$O(n \cdot k^n)$
[Reda et al. 2009]	IMH	$O(m \cdot n^2 \cdot k^2)$	$O(k^2)$
[Reda et al. 2009]	ILP/UB	$O(\frac{m}{k} \cdot (k!)^{n-1})^*$	$O(n \cdot k^n)$
[Verbree et al. 2010; Singh 2011]	Greedy	$O(m \cdot k^2 \cdot n)$	$O(k^2)$
Ours	FIFO1	$O(m \cdot k \cdot n)$	$O(n)$
Ours	FIFOn	$O(m \cdot k \cdot n)$	$O(n)$
Ours	Best Pair	$O(m \cdot k \cdot n)$	$O(k^2 + n)$
Ours	AL;WbW	$O(m \cdot k^{(n-1)})$	$O(k^n)$

*denotes the complexity of the search space

significantly outperforms random stacking for all considered stack sizes and die yields.

Lets now compare our results with prior work considering also wafer rotation. Table IX presents the results for quarter-rotation for running repositories with the Greedy scenario for static repositories proposed in [Singh 2011] and compares them with adaptive BP combined with quarter-wafer rotation. The simulation is done for k=50, d=590 and $Y_D = 81.67$ as in [Singh 2011]. Again, the results show that running repositories slightly outperform static repositories, and that in which the improvement increases with stack size.

Overall, Table V to Table IX show that running repositories slightly outperform static repositories (up to 2.29%). Nevertheless, the greatest advantage of the proposed approach is the significant time and memory complexity reduction. Table X summarizes the time complexity and memory complexity for the different scenarios discussed in this paper. As already shown, BP scenario is the best to use in order to realize the highest compound yield. The time complexity of such approach is only $O(m \cdot k \cdot n)$; while that of Greedy [Verbree et al. 2010; Singh 2011] is $O(m \cdot k^{n-1} \cdot log(k))$ and ILP/UB [Reda et al. 2009] is $O(\frac{m}{k} \cdot (k!)^{n-1})$. To get more insight in the significant time complexity reduction, let us consider the following example. The optimal static scenario in [Reda et al. 2009] implemented in C++ requires 0.392 seconds to solve an instance for $n = 3$ and 40.64 seconds for n=4, and it runs out of memory (which is 2 GB) for larger number of stacked layers [Reda et al. 2009]. The authors used an Intel Core 2 Duo Extreme edition processor running at 2.93GHz with 2GB of dynamic memory for their experiments. For the same parameters (n=4 etc.), our adaptive BP scenario implemented in Matlab requires only 0.04 seconds while using a negligible amount of memory and achieves a time reduction of three order of magnitudes for a comparable machine, i.e., an Intel Core 2 Duo processor running at 2.93GHz with 3GB of dynamic memory.

8. CONCLUSION

In this paper, wafer matching as a mean for yield improvement for 3D W2W stacked ICs is presented. A complete framework covering all possible wafer matching scenarios is defined; a matching scenario combines four parameters (1) matching process, (2)

matching criterion, (3) wafer rotation and (4) repository type. Different scenarios are analyzed to evaluate their importance.

The results show that using exhaustive matching processes is not necessary needed to realize optimal yield improvements; matching processes with the lowest time and memory complexity can result in similar/same improvement when combined with appropriate matching criterion. The best matching criterion to be used for optimal compound yield improvement is strongly stack size and die yield dependent; hence using adaptive matching criterion selection is the optimal solution. Moreover, the results show that running repositories outperform static repositories irrespective of the design and manufacturing parameter values (e.g. stack size, die yield) and that the added value of wafer rotation is negligible.

REFERENCES

V.D. Agrawal. 2000. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer. http://books.google.nl/books?id=CgnLg99GumMC

J. Baliga. 2004. Chips go vertical [3D IC interconnection]. *Spectrum, IEEE* 41, 3 (March 2004), 43–47. DOI:http://dx.doi.org/10.1109/MSPEC.2004.1270547

J.A. Davis, R. Venkatesan, Alain Kaloyeros, M. Beylansky, S.J. Souri, K. Banerjee, K.C. Saraswat, Arifur Rahman, Rafael Reif, and J.D. Meindl. 2001. Interconnect limits on gigascale integration (GSI) in the 21st century. *Proc. IEEE* 89, 3 (Mar 2001), 305–324. DOI:http://dx.doi.org/10.1109/5.915376

W.R. Davis, J. Wilson, S. Mick, J. Xu, Hao Hua, C. Mineo, A.M. Sule, M. Steer, and P.D. Franzon. 2005. Demystifying 3D ICs: the pros and cons of going vertical. *Design Test of Computers, IEEE* 22, 6 (Nov 2005), 498–510. DOI:http://dx.doi.org/10.1109/MDT.2005.136

D.K. De Vries. 2005. Investigation of gross die per wafer formulas. *Semiconductor Manufacturing, IEEE Transactions on* 18, 1 (Feb 2005), 136–139. DOI:http://dx.doi.org/10.1109/TSM.2004.836656

Cesare Ferri, Sherief Reda, and R. Iris Bahar. 2008. Parametric Yield Management for 3D ICs: Models and Strategies for Improvement. *J. Emerg. Technol. Comput. Syst.* 4, 4, Article 19 (Nov. 2008), 22 pages. DOI:http://dx.doi.org/10.1145/1412587.1412592

Philip Garrou, Christopher Bower, and Peter Ramm. 2008. *Handbook of 3D Integration: Volume 1 - Technology and Applications of 3D Integrated Circuits*. Wiley.

Feihul Li, C. Nicopoulos, T. Richardson, Yuan Xie, V. Narayanan, and M. Kandemir. 2006. Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. In *Computer Architecture, 2006. ISCA '06. 33rd International Symposium on*. 130–141. DOI:http://dx.doi.org/10.1109/ISCA.2006.18

Gabriel H. Loh, Yuan Xie, and Bryan Black. 2007. Processor Design in 3D Die-Stacking Technologies. *Micro, IEEE* 27, 3 (May 2007), 31–48. DOI:http://dx.doi.org/10.1109/MM.2007.59

R.S. Patti. 2006. Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs. *Proc. IEEE* 94, 6 (June 2006), 1214–1224. DOI:http://dx.doi.org/10.1109/JPROC.2006.873612

K. Puttaswamy and G.H. Loh. 2009. 3D-Integrated SRAM Components for High-Performance Microprocessors. *Computers, IEEE Transactions on* 58, 10 (Oct 2009), 1369–1381. DOI:http://dx.doi.org/10.1109/TC.2009.92

S. Reda, Gregory Smith, and Larry Smith. 2009. Maximizing the Functional Yield of Wafer-to-Wafer 3-D Integration. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 17, 9 (Sept 2009), 1357–1362. DOI:http://dx.doi.org/10.1109/TVLSI.2008.2003513

E. Singh. 2011. Exploiting rotational symmetries for improved stacked yields in W2W 3D-SICs. In *VLSI Test Symposium (VTS), 2011 IEEE 29th*. 32–37. DOI:http://dx.doi.org/10.1109/VTS.2011.5783751

E. Singh. 2012. Impact of Radial defect clustering on 3D stacked IC yield from wafer to wafer stacking. In *Test Conference (ITC), 2012 IEEE International*. 1–7. DOI:http://dx.doi.org/10.1109/TEST.2012.6401567

G. Smith, Larry Smith, S. Hosali, and S. Arkalgud. 2007. Yield considerations in the choice of 3D technology. In *Semiconductor Manufacturing, 2007. ISSM 2007. International Symposium on*. 1–3. DOI:http://dx.doi.org/10.1109/ISSM.2007.4446880

M. Taouil, S. Hamdioui, J. Verbree, and E.J. Marinissen. 2010. On maximizing the compound yield for 3D Wafer-to-Wafer stacked ICs. In *Test Conference (ITC), 2010 IEEE International*. 1–10. DOI:http://dx.doi.org/10.1109/TEST.2010.5699218

Yuh-Fang Tsai, Feng Wang, Yuan Xie, N. Vijaykrishnan, and M.J. Irwin. 2008. Design Space Exploration for 3-D Cache. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 16, 4 (April 2008), 444–455. DOI:http://dx.doi.org/10.1109/TVLSI.2007.915429

J. Verbree, E.J. Marinissen, P. Roussel, and D. Velenis. 2010. On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking. In *Test Symposium (ETS), 2010 15th IEEE European*. 36–41. DOI:http://dx.doi.org/10.1109/ETSYM.2010.5512785

Sixteenth IEEE European Test Symposium

Layer Redundancy Based Yield Improvement for 3D Wafer-to-Wafer Stacked Memories

Mottaqiallah Taouil Said Hamdioui
Computer Engineering Laboratory
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: {M.Taouil, S.Hamdioui}@tudelft.nl

***Abstract*— Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs) such as memories based on Wafer-to-Wafer (W2W) stacking. One of the major challenges facing W2W stacking is the low compound yield, especially for larger stack sizes. This paper investigates compound yield improvement for W2W stacked memories using layer redundancy. First, a analytical model is provided to prove the added value of layer redundancy. Second, the impact of such a scheme on the manufacturing cost is evaluated. Finally, these two parts are integrated to analyze the trade-off between yield improvement and its associated cost; the realized yield improvement is also compared to yield gain obtained when using wafer matching. The simulation results show that for higher stack sizes layer redundancy realizes a significant yield improvement as compared to wafer matching, and at even lower cost. For example, for a stack size of six layers and a die yield of 85%, a relative yield improvement of 82.46% is obtained using one redundant layer, while this is 10.27% with wafer matching. The additional cost due to redundancy pays off; the cost of producing a good 3D stacked memory chip reduces with 38.45% when using layer redundancy and only with 10.27% when using wafer matching. Moreover, the results show that the benefits of layer redundancy become extremely significant for lower die yields.**

Keywords: *3D stacked-IC, yield enhancement, memory redundancy, 3D memory.*

I. Introduction

The increasing demand for more functionality on ICs has been met by the semiconductor industry adhering to Moore's law. Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs), which are electrically interconnected by Through Silicon Vias (TSV). This opened up new research directions that could be investigated to continue the trend of performance increase. A TSV based 3D-SIC is an emerging technology that provides a smaller footprint, higher interconnect density between stacked dies, higher performance and lower power consumption due to shorter wires as compared to planar ICs [1]. Moreover, heterogeneous integration in 3D-SICs allows dies to be manufactured with dissimilar processing and technology nodes. For example, memory layers can be stacked on a processor [2].

The key manufacturing steps in assembling 3D-SICs are the stacking and bonding of dies. The three existing bonding methods are Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) and Die-to-Die (D2D) bonding. W2W stacking allows for (a) high manufacturing throughput due to single wafer alignment, (b) thinned wafers and small die handling, and (c) the ability to create a higher TSV density. High alignment accuracy can be applied in D2W and D2D bonding, but it negatively affects the throughput due to many dies that have to be aligned. In D2W and D2D bonding, Known Good Die (KGD) stacking can be applied to prevent faulty dies from being stacked [3]. Due to their similar size and regularity, stacked memories are very attractive to W2W stacking. However, one of the major drawback of W2W stacking is a yield decrease with increased number of stacked layers.

Traditionally, memory yield improvement in 2D chips is realized by using spare rows and/or columns to repair defective ones. 3D stacked memories allow the exploration of new repair schemes that take advantage of the vertical dimension, e.g., inter-layer redundancy and layer redundancy. In inter-layer redundancy, if a memory layer is not repairable because the number of defective rows and/or columns is more than the spares, then additional resources (spares) from the neighboring layers could be borrowed and used. A drawback of this approach is the additional required number of TSVs and the routing complexity to mutually share and access the spare resources among the layers in the stack. The second scheme, layer redundancy, can be applied at the wafer level. Additional redundant layer(s) are stacked to replace the faulty irreparable memory dies in the stack.

This paper investigates layer redundancy as a mean for compound yield improvement for 3D W2W stacked memories. The main contributions of this paper are:

- A classification of 3D memories and 3D memory redundancy repair schemes.
- An analytical model that formulates the yield gain as a result of layer redundancy.
- A comparison of 3D W2W stacked memories with and without layer redundancy in terms of the cost of producing good 3D stacks.
- A memory layer replacement circuit that modifies addresses of faulty memory layer(s) to the spare layer(s).

The remainder of this paper is organized as follows. Section II provides prior work related to W2W 3D stacked

1530-1877/11 $26.00 © 2011 IEEE
DOI 10.1109/ETS.2011.55

memories and yield improvement including 3D memory architectures, wafer matching and a classification of 3D memory redundancy. Section III introduces an analytical yield model for 3D stacked memories both with and without layer redundancy. Section IV gives the simulation results and shows the superiority of layer redundancy. Section V proposes a layer replacement scheme for 3D stacked memories. Section VI concludes this paper.

II. Related Prior Work

This section describes previous work in W2W stacking and 3D stacked memories. Section II-A provides a brief overview of 3D memory architectures, and highlights the targeted architecture in this paper; note that the work presented here can be extended to any possible 3D memory architecture. Subsequently, Section II-B describes wafer matching, a general technique to increase the W2W compound yield. Finally, Section II-C presents a classification of possible memory redundancy schemes.

A. 3D memory architectures

Partitioning memories across multiple device layers can take place at different granularities, resulting in different architectures. A top to bottom perspective is presented in the following.

- **Stacked banks** - The coarsest granularity partitioning of memory takes place at the bank level, by stacking banks on the top of each other. Each bank consist of a complete memory system (i.e., memory cell array, address decoder, write drivers, etc.). An overall reduction in wire length is obtained (about 50 percent for certain configurations), resulting into significant reduction in both power and delay [5,6]. A 3D manufactured DRAM based on the stacking of banks is realized in [7].
- **Cell arrays stacked on logic** - This approach, in contrast to the previous one, separates the peripheral logic (row decoders, sense amplifier, column select logic, etc), from the cell arrays. The peripheral logic is placed on the bottom layer while the cell array is split across multiple layers. This is considered to be the *true 3D memory* [5]. Research in this area has been performed for both SRAMs [5,9] and DRAMs [8,10]. By using this separation method, the peripheral logic can be optimized independently for speed, while the cell arrays can be arranged to meet different criteria (density, footprint, thermal, etc). 3D manufactured DRAM based on the stacking of cell arrays on logic is realized in [11,12]. A classification within the array layer can also be made.
  - **divided-columns**: in which bitlines are divided and mapped onto different layers;
  - **divided-rows**: in which wordlines are divided and mapped onto different layers, requiring one die-to-die TSV per wordline.

  Both organizations reduce latency and power due to reduced wordline/bitline lengths.

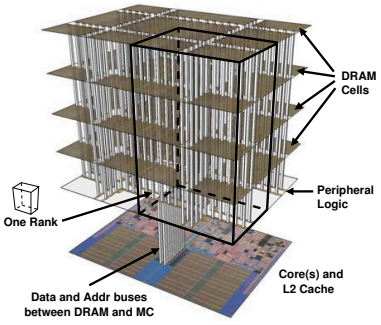


Fig. 1. *Cell arrays stacked on logic* 3D stacked DRAM [10].

- **Intra-cell (bit) partitioning** - Here, memory cells are split among one or more layers. At this fine granularity level, the relative small size of the cell and the size of the TSV make the splitting across layers a difficult task [9]. Nevertheless, the authors in [5] suggest that this option can be feasible for multi-port SRAM arrays, such as register files.

Until now, most research focused primarily on latency and power improvements. Redundancy in 3D memories is gaining more popularity and requires further research.

The targeted memory architecture in this paper is *cell arrays stacked on logic*. An example of such an architecture is show in Figure 1 [10]. This architecture makes heterogeneous integration feasible. For example memory layers manufactured in DRAM process technology optimized for area can be stacked on the peripheral circuits manufactured in a logic process optimized for speed. In the figure, the top four memory layers contain the memory array, while the layer thereunder contains the peripheral circuitry; the complete memory is stacked on the processor layer.

B. W2W matching

The yield impact in Wafer-to-Wafer stacking has been researched by several authors [13–16]. The 3D-SIC yield decreases exponentially with increased stack size. The authors suggest wafer matching to mitigate this yield drop. Wafer matching is a technique to improve the yield by stacking wafers with similar fault maps. In case of high stack sizes or low die yields significant improvements can be realized. However, in case the die yield is high this technique is not sufficient enough. For example, for a stack size of two layers with a die yield of 85% and 1278 dies per wafer, wafer matching is able to increase the stack yield from 72.3% (for random W2W stacking) to 73.1% [16] only.

Wafer matching could not be applicable for *cell arrays stacked on logic* architecture, see Figure 1. Wafer matching requires wafer tests prior to stacking. Due to the absence of peripheral circuits on the memory layers, pre-bond wafer tests can not be performed. Nevertheless, the memory of Figure 1 has the advantage that a single BIST engine can be placed on the peripheral layer, and shared by all array

layers.

C. 3D memory redundancy

Traditionally, yield improvement for 2D memories is based on the use of spare rows and/or columns [17]. 3D stacked memories, however, provide additional repair features due to the vertical dimension. The redundancy schemes for 3D memories can be classified into 3 groups.

1) Intra-layer redundancy: Redundancy within each layer is similar to that in planar memories. Each layer may have spare rows and/or columns that can be used within the same layer to improve the yield.
2) Inter-layer redundancy: In inter-layer redundant memories, spare rows and/or columns can not be accessed only from the die they belong to, but also from neighbor dies. Hence, they can borrow additional resources in case they run out of their own. Tezzaron memories are examples of memory architectures that use inter-layer redundancy [2]. In [18], inter-layer redundancy is used by the authors to increase the stacked memory yield for different allocation algorithms.
3) Layer redundancy: Redundancy at the wafer or die level. A faulty irreparable memory layer is disabled and instead is replaced with a complete redundant layer. A memory layer is not repairable if the required number of spares exceed the existing spares within it.

In this paper, we analyze the yield increase based on layer redundancy.

III. Yield and Cost Modeling

Before presenting the yield and cost models to measure and evaluate the impact of layer redundancy, first the yield parameters used in this work and made assumptions will be presented.

A. Process parameters and assumptions

In order to accurately evaluate the memory yield improvement due to layer redundancy, different process parameters have to be appropriately chosen. A 3D stacked memory consists of multiple stacked layers/dies interconnected by TSVs. Each die in the stack can be either faulty or non-faulty (i.e., functional). Similar to what is reported in the ITRS roadmap [20], we will assume the die yield Y_D of the memory dies to be Y_D=85%. In addition to die yield, new possible defects introduced during the stacking process have to be taken into consideration. Dies/layers that enter the stack could get corrupted e.g., due to bonding and thinning. The new introduced faults due to stacking are modeled by the stacked-die yield Y_{SD} and this parameter is assumed to be 99% [15]. For the TSVs, we assume an interconnect yield Y_{INT} to be 97% per stacked layer [15]. In this paper, the yield parameters are assumed to be fixed unless otherwise stated. Other parameters that influence the stack yield are the stack size n and the number of redundant layers r. The complete stack *size* is denoted by s=$n + r$. The overall compound yield of the stack is denoted by Y.

Faults that are introduced during the stacking process are unrecoverable (i.e, faults introduced into the dies and interconnects). The following assumptions are made in this paper with respect to the stack yield parameters:

- The memory layers in the stack are considered to be independent; each layer can be either faulty or non-faulty.
- Since many TSVs are shared (e.g., for address or data buses), it is assumed that any malfunction in communication between two layers results in faulty stacked memory.
- A failure due to 3D stacking (due to e.g., thinning, bonding, etc) will result in a faulty stacked IC.
- We do not consider the peripheral circuit layer in the model to-be-presented as it impacts both 3D stacked memories with or without layer redundancy in a similar way.

B. Yield modeling

The model will be presented first for 3D stacked memories without layer redundancy and thereafter for those with layer redundancy.

a) Memories without layer redundancy: In case there is no redundancy, i.e. s=n and r=0, each layer in the stack must operate to ensure memory functionality. The compound yield $Y(n)$ can be described as a function of the die yield and stack size. Besides the dies, also the interconnects and the 3D bonding must be fault free. This leads to the following yield expression for non-redundant memories.

$$Y(n) = Y_D^n \cdot Y_{SD}^{n-1} \cdot Y_{INT}^{n-1} \tag{1}$$

Note that 3D stacked memory with n layers requires $n-1$ stacking steps.

b) Memories with layer redundancy: In this case, r redundant layers are appended to the stack with n layers resulting in a total layers of $s = n+r$. If n or more layers out of the stacked s layers are functionally correct, then the final 3D-SIC is assumed to be non-faulty. The probability $p(i)$ that i layers out of s layers are non-faulty can be formulated by the binomial expression:

$$p(i) = \binom{s}{i} \cdot Y_D^i \cdot (1 - Y_D)^{s-i} \tag{2}$$

We extend the symbol $Y(n)$ for non-redundant memories, to $Y(n, s)$. The yield $Y(n, s)$ expresses the yield of an s layered stack with r=s−n redundant layers. The yield for layer redundant enabled memories can be expressed now by:

$$Y(n,s) = \left(\sum_{i=n}^{s} p(i)\right) \cdot Y_{SD}^{s-1} \cdot Y_{INT}^{s-1} = \left(\sum_{i=n}^{s} \binom{s}{i} \cdot Y_D^i \cdot (1 - Y_D)^{s-i}\right) \cdot Y_{SD}^{s-1} \cdot Y_{INT}^{s-1} \tag{3}$$

In order for the stack to be considered defect-free, at least n out of s layers must be defect-free. Note that the redundant layers can be faulty as well. Equation 1 and 3 are equivalent in case n=s, i.e., in case there is no layer redundancy.

TABLE I
YIELD IMPROVEMENT GAINED BY REDUNDANCY IN PERCENTAGE.

	n=1	n=2	n=3	n=4	n=5	n=6
Abs. yield	85.00	69.38	56.63	46.23	37.73	30.80
r=1	**10.43**	24.84	39.24	53.65	68.05	82.46
r=2	n.a.	**26.11**	**46.16**	**68.30**	92.50	118.79
r=3	n.a.	n.a.	43.35	67.59	**95.32**	**126.84**
r=4	n.a.	n.a.	n.a.	62.45	90.58	123.26

C. Manufacturing Cost

The question rises whether it is cost-wise justified to increase the yield by adding more redundant layers. In this section, we present the manufacturing cost for any particular stack size. The manufacturing costs $C_m(s)$ for a stack size of s can be formulated by:

$$C_m(s) = s \cdot C_w + (s-1) \cdot C_{3D} \tag{4}$$

where C_w is the wafer cost and C_{3D} the cost related to 3D stacking processes including TSV, back side processing, bonding processing, etc. Note that s wafers are needed and that the stacking process operation has to be performed $s-1$ times. Obviously, for 3D stacked memories without layer redundancy, the manufacturing cost $C_m(n)$ is:

$$C_m(n) = n \cdot C_w + (n-1) \cdot C_{3D} \tag{5}$$

IV. RESULTS

In this section, we evaluate the cost of the additional redundant layers by attributing the manufacturing cost to the good stacked ICs. First, we analyze the yield gain due to layer redundancy. Thereafter, the associated cost will be analyzed. Finally, the obtained results will be compared with those of wafer matching.

A. Yield improvement

The relative yield improvement of memories enabled with redundancy over memories without layer redundancy can be expressed by normalizing Eq. 3 over Eq. 1. The following equation describes the obtained result:

$$\frac{Y(n,s)}{Y(n)} = \frac{(\sum_{i=n}^{s} p(i))}{Y_D^n} \cdot Y_{SD}^{s-n} \cdot Y_{INT}^{s-n} = \left(\sum_{i=n}^{s} \binom{s}{i} \cdot Y_D^{i-n} \cdot (1-Y_D)^{s-i} \right) \cdot Y_{SD}^{s-n} \cdot Y_{INT}^{s-n} \tag{6}$$

Table I shows the yields for memories with and without layer redundancy. The first row gives the absolute yield (Abs. yield) of the stack without using layer redundancy. The rest of the table gives the yield improvement as a consequence of layer redundancy for different stack sizes n and different number of redundant layers r. For cost reasons it is assumed that $r \leq n$; i.e., the number of redundant layers is considered smaller than or equal to the stack size n. Each entry in the table (except the first row) lists the relative yield improvement $\frac{Y(n,s)}{Y(n)}$ (Eq. 6) in percentage for each value of n and r; entities where $r > n$ are indicated as 'n.a.' (not applicable). Inspecting the table reveals the following:

- Layer redundancy improves the memory yield irrespective of the considered stack size and number of redundant layers. The yield improvement becomes significant as the stack size increases; this is because the occurrence probability of faulty layers increases.
- Adding more redundant layers does not always result in better yield improvement. The minimum number of redundant layers that have to be added to achieve the maximal yield improvement depends in addition to n also on the process parameters under consideration such as Y_D, Y_{SD} and Y_{INT}. For example, the yield improvement for n=4 realized with r=2 is larger than that realized with r=4. This yield drop is a consequence of additional faults introduced in the larger stack due to the extra 3D processing steps.

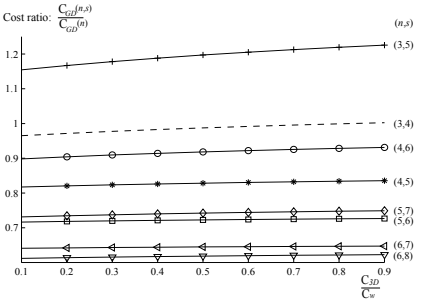


Fig. 2. Impact of layer redundancy on the cost ratio $\frac{C_{GD}(n,s)}{C_{GD}(n)}$

B. Cost Evaluation

To evaluate the additional yield gain of a redundant memory fairly, its increased manufacturing cost must be compensated for. In order to do that, we define the cost of a *good die* C_{GD} as the cost of manufacturing a good stacked IC; i.e., normalizing the manufacturing cost $C_m(s)$ to the yield. This cost for 3D stacked memory without and with layer redundancy are given in Eq. 7 and Eq. 8 respectively.

$$C_{GD}(n) = C_m(n)/Y(n) \tag{7}$$

$$C_{GD}(n,s) = C_m(s)/Y(n,s) \tag{8}$$

By using these equations, the relative improvement or depreciation of the price of a good 3D-SIC with layer redundancy over one without layer redundancy can be expressed as:

$$\frac{C_{GD}(n,s)}{C_{GD}(n)} = \frac{s \cdot \frac{C_w}{C_{3D}} + (s-1)}{n \cdot \frac{C_w}{C_{3D}} + (n-1)} \cdot \frac{Y(n)}{Y(n,s)} \tag{9}$$

Here, Eq. 4 and Eq. 5 are substituted for $C_m(s)$ and $C_m(n)$.

Figure 2 shows the above cost ratio for various values of n and s, and for $0.1 \leq \frac{C_{3D}}{C_w} \leq 0.9$, i.e., the 3D processing cost lies between 10% and 90% of the wafer cost. The following can be concluded from the figure:

- The impact of the ratio $\frac{C_w}{C_{3D}}$ on the cost ratio $\frac{C_{GD}(n,s)}{C_{GD}(n)}$ is negligible, especially for $n > 3$.
- Except for n=3 and s=5, the realized yield improvement is high enough to pay off the additional cost

TABLE II
COST RATIO $\frac{C_{GD(n,s)}}{C_{GD(n)}}$ IN % FOR VARIOUS n AND r

	n=1	n=2	n=3	n=4	n=5	n=6
r=1	208.27	125.38	**97.75**	**82.35**	**71.98**	64.31
r=2	n.a.	168.94	117.83	90.95	73.73	**61.55**
r=3	n.a.	n.a.	145.33	107.16	83.40	67.01
r=4	n.a.	n.a.	n.a.	126.89	96.48	75.85

TABLE III
IMPACT OF Y_D AND Y_S ON THE COST RATIO $\frac{C_{GD(n,s)}}{C_{GD(n)}}$

Y_D	Y_S	$\frac{C_{GD(n,s)}}{C_{GD(n)}}$ (%)	Y_D	Y_S	$\frac{C_{GD(n,s)}}{C_{GD(n)}}$ (%)
	0.91	44.10		0.91	66.16
0.6	0.95	42.25	0.8	0.95	63.37
	0.99	40.54		0.99	60.81
	0.91	52.93		0.91	88.21
0.7	0.95	50.70	0.9	0.95	84.49
	0.99	48.65		0.99	81.08

made (related to additional memory layers and stacking process). Again, this conclusion applies for our case study and the assumed process parameters. Other process parameters may result in other conclusions. Nevertheless, the figure clearly shows that generally speaking, the achieved yield improvement using layer redundancy results in lower cost per good stack.

- The larger n, the larger the impact of layer redundancy; i.e., the cheaper the cost of manufacturing a good 3D stacked memory. For example, for n=3 and r=1 (i.e., s=4), the cost reduction achieved is 2.25%, while this is 28.02% for n=5 and s=6.

Next, the impact of different values of n and r on the cost ratio $\frac{C_{GD(n,s)}}{C_{GD(n)}}$ will be analyzed. The results are summarized in Table II; it is assumed that that $\frac{C_{3D}}{C_w}$=0.3. The table shows that for n=r=1, the cost of producing a good stacked IC using layer redundancy is twice more expensive. This can be explained by the fact that adding a single redundant layer to n=1 doubles the wafer cost. The associated cost with layer redundancy starts to pay off from n=3 on. As the table shows, additional redundant layers do not always results in higher yield at *lower* cost. It strongly depends on the stack size and the number of the-to-be added redundant layers (as well as on the process parameters). Nevertheless, the larger n, the more benefits can be realized.

Another aspect which is worth to examine is the impact of the die yield Y_D and the stacking yield parameters Y_{INT} and Y_{SD} on the cost. Let $Y_S = Y_{INT} \cdot Y_{SD}$ denotes the stacking yield and assume the case where n=5 and r=1. The cost ratio $\frac{C_{GD(n,s)}}{C_{GD(n)}}$ for different values of stack yield Y_S and die yield Y_D is given in Table III; Y_D is considered to between 50% and 90%, and Y_S between 91% and 99%. The table reveals that the die yield has the highest impact on the cost ratio; the lower the die yield, the higher the benefits obtained by layer redundancy. For example, for a Y_D=60% a cost improvement of about 60% is obtained for Y_S=99%, while this does not exceed 19% for Y_D=90% for the same stack yield Y_S. Moreover, the table shows that the higher the stack yield, the higher the benefit of layer redundancy.

TABLE IV
COST REDUCTION: WAFER MATCHING VS LAYER REDUNDANCY

n	BP [16] (%)	Layer Redundancy
2	1.07	-25.38%
3	2.55	2.25%
4	4.63	17.65%
5	7.37	28.02%
6	10.27	38.45%

C. Comparison with wafer matching

This section compares cost improvements due to layer redundancy and wafer matching. Improving the yield for 3D circuits based on wafer matching was discussed by many authors [13–16]. The state-of-the art in wafer matching shows that the algorithm, referred to as the adaptive Best Pair (BP) algorithm [16], is the best in terms of yield improvement. Therefore, this algorithm will be compared with layer redundancy based yield improvement.

We will use the adaptive Best Pair (BP) algorithm for the same process parameters as that used for the evaluation of layer redundancy for a stack size $2 \leq n \leq 6$. We assume that in wafer matching the yield improvement over random stacking directly translate in the same cost improvement. This assumption is based on the fact that the manufacturing cost is independent of whether wafer matching is performed or not, while the yield improvement directly translates in cost reduction. Table IV shows the cost improvements due to wafer matching (denoted by BP) and layer redundancy respectively. It should be noted that depending on n, an optimal number of redundant layers r (realizing maximal yield improvement) is selected for the comparison. It can be seen from the table that for n=2, layer redundancy does not pay off and it results in larger cost; similar explanation can be given here as before. For larger n, however, layer redundancy is more cost-effective than wafer matching; the larger n, the larger the benefit. For example, for a six stacked IC wafer matching is able to reduce the cost with 10.27% as compared to random stacking, while layer redundancy is able to reduce this with 38.45%. However, for n=2 layer redundancy will result in an additional cost of 25.38%.

V. DESIGN FOR MEMORY REPAIR

In the previous section the cost advantages of memory layer redundancy has been shown. In this section, we will briefly discuss the different existing options to realize layer redundancy and thereafter we propose a layer replacement scheme for 3D stacked memories.

A. Traditional approaches

Redundancy for 2D memories (intra-layer) is typically performed by replacing the faulty row/column with spares. The address of the faulty row/column is stored in a programmable non-volatile device before shipping the chip to retain the information during the power off. When the

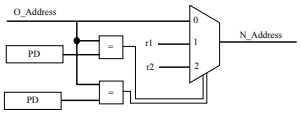


Fig. 3. Layer replacement circuit.

memory is accessed, it checks if the addressed location is faulty by comparing it to the stored faulty addresses in the programmable devices. In case faulty, the initial (faulty) location is prevented from being accessed and the spare location is activated instead.

The programmable devices can include fuses, anti-fuses or nonvolatile memory cells. Fuses may include material such as polysilicon, silicides or metals such as copper; they can be blown (programmed) by either laser or electric current. Obviously laser fusing cannot work for intra-layer redundancy if the memory cells are stacked on the top of the peripheral logic layer. Once stacked, blowing fuses by laser become not feasible. On the other hand, electrical fusing, which require no special equipment, can be applied for layer redundancy [23]. Faulty addresses can be even programmed after packaging. However, it requires an on-chip programming circuit [24]. Similarly to fuses, anti-fuses can be programmed (e.g., breaking down a dielectric) electrically [25] or by using laser pulses. Last, non-volatile memory cells can be also used to store the faulty addresses, especially for non-volatile memories such as EEPROM.

B. Layer replacement scheme for 3D stacked memories

Memory repair based on layer redundancy needs to store the ID (index) of the faulty layer in non-volatile memory. As already mentioned, this can be done with electrical fusing, electrical anti-fusing or by using nonvolatile memory cells. If the faulty layer is accessed, the repair scheme should redirect the address to a redundant layer. In the rest of this section, we will show a concept that can realize such a task. Let us assume that the size of the s memory layers are the same; hence, log(s) bits can be used to distinguish between the different layers. We assume further that the log(s) bits are the most significant bits (MSB's) of the memory address; therefore, they are unique for each layer. Figure 3 shows how this MSB's can be used to redirect the address to a redundant layer rather than the faulty layer. The programmable devices PD in the figure store the ID (i.e., the MSB's) of the faulty layers. The MSB's of the original address O_Address will be compared with the stored bits in the PD's; if a hit occurs, then the O_Address has to be mapped to the MSB's of a redundant layer (denoted by r1 and r2). For example, assume a stack size n=2 and the number of redundant layers is r=2 as in the figure, then 2 bits (=log(4)) needed as MSB's to identify the four layers. Assume further that the combinations 00 and 01 identify the layers L1 and L2 and the combination 10 and 11 identify the spare redundant layers R1 and R2. If L1 is faulty, then its access will be inhibited as the comparator will produce a hit and force the mux to select the new address r1; in this case the address is converted from 00 to 10, hence accessing the redundant layer instead of the faulty layer. Similarly, if L2 is faulty, its address will be remapped to the address r2=11.

VI. Conclusion

This paper introduces the concept of layer redundancy and investigates it as a scheme to improve the yield of 3D stacked memories. It proposes an analytical model to evaluate the yield improvement due to layer redundancy.

Simulation results show that layer redundancy not only outperforms wafer matching (as a yield improvement scheme), but also realize a significant yield improvement, especially for larger stack size. For example, a cost improvement of 38.45% is obtained in case two redundant layers are added to a stack of six layers.

References

[1] W. Rhett Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Design Test on Computers*, Vol 22, Issue 8, pp. 498-510, Nov 2005.
[2] R. S. Patty, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proc. of the IEEE*, Vol 94, Issue 6, pp. 1214-1224, June 2006.
[3] P. Garrou, Christopher Bower and Peter Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.
[4] Yuan Xie et al., "Design Space Exploration for 3D Architectures", *ACM Journal on Emerging Technologies in Computing Systems*, Vol 2, Issue 2, pp. 65-103, April 2006.
[5] K. Puttaswamy et al., "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, Vol 58, Issue 10, pp. 1369-1381, Oct. 2009.
[6] P. Ree et al. "Design Aspects of a Microprocessor Data Cache using 3D Die Interconnect Technology"' *Int. Conf. on Integrated Circuit Design and Technology* pp. 15-18, 2005.
[7] U. Kahng et al. "8 Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology"' *IEEE Journal of Solid-State Circuits*, Vol 45, pp. 111-119, 2010.
[8] R. Anigundi et al., "Architecture Design Exploration of Three-Dimensional (3D) Integrated DRAM", *Int. Symp. on Quality Electronic Design*, pp. 86-90, 2009.
[9] Y-F Tsai et al., "Design Space Exploration for 3D Cache", *IEEE Transactions on Very Large Scale Integration Systems*, Vol 16, Issue 4, 2008.
[10] G. H. Loh, "3D-Stacked Memory Architectures for Multi-Core Processors", *Int. Symp. on Computer Architecture*, pp. 453-464, Jun 2008.
[11] M. Kawano et al., "A 3D Packaging Technology for 4 Gbit Stacked DRAM with 3 Gbps Data Transfer", *Int. Electron Devices Meeting*, pp. 1-4, 2006.
[12] www.tezzaron.com
[13] L. Smith et al., "Yield Considerations in the Choice of 3D Technology", *IEEE Int. Symp. on Semiconductor Manufacturing*, pp. 535-537, 2007.
[14] S. Reda et al. "Maximizing the Functional Yield of Wafer-to-Wafer 3-D Integration", *IEEE Transactions on Very Large Scale Integration Systems*, Vol 17, Issue 9, pp. 1357 - 1362, 2010.
[15] J. Verbree et al. "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking", *IEEE European Test Symposium*, pp. 36-41, May 2010.
[16] M. Taouil et al. "On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs", *IEEE International Test Conference*, Nov. 2010.
[17] R.D. Adams, *High Performance Memory Testing - Design Principles, Fault Modeling and Self-Test*, Kluwer Academic Publishers Group, 2003.
[18] L. Jiang, R. Ye and Q. Xu, "Yield Enhancement for 3D-Stacked Memory by Redundancy Sharing across Dies", *IEEE/ACM Int. Conf. on Computer-Aided Design*, Nov. 2010.
[19] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference*, Nov. 2009.
[20] "ITRS Report Yield Enhancement 2009 Edition", *http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Yield.pdf*.
[21] N. Miyakawa, "A 3D Prototyping Chip Based on a Wafer-level Stacking Technology", *Design Automation Conference (ASP-DAC)*, pp. 416-420, 2009.
[22] M. Taouil et al. "Test Cost Analysis for 3D Die-to-Wafer Stacking", *Asian Test Symposium*, pp. 435-441, Dec. 2010.
[23] K. Lim et al. "Bit Line Coupling Scheme and Electrical Fuse Circuit For Reliable Operation of High, Density DRAM", *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 33-34, 2001.
[24] E.A. Reese et al. "A 4Kx8 Dynamic RAM with Self-refresh", *IEEE Journal of Solid State Circuits*, Vol 16, pp. 479-487, Oct. 1981.
[25] J.-K. Wee et al. "An Antifuse EPROM Circuitry Scheme for Field Programmable Repair in DRAMs", *IEEE Journal of Solid State Circuits*, Vol 35, pp. 1408-1414, Oct. 2000.

J Electron Test (2012) 28:523–534
DOI 10.1007/s10836-012-5314-3

Yield Improvement for 3D Wafer-to-Wafer Stacked Memories

Mottaqiallah Taouil · Said Hamdioui

Received: 23 October 2011 / Accepted: 26 June 2012 / Published online: 21 July 2012

Abstract Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs) such as memories based on Wafer-to-Wafer (W2W) stacking. One of the major challenges facing W2W stacking is the low compound yield. This paper investigates compound yield improvement for W2W stacked memories using layer redundancy and compares it to wafer matching. First, an analytical model is provided to prove the added value of layer redundancy. Second, the impact of such a scheme on the manufacturing cost is evaluated. Third, these two parts are integrated to analyze the trade-off between yield improvement and its associated cost; the realized yield improvement is also compared to yield gain obtained when using wafer matching. The simulation results show that for higher stack sizes layer redundancy realizes a significant yield improvement as compared to wafer matching, even at lower cost. For example, for a stack size of six stacked layers and a die yield of 85 %, a relative yield improvement of 118.79 % is obtained with two redundant layers, while this is 14.03 % only with wafer matching. The additional cost due to redundancy pays off; the cost of producing a good 3D stacked memory chip reduces with 37.68 % when using layer redundancy and only with 12.48 % when using wafer matching. Moreover, the results show that the benefits of layer redundancy become extremely significant for lower die yields. Finally, layer redundancy and wafer matching are integrated to obtain further cost reductions.

Keywords 3D stacked-IC · Yield enhancement · Memory redundancy · 3D memory · Wafer matching

Responsible Editor: C. Metra

M. Taouil ( ) · S. Hamdioui
Faculty of EE, Mathematic and CS,
Delft University of Technology, Mekelweg 4,
2628 CD Delft, The Netherlands
e-mail: M.Taouil@tudelft.nl

S. Hamdioui
e-mail: S.Hamdioui@tudelft.nl

1 Introduction

The increasing demand for more functionality on ICs has been met by the semiconductor industry adhering to Moore's law. Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs), which are electrically interconnected by Through Silicon Vias (TSV). This opened up new research directions that could be investigated to continue the trend of performance increase. A TSV based 3D-SIC is an emerging technology that provides a smaller footprint, higher interconnect density between stacked dies, higher performance and lower power consumption due to shorter wires as compared to planar ICs [4]. Moreover, heterogeneous integration in 3D-SICs allows dies to be manufactured with dissimilar processing and technology nodes. For example, memory layers can be stacked on a processor [15].

The key manufacturing steps in assembling 3D-SICs are the stacking and the bonding of dies. The three existing bonding methods are Die-to-Die (D2D), Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) bonding [6]. High alignment accuracy is feasible in D2D and D2W bonding, but it impacts the throughput negatively. In D2D and D2W bonding, Known Good Die

(KGD) stacking can be applied to prevent faulty dies from being stacked [6]. W2W stacking allows for high manufacturing throughput due to single wafer alignment and thinned wafers and small die handling, but requires stacking of dies with the same area. Due to their regularity, stacked memories are very attractive to W2W stacking. However, one of the major drawbacks of W2W stacking is low compound yield especially with increased number of stacked layers.

Traditionally, memory yield improvement in 2D chips is realized by using spare rows and/or columns to repair defective ones. 3D stacked memories allow the exploration of new repair schemes that take advantage of the vertical dimension, e.g., inter-layer redundancy [8] and layer redundancy [22]. In inter-layer redundancy, if a memory layer is not repairable because the number of defective rows and/or columns is more than the spares, then additional resources (spares) from the neighboring layers could be borrowed and used. A drawback of this approach is the additional required number of TSVs and the routing complexity to mutually share and access the spare resources among the layers in the stack. The second scheme, layer redundancy, can be applied at the wafer level. Additional redundant layer(s) are stacked to replace the faulty irreparable memory dies in the stack.

This paper investigates layer redundancy as a mean for compound yield improvement for 3D W2W stacked memories. In addition, it compares the results with wafer matching [24]; a technique to improve W2W stacking by matching wafers with similar fault distributions. Finally, it combines both techniques and investigates the realized yield improvement.

The main contributions of this paper are:

- A classification of 3D memories and 3D memory redundancy repair schemes.
- An analytical model that formulates the yield gain as a result of layer redundancy.
- A memory layer replacement circuit that modifies addresses of faulty memory layer(s) to the spare layer(s).
- A comparison of 3D W2W stacked memories with and without layer redundancy in terms of the cost of producing good 3D stacks.
- A comparison between 3D W2W stacked memories using layer redundancy and wafer matching as yield improvement schemes.
- The integration of both layer redundancy and wafer matching into a single technique in order to make use of both methods.

The remainder of this paper is organized as follows. Section 2 classifies 3D memory architectures. Section 3 presents the two yield improvement schemes, i.e., wafer matching and 3D memory redundancy. Sections 4 and 5 respectively introduce models to evaluate these two schemes. The simulation results for layer redundancy are provided in Section 6. Thereafter, this scheme is compared with wafer matching in Section 7. Section 8 combines both methods to obtain further cost improvements. Finally, Section 9 concludes this paper.

2 3D Memory Architectures

This section provides a brief overview of 3D memory architectures and highlights the targeted architecture in this paper; note that the work presented here can be extended to any possible 3D memory architecture.

Partitioning memories across multiple device layers can take place at different granularities, resulting in three architectures. A top to bottom perspective is presented in the following.

1. **Stacked banks**—The coarsest granularity partitioning of memory takes place at the bank level, by stacking banks on the top of each other. Each bank consists of a complete memory system (i.e., memory cell array, address decoder, write drivers, etc.). An overall reduction in wire length is obtained (about 50 % for certain configurations), resulting into significant reduction in both power and delay [16, 18]. A 3D manufactured DRAM based on the stacking of banks manufactured by Samsung is described in [9].
2. **Cell arrays stacked on logic**—This approach, in contrast to the previous one, separates the peripheral logic (row decoders, sense amplifier, column select logic, etc), from the cell arrays. The peripheral logic is placed on the bottom layer while the cell array is split across one or multiple layers. This is considered to be the *true 3D memory* [16]. Research in this area has been performed for both SRAMs [16, 25] and DRAMs [2, 13]. By using this separation method, the peripheral logic can be optimized independently for speed, while the cell arrays can be arranged to meet different criteria (density, footprint, thermal, etc). Examples of 3D manufactured DRAM based on *cell arrays stacked on logic* are manufactured by NEC Electronics, Elpida Memory [10] and Tezzaron [29]. A classification within the array layer can also be made.
   - **Divided-columns**: in which bitlines are divided and mapped onto different layers;

– **Divided-rows**: in which wordlines are divided and mapped onto different layers, requiring one die-to-die TSV per wordline.

Both organizations reduce latency and power due to reduced wordline/bitline lengths.

3. **Intra-cell (bit) partitioning**—Here, memory cells are split among one or more layers. At this fine granularity level, the relative small size of the cell and the size of the TSV make the splitting across layers a difficult task [25]. Nevertheless, the authors in [16] claim that this option can be feasible for multi-port SRAM arrays, such as register files, when the access transistors of the cell are split among multiple layers.

An example of an architecture that could benefit from redundancy is the memory architecture considered in [13]. This architecture, *cell arrays stacked on logic*, makes heterogeneous integration feasible. For example, memory layers manufactured in DRAM process technology optimized for area can be stacked on the peripheral circuits manufactured in a logic process optimized for speed.

3 Yield Improvement Schemes

This section describes two types of yield improvement schemes. Section 3.1 describes wafer matching, a general technique to increase the W2W compound yield. Subsequently, Section 3.2 presents a classification of possible memory redundancy schemes and discusses the method analyzed in this paper.

3.1 W2W Matching

As already mentioned, W2W stacking suffers from a low compound yield. Wafer matching has been researched to mitigate this drawback by many authors [17, 20, 21, 24, 27]; it is a technique based on the matching of wafers with similar fault maps. In case of a large stack size or low die yield, the improvement can be significant. The improvement decreases for higher die yield. For example, for a stack size of two layers with a die yield of 85 % and 1,278 dies per wafer, wafer matching is able to increase the compound yield from 72.3 % (for random W2W stacking) to 73.1 % [24].

Wafer matching may not be applicable for *cell arrays stacked on logic* architecture as it requires wafer tests prior to stacking. Depending on the memory architecture and implementation, performing pre-bond wafer tests may not always be possible, due to the absence of peripheral circuits.

3.2 3D Memory Redundancy

To increase the memory yield, a memory repair scheme can be added to any of the memory architectures presented in Section 2. Traditionally, yield improvement for 2D memories is based on the use of spare rows and/or columns [1]. 3D stacked memories, however, provide additional repair features due to the vertical dimension. The redundancy schemes for 3D memories can be classified into three groups.

1. Intra-layer redundancy: Redundancy within each layer is similar to that in planar memories. Each layer may have spare rows and/or columns that can be used within the same layer to improve the yield.
2. Inter-layer redundancy: In inter-layer redundant memories, spare rows and/or columns cannot be accessed only from the die they belong to, but also from neighbor dies. Hence, they can borrow additional resources in case they run out of their own. Tezzaron memories are examples of memory architectures that use inter-layer redundancy [15]. In [8], inter-layer redundancy is used by the authors to increase the stacked memory yield for different allocation algorithms.
3. Layer redundancy: Redundancy at the wafer or die level. A faulty irreparable memory layer is disabled and instead is replaced with a complete redundant layer. A memory layer is not repairable if the required number of spares exceed the existing spares within it.

In this paper, we analyze the yield increase based on layer redundancy.

4 Layer Redundancy for Yield Improvement

This section covers the modeling of yield and cost for layer redundancy; it also presents a simple design for memory repair. Section 4.1 discusses the assumptions made for layer redundancy. Thereafter, the yield and cost modeling are described in Sections 4.2 and 4.3 respectively. Finally, Section 4.4 presents an example of a memory-repair scheme.

4.1 Definitions and Assumptions

In order to accurately evaluate the memory yield improvement due to layer redundancy, different process parameters have to be appropriately chosen. A 3D stacked memory consists of multiple stacked layers/dies interconnected by TSVs. Each die in the stack can be either faulty or non-faulty (i.e., functional). The yield of

the die is modeled by Y_D. In addition, new defects may be introduced during the stacking process and have to be taken into consideration [14]. Dies/layers that enter the stack could get corrupted e.g., due to bonding and thinning. The new introduced faults due to stacking are modeled by the stacked-die yield Y_{SD}. For the TSVs, the interconnect yield is represented by Y_{INT}. Other parameters that influence the compound yield are the stack size n and the number of redundant layers r. The complete stack *size* is denoted by $s = n + r$.

The following assumptions are made in this paper with respect to layer redundancy analysis:

- The memory layers in the stack are considered to be independent; each layer can be either faulty or non-faulty.
- Since many TSVs are shared (e.g., for address or data buses), it is assumed that any malfunction in communication between two layers results in faulty stacked memory.
- We do not consider the peripheral circuit layer in the model to-be-presented as it impacts both 3D stacked memories with or without layer redundancy in a similar way.

To calculate the cost per 3D-SIC, we need to include the manufacturing, test and packaging costs. The manufacturing cost depends on the stack size, wafer cost and 3D stacking cost. The test cost is a function of the number of dies per wafer d, and the cost to test the interconnects and dies. The complete test cost for a stack size of n layers equals $C_t = (n-1) \cdot d \cdot t_{\mathrm{int}} + n \cdot d \cdot t_{\mathrm{die}}$. Here, t_{int} is the interconnect cost and t_{die} the test cost per die. We denote the packaging cost to be $C_{\mathrm{packaging}}$ for a single 3D-SIC. The number of dies per wafer can be derived from the wafer size and die area A.

4.2 Yield Modeling

The model will be presented first for 3D stacked memories without layer redundancy and thereafter for those with layer redundancy.

Memories Without Layer Redundancy In case there is no redundancy, i.e. $s = n$ and $r = 0$, each layer in the stack must operate to ensure memory functionality. The compound yield $Y(n)$ can be described as a function of the die yield Y_{D} and stack size n. Besides the dies, also the interconnects and the 3D bonding must be fault free; hence, the stacked-die yield Y_{SD} and the interconnect yield Y_{INT} have to be considered as well. This leads to the following yield expression for non-redundant memories.

$$Y(n) = Y_{\mathrm{D}}^{n} \cdot Y_{\mathrm{SD}}^{n-1} \cdot Y_{\mathrm{INT}}^{n-1} \tag{1}$$

Note that 3D stacked memory with n layers requires $n-1$ stacking steps. For the interconnect yield Y_{INT}, the yield after repair is assumed, if TSV redundancy is provided.

Memories with Layer Redundancy In this case, r redundant layers are appended to the stack with n layers resulting in a total layers of $s = n + r$. If n or more layers out of the stacked s layers are functionally correct, then the final 3D-SIC is assumed to be non-faulty. The probability $p(i)$ that i layers out of s layers are non-faulty can be formulated by the binomial expression:

$$p(i) = \binom{s}{i} \cdot Y_{\mathrm{D}}^{i} \cdot (1 - Y_{\mathrm{D}})^{s-i} \tag{2}$$

We extend the symbol $Y(n)$ for non-redundant memories to $Y_{\mathrm{LR}}(n, s)$ to denote the yield of a stack containing s layers with $r = s - n$ redundant layers. The yield for layer redundant enabled memories can be expressed now by:

$$\begin{aligned} Y_{\mathrm{LR}}(n,s) &= \left(\sum_{i=n}^{s} p(i) \right) \cdot Y_{\mathrm{SD}}^{s-1} \cdot Y_{\mathrm{INT}}^{s-1} \\ &= \left(\sum_{i=n}^{s} \binom{s}{i} \cdot Y_{\mathrm{D}}^{i} \cdot (1 - Y_{\mathrm{D}})^{s-i} \right) \\ &\quad \cdot Y_{\mathrm{SD}}^{s-1} \cdot Y_{\mathrm{INT}}^{s-1} \end{aligned} \tag{3}$$

In order for the stack to be considered defect-free, at least n out of s layers must be defect-free. Note that the redundant layers can be faulty as well. Equations 1 and 3 are equivalent in case $n = s$, i.e., in case there is no layer redundancy.

4.3 Cost Modeling

The question rises whether it is cost-wise justified to increase the yield by adding more redundant layers. To answer this question, the cost for layer redundancy, C_{LR}, will be calculated for later evaluation. In this section, we present the cost C_{LR} for layer redundancy. The cost $C_{\mathrm{LR}}(s)$ for a stack size s can be formulated by Eq. 4.

$$C_{\mathrm{LR}}(s) = C_{\mathrm{LR,m}}(s) + C_{\mathrm{LR,t}}(s) + C_{\mathrm{LR,p}}(s) \tag{4}$$

$$C_{\mathrm{LR,m}}(s) = s \cdot C_{\mathrm{w}} + (s-1) \cdot C_{\mathrm{3D}} \tag{5}$$

$$C_{\mathrm{LR,t}}(s) = C_{\mathrm{LR,t,post}}(s) + C_{\mathrm{LR,t,final}}(s) \tag{6}$$

$$C_{\mathrm{LR,t,post}}(s) = (s-1) \cdot d \cdot t_{\mathrm{int}} + Y_{\mathrm{INT}}^{s-1} \cdot s \cdot d \cdot t_{\mathrm{die}} \tag{7}$$

$$\begin{aligned} C_{\mathrm{LR,t,final}}(s) &= Y_{\mathrm{LR}}(n,s) \\ &\quad \cdot \{(s-1) \cdot d \cdot t_{\mathrm{int}} + s \cdot d \cdot t_{\mathrm{die}}\} \end{aligned} \tag{8}$$

$$C_{\mathrm{LR,p}}(s) = Y_{\mathrm{LR}}(n,s) \cdot d \cdot C_{\mathrm{package}} \tag{9}$$

In this equation, $C_{\mathrm{LR,m}}(s)$ presents the manufacturing cost, $C_{\mathrm{LR,t}}(s)$ the test cost and $C_{\mathrm{LR,p}}(s)$ the packaging cost. In Eq. 5, which presents the manufacturing cost, C_{w} presents the wafer cost and C_{3D} the cost related to 3D stacking processes including TSV, back side processing, bonding processing, etc. Note that s wafers are needed and that the stacking process operation has to be performed $s-1$ times.

Testing 3D-SICs are can be performed at several stages, pre-bond testing (prior stacking), mid-bond testing (during stacking), post-bond testing (prior packaging) and a final testing (post-packaging) [14]. For layer reduncy, we ignore the pre-bond and mid-bond tests $\mathrm{T_{mi}}$ as dies are stacked based on the wafer level. Intermediate mid-bond tests cannot prevent faulty dies to be stacked as the case is for D2W stacking. Therefore, the test cost $C_{\mathrm{LR,t}}(s)$ in Eq. 6 is composed out of two phases, a post-bond test prior to packaging (Eq. 7) and a final test after packaging (Eq. 8). In each testing phase, we assume that interconnects are tested first, similarly as in [23]. As some of the faulty interconnects are detected, some die tests for 3D-SICs can be skipped. For example, in Eq. 7 after defective interconnects are identified, only dies of the 3D-SICs with fault-free interconnects should be further tested. This remaining fraction equals $1-Y_{INT}^{s-1}$. The total test cost depends on the number of dies d on the wafer, the test cost for a single interconnect t_{int} and the test cost per die t_{die}.

The total packaging cost (Eq. 9) equals the number of packaged ICs times the packaging cost $C_{\mathrm{packaging}}$ per 3D-SIC. Note that we assume a packaging yield of 100 %.

Obviously, for 3D stacked memories without layer redundancy, the cost $C(n)$ can be derived similarly and is described by the following equations.

$$C(n) = C_{\mathrm{m}}(n) + C_t(n) + C_p(n) \tag{10}$$

$$C_{\mathrm{m}}(n) = n \cdot C_{\mathrm{w}} + (n-1) \cdot C_{\mathrm{3D}} \tag{11}$$

$$C_{\mathrm{t}}(n) = C_{\mathrm{t,post}}(n) + C_{\mathrm{t,final}}(n) \tag{12}$$

$$C_{\mathrm{t,post}}(n) = (n-1) \cdot d \cdot t_{\mathrm{int}} + Y_{\mathrm{INT}}^{n-1} \cdot n \cdot d \cdot t_{\mathrm{die}} \tag{13}$$

$$C_{\mathrm{t,final}}(n) = Y(n) \cdot \{(n-1) \cdot d \cdot t_{\mathrm{int}} + n \cdot d \cdot t_{\mathrm{die}}\} \tag{14}$$

$$C_{\mathrm{p}}(n) = Y(n) \cdot d \cdot C_{\mathrm{packaging}} \tag{15}$$

4.4 Design for Memory Repair

In the previous sections, yield and cost formulation for layer redundancy were presented. In this section, we will briefly discuss the different existing techniques to realize layer redundancy and thereafter we propose a layer replacement scheme for 3D stacked memories.

4.4.1 Traditional Approaches

Redundancy for 2D memories (intra-layer) is typically performed by replacing the faulty row/column with spares. The address of the faulty row/column is stored in a programmable non-volatile memory before shipping the chip to retain the information during the power off. When the memory is accessed, it checks if the addressed location is faulty by comparing it to the stored faulty addresses in the programmable devices. In case faulty, the initial (faulty) location is prevented from being accessed and the spare location is activated instead.

The programmable devices can include fuses, anti-fuses or nonvolatile memory cells. Fuses may include material such as polysilicon, silicides or metals such as copper; they can be blown (programmed) by either laser or electric current. Obviously laser fusing cannot work for intra-layer redundancy if the memory cells are stacked on the top of the peripheral logic layer. Once stacked, blowing fuses by laser might become unfeasible, as they are not reachable by the laser beam. On the other hand, electrical fusing can be applied for layer redundancy [12]. Faulty addresses can be programmed even after packaging. However, it requires an on-chip programming circuit [19]. Similarly to fuses, anti-fuses can be programmed (e.g., breaking down a dielectric) electrically [28] or by using laser pulses. Last, non-volatile memory cells can be also used to store the faulty addresses, especially for non-volatile memories such as EEPROM.

4.4.2 Layer Replacement Scheme for 3D Stacked Memories

Memory repair based on layer redundancy needs to store the ID (index) of the faulty layer in a programmable non-volatile device. As already mentioned, this can be done with electrical fusing, electrical anti-fusing or by using nonvolatile memory cells. If the faulty layer is accessed, the repair scheme should redirect the address to a redundant layer. In the rest of this section, we will show a concept that can realize such a task.

Let us assume that the size of the s memory layers are the same; hence, $\log_2(s)$ bits can be used to distinguish between the different layers. We assume further that the $\log_2(s)$ bits are the most significant bits (MSB's) of the memory address; therefore, they are unique for each layer. Figure 1 shows how this MSB's can be used to redirect the address to a redundant layer rather than

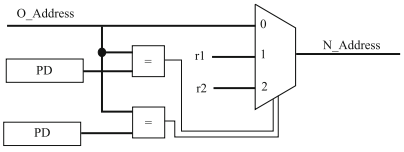


Fig. 1 Layer replacement circuit

the faulty layer. The programmable devices PD in the figure store the ID (i.e., the MSB's) of the faulty layers. The MSB's of the original address O_Address will be compared with the stored bits in the PD's; if a hit occurs, then the O_Address has to be mapped to the MSB's of a redundant layer (denoted by r1 and r2). For example, assume a stack size $n = 2$ and the number of redundant layers is $r = 2$ as in the figure, then 2 bits ($= \log_2(4)$) needed as MSB's to identify the four layers. Assume further that the combinations 00 and 01 identify the layers L1 and L2 and the combination 10 and 11 identify the spare redundant layers R1 and R2. If L1 is faulty, then its access will be inhibited as the comparator will produce a hit and force the mux to select the new address r1; in this case the address is converted from 00 to 10, hence accessing the redundant layer instead of the faulty layer. Similarly, if L2 is faulty, its address will be remapped to the address $r2 = 11$.

5 Wafer Matching for Yield Improvement

This section briefly presents wafer matching as it is used for comparison with layer redundancy. First, the process assumptions and definition for wafer matching are presented. Thereafter, a yield and a cost model are described.

5.1 Definitions and Assumptions

To fairly compare layer redundancy with wafer matching, the same yield parameters used in layer redundancy are used here, i.e., die yield Y_D, interconnect yield Y_{INT} and stacked-die yield Y_{SD} have to be used. However, due to the nature of wafer matching an additional parameters must be considered, the repository size.

A repository contains a collection of wafers with the same functionality. The larger the size of the repository the better the quality of the matching, since there are more wafers to select from. The symbol k is used to denote the repository size.

The yield improvement of wafer matching heavily depends on the number of dies per wafer d. As wafer matching requires pre-bond testing, each die has to be tested prior entering the stack. We use the same symbols t_{int} and t_{die} denote the cost per interconnect and die.

5.2 Yield and Cost Models for Wafer Matching

Improve yield for 3D circuits based on wafer matching has been discussed by many authors [17, 20, 21, 24, 27]. In this paper, we use the adaptive Best Pair (BP) algorithm [24] to determine the yield increase due to wafer matching.

The BP matching scenario realizes a yield $Y_{BP} = f(n, k, d, Y_D)$, which is a function of the stack size n, the repository size k, the number of dies per wafer d and the die yield Y_D. By assuming k and d to be constant, we can define $Y_{BP}(n, Y_D)$; i.e., it is primarily a function of the stack size and the die yield. This yield can be recursively described by the following equation:

$$Y_{BP}(n, Y_D) = Y_{BP}(n-1, Y_D) \cdot \text{Match}(n-1, Y_D). \quad (16)$$

Here $Match(n-1, Y_D)$ presents the die yield of the best wafer that matches with the stacked $n-1$ layers (given a certain matching criterion).

To calculate the compound yield due to wafer matching, Y_{WM}, both stacked-die yield Y_{SD} and interconnect yield Y_{INT} have to be incorporated with Y_{BP}. We define the wafer matching yield as follows:

$$Y_{WM}(n) = Y_{BP}(n, Y_D) \cdot Y_{SD}^{n-1} \cdot Y_{INT}^{n-1} \quad (17)$$

The cost to perform the wafer matching consist also of three components: manufacturing, test and packaging cost. Equation 18 describes this cost.

$$C_{WM}(n) = C_{WM,m}(n) + C_{WM,t}(n) + C_{WM,p}(n) \quad (18)$$

$$C_{WM,m}(n) = n \cdot C_w + (n-1) \cdot C_{3D} \quad (19)$$

$$C_{WM,t}(n) = C_{WM,t,pre}(n) + C_{WM,t,post}(n) + C_{WM,t,final}(n) \quad (20)$$

$$C_{WM,t,pre}(n) = n \cdot d \cdot t_{int} \quad (21)$$

$$C_{WM,t,post}(n) = Y_{BP} \cdot (n-1) \cdot d \cdot t_{int} \quad (22)$$

$$C_{WM,t,final}(n) = Y_{BP} \cdot Y_{INT}^{n-1} \cdot \{(n-1) \cdot d \cdot t_{int} + n \cdot d \cdot t_{die}\} \quad (23)$$

$$C_{WM,p}(n) = Y_{BP} \cdot Y_{INT}^{n-1} \cdot d \cdot C_{packaging} \quad (24)$$

The manufacturing cost is assumed to be the same as for the case no wafer matching is used. The test cost, however, differs as a pre-bond test is required ($C_{WM,t,pre}(n)$). In the pre-bond test only dies are tested. In the post-bond test, die test are skipped as it is proven to be more cost-effective [27]. Here, only the

J Electron Test (2012) 28:523–534

interconnects are tested during the post-bond test. As a consequence of this, some faulty stacked dies will escape the test and therefore will be packaged. These faulty chips, however, will be detected in the final test.

In case wafer matching is not performed, the yield and cost are given by Eqs. 1 and 10 respectively.

6 Simulation Results for Layer Redundancy

In this section we analyze the yield gain due to layer redundancy and its associated cost by attributing the manufacturing cost to the good stacked ICs. However, first the process parameters used for simulation will be given.

6.1 Process Parameters

The defined parameters in Section 4.1 need to have actual values for the simulation. In this section, we justify their values. We assume a die yield of $Y_D = 85\ \%$ as reported in the ITRS roadmap [7]. The stacked-die yield Y_{SD} is assumed to be 99 % [27]. The interconnect yield Y_{INT} is assumed to be 97 % per stacked layer [27].

In order to determine the number of dies per wafer d, we need to know the wafer size and die area. A standard 300 mm diameter wafer is selected with an edge clearance of 3 mm. The memory die area selected belonging to the considered die yield is assumed to be $A = 93\ \text{mm}^2$ [7]. For this die area and wafer size, the number of Gross Dies per Wafer (GDW) approximately equals to $d = 675$ [5].

For the test cost, we assume a test cost per die $t_{die} = 0.23$ cent [3, 11]. We assume that the interconnect test are 100 less in cost, similar as in [27].

The packaging cost forms a significant fraction of the overall cost and depends on the used technique [26]. In this paper, we assume the packaging cost to be 50 % of a die cost.

6.2 Yield Improvement

The relative yield improvement of memories enabled with redundancy over memories without layer redundancy can be expressed by normalizing Eq. 3 over Eq. 1. The following equation describes the obtained result:

$$\begin{aligned}\frac{Y_{LR}(n,s)}{Y(n)} &= \frac{\left(\sum_{i=n}^{s} p(i)\right)}{Y_D^n} \cdot Y_{SD}^{s-n} \cdot Y_{INT}^{s-n} \\ &= \left(\sum_{i=n}^{s} \binom{s}{i} \cdot Y_D^{i-n} \cdot (1-Y_D)^{s-i}\right) \\ &\quad \cdot Y_{SD}^{s-n} \cdot Y_{INT}^{s-n} \end{aligned} \tag{25}$$

Table 1 shows the yields for memories with and without layer redundancy. The second row gives the absolute yield (Abs. yield) of the stack without using layer redundancy. The rest of the table gives the yield improvement as a consequence of layer redundancy for different stack sizes n and different number of redundant layers r. For cost reasons it is assumed that $r \leq n$; i.e., the number of redundant layers is considered smaller than or equal to the stack size n. Each entry in the table (except the Abs. yield row) lists the relative yield improvement $\frac{Y_{LR}(n,s)}{Y(n)}$ (Eq. 25) in percentage for each value of n and r; entities where $r > n$ are indicated as 'n.a.' (not applicable). Inspecting the table reveals the following:

- Layer redundancy improves the memory yield irrespective of the considered stack size and number of redundant layers. The yield improvement becomes significant as the stack size increases; this is because the occurrence probability of faulty layers increases.
- Adding more redundant layers does not always result in better yield improvement. The minimum number of redundant layers that have to be added to achieve the maximal yield improvement depends in addition to n also on the process parameters under consideration such as Y_D, Y_{SD} and Y_{INT}. For example, the yield improvement for $n = 4$ realized with $r = 2$ is larger than that realized with $r = 4$. This yield drop is a consequence of additional faults introduced in the larger stack due to the extra 3D processing steps.

6.3 Cost Evaluation

To evaluate the additional yield gain of a redundant memory fairly, its increased manufacturing cost must be compensated for. In order to do that, we define the cost of a *good die* C^G as the cost of manufacturing a good stacked IC; i.e., normalizing the cost $C(n)$ to the yield. This cost for 3D stacked memory without

Table 1 Relative yield improvement using layer redundancy in % for various n and r

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Abs. yield	85.00	69.38	56.63	46.23	37.73	30.80
$r=1$	**10.43**	24.84	39.24	53.65	68.05	82.46
$r=2$	n.a.	**26.11**	**46.16**	**68.30**	92.50	118.79
$r=3$	n.a.	n.a.	43.35	67.59	**95.32**	**126.84**
$r=4$	n.a.	n.a.	n.a.	62.45	90.58	123.26

Bold entries show the optimal values

and with layer redundancy are given in Eqs. 26 and 27 respectively.

$$C^G(n) = C(n)/Y(n) \tag{26}$$

$$C^G_{\mathrm{LR}}(n,s) = C_{\mathrm{LR}}(s)/Y_{\mathrm{LR}}(n,s) \tag{27}$$

By using these equations, the relative improvement or depreciation of the price of a good 3D-SIC with layer redundancy over one without layer redundancy can be expressed as:

$$\frac{C^G_{\mathrm{LR}}(n,s)}{C^G(n)} = \frac{C_{\mathrm{LR}}(s)}{C(n)} \cdot \frac{Y(n)}{Y_{\mathrm{LR}}(n,s)} \tag{28}$$

Here, Eqs. 4 and 10 give the expressions for $C_{\mathrm{LR}}(s)$ and $C(n)$. The last part of the equation, $\frac{Y(n)}{Y_{\mathrm{LR}}(n,s)}$, can be evaluated by using Eq. 25.

Figure 2 shows the above cost ratio for various values of n and s, and for $0.1 \leq \frac{C_{3D}}{C_w} \leq 0.9$, i.e., the 3D processing cost lies between 10 and 90 % of the wafer cost. The following can be concluded from the figure:

- The impact of the ratio $\frac{C_{3D}}{C_w}$ on the cost ratio $\frac{C^G_{\mathrm{LR}}(n,s)}{C^G(n)}$ is negligible, especially for $n > 3$.
- Except for $n = 3$ and $s = 5$, the realized yield improvement is high enough to pay off the additional cost made (related to additional memory layers and stacking process). Again, this conclusion applies for our case study and the assumed process parameters. Other process parameters may result in other conclusions. Nevertheless, the figure clearly shows that generally speaking, the achieved yield improvement using layer redundancy results in lower cost per good stack.

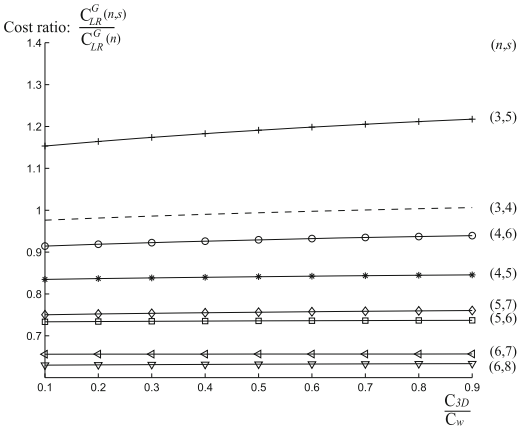


Fig. 2 Impact of layer redundancy on the cost ratio $\frac{C^G_{\mathrm{LR}}(n,s)}{C^G(n)}$

Table 2 Relative cost improvement using layer redundancy for various n and r

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
$r=1$	170.18	20.21	**−2.73**	**−17.00**	**−27.10**	−34.79
$r=2$	n.a.	54.37	14.30	−9.55	−25.79	**−37.68**
$r=3$	n.a.	n.a.	37.28	4.63	−17.13	−32.75
$r=4$	n.a.	n.a.	n.a.	21.66	−5.48	−24.73

Bold entries show the optimal values

- The larger n, the larger the impact of layer redundancy; i.e., the better the cost improvement due to layer redundancy. For example, for $n = 3$ and $s = 4$, the cost reduction achieved is around 2.73 %, while this is 27.10 % for $n = 5$ and $s = 6$.

Next, the impact of different values of n and r on the cost ratio $\frac{C^G_{\mathrm{LR}}(n,s)}{C^G(n)}$ will be analyzed. The results are summarized in Table 2; it is assumed that $\frac{C_{3D}}{C_w} = 0.3$. The table shows that for $n = r = 1$, the cost of producing a good stacked IC using layer redundancy is more than twice expensive. This can be explained by the fact that adding a single redundant layer to $n = 1$ doubles the wafer cost. The associated cost with layer redundancy starts to pay off from $n = 3$ on. As the table shows, additional redundant layers do not always result in *lower* cost. It strongly depends on the stack size and the number of the-to-be added redundant layers (as well as on the process parameters). Nevertheless, the larger n, the more benefits can be realized. For example, for $n = 6$ a cost reduction of 37.68 % can be obtained.

Another aspect which is worth to examine is the impact of the die yield Y_{D} and the stacking yield parameters Y_{INT} and Y_{SD} on the cost. We still assume the case where $n = 5$ and $s = 6$. The cost ratio $\frac{C^G_{\mathrm{LR}}(5,6)}{C^G(5)}$ for different values of stack yield Y_{int}, Y_{SD} and die yield Y_{D} is given in Table 3; Y_{int} and Y_{SD} are considered between 91 and 99 % and Y_{D} is considered to between 60 and 90 %,. The table reveals that the die yield has the

Table 3 Relative cost improvement using layer redundancy for various Y_{D}, Y_{int} and Y_{SD}

Y_{INT}	Y_{SD}	$\frac{C^G_{\mathrm{LR}}(5,6) - C^G(5)}{C^G(5)}$			
		$Y_{\mathrm{D}}=0.6$	$Y_{\mathrm{D}}=0.7$	$Y_{\mathrm{D}}=0.8$	$Y_{\mathrm{D}}=0.9$
0.91	0.91	−51.50	−41.64	−26.99	−3.17
0.91	0.95	−53.47	−43.96	−29.87	−7.00
0.91	0.99	−55.27	−46.06	−32.43	−10.46
0.95	0.91	−53.48	−43.97	−29.87	−7.02
0.95	0.95	−55.35	−46.16	−32.55	−10.63
0.95	0.99	−57.06	−48.14	−34.97	−13.86
0.99	0.91	−55.28	−46.08	−32.44	−10.48
0.99	0.95	−57.06	−48.15	−34.97	−13.86
0.99	0.99	−58.67	−50.01	−37.22	−16.87

J Electron Test (2012) 28:523–534

highest impact on the cost ratio; the lower the die yield, the higher the benefits obtained by layer redundancy. For example, for a $Y_D = 60$ % a cost improvement around 55 % is obtained, while this does not exceed 16.87 % for $Y_D = 90$ %. Moreover, the table shows that the higher the stack yield, the higher the benefit of layer redundancy.

7 Comparison with Wafer Matching

This section gives first the simulation results for wafer matching; these are thereafter compared with those obtained for layer redundancy.

7.1 Simulation Results for Wafer Matching

In this section, we derive the equations to evaluate the cost for wafer matching and simulate them. Again, we consider the yield and cost improvements with respect to the case where wafer matching is not used.

The defined parameters in Section 5.1 need to have actual values for the simulation. We use exactly the same parameters as defined in Section 6.1. The repository size for the wafer repositories is assumed to be $k = 50$.

7.1.1 Yield Improvement

The relative yield improvement of memories enabled with wafer matching over memories without wafer matching can be expressed by normalizing Eq. 17 over Eq. 1. The following expression describes the obtained result:

$$\frac{Y_{WM}(n)}{Y(n)} = \frac{Y_{BP} \cdot Y_{SD}^{n-1} \cdot Y_{INT}^{n-1}}{Y_D^n \cdot Y_{SD}^{n-1} \cdot Y_{INT}^{n-1}} = \frac{Y_{BP}}{Y_D^n} \tag{29}$$

This yield is exactly reported by the tool that implements the Best Pair (BP) matching scenario [24]. Table 4 shows the absolute yield (second row) and the relative yield improvement (third row) for different stack sizes n.

Wafer matching is only applicable for a stack of two or more layers. The larger the stack size, the higher the yield gain. This relative yield improvement increases from 1.62 % up to 14.03 % for stack sizes of two and six layers respectively.

7.1.2 Cost Evaluation

To evaluate the additional yield gain of a redundant memory fairly, its manufacturing and additional test cost must be compensated for. In order to do that, we define the cost of a *good die* C_{WM}^G as the cost of a good stacked IC using wafer matching, similarly as in Eqs. 26 and 27.

$$C_{WM}^G(n) = \frac{C_{WM}(n)}{Y_{WM}(n)} \tag{30}$$

Using this equation and Eq. 26, the relative improvement or depreciation of the price of a good 3D-SIC with wafer matching over one without it can be expressed as:

$$\frac{C_{WM}^G(n)}{C^G(n)} = \frac{C_{WM}(n)}{C(n)} \cdot \frac{Y(n)}{Y_{WM}(n)} \tag{31}$$

Here, Eqs. 18 and 10 give the expressions for $C_{WM}(n)$ and $C(n)$ respectively. The last part of the equation, $\frac{Y(n)}{Y_{WM}(n)}$, can be evaluated by using Eq. 29.

The results of this equation are depicted in the last row of Table 4. It shows that wafer matching becomes more lucrative for increased stack sizes. For a stack size of 2, the improvement is only 2.56 %; it grows to 12.48 % for a stack size of six layers.

7.2 Comparison

Sections 6 and 7.1 describe the yield improvement schemes layer redundancy and wafer matching respectively. In this section, we summarize both methods and compare the cost improvements between them. Table 5 shows this comparison. The first column contains the stack size. The second and third columns contain the yield improvements for both techniques and the fourth column gives the number of redundant layers used to achieve the yield improvement in the third column. The fifth and sixth column show the cost improvements of both schemes, while the last column shows the number of redundant layers used to obtain the cost improvement in the sixth column. It should be noted that depending on n, an optimal number of redundant layers r (realizing maximal yield or cost improvement)

Table 4 Relative yield and cost improvements for various n

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Abs. yield	85.00	69.38	56.63	46.23	37.73	30.80
$\frac{Y_{WM}(n) - Y(n)}{Y(n)}$ (%)	–	1.62	3.71	6.49	10.00	14.03
$\frac{C_{WM}^G(n) - C^G(n)}{C^G(n)}$ (%)	–	−2.56	−4.50	−6.79	−9.47	−12.48

Table 5 Yield and cost comparison between wafer matching and layer redundancy

n	$\frac{Y_{\mathrm{WM}}(n)-Y(n)}{Y(n)}$ (%)	$\frac{Y_{\mathrm{LR}}(n,s)-Y(n)}{Y(n)}$ (%)	r	$\frac{C^G_{\mathrm{WM}}(n)-C^G(n)}{C^G(n)}$ (%)	$\frac{C^G_{\mathrm{LR}}(n,s)-C^G(n)}{C^G(n)}$ (%)	r
2	1.62	26.11	2	−2.56	20.21	1
3	3.71	46.16	2	−4.50	−2.73	1
4	6.49	68.30	2	−6.79	−17.00	1
5	10.00	95.58	3	−9.47	−27.10	1
6	14.03	126.84	3	−12.48	−37.68	2

is selected for the comparison. Considering yield only, layer redundancy outperforms wafer matching by an order of magnitude. Even the cost picture of these two schemes confirms the superiority of layer redundancy except for $n = 2$; the larger n, the larger the benefit. For example, for a six stacked IC wafer matching is able to reduce the cost with 12.48 % as compared to random stacking, while layer redundancy is able to reduce this with 37.68 %. However, for $n = 2$ layer redundancy will result in an additional cost of 20.21 %.

8 Combining Layer Redundancy and Wafer matching

In this section, we combine the two methods. In order to achieve that, a new algorithm is developed. This algorithm is described in Section 8.1. Thereafter, we present the results and analyze the additional cost improvements in Section 8.2. Finally, we compare the two stand-alone techniques with their combined version in Section 8.3.

8.1 Algorithm

To combine layer redundancy and wafer matching, a two-step algorithm is used. The first step performs the matching of the first n layers; the BP matching scenario is used with slight modifications such as keeping track of the number of good dies per stack. The second step consists of matching the r redundant layers to the stacked n layers. Two different methods can be used for this step:

- *Match The Best*: To maximize the compound yield, each matching step targets stacks with $n-1$ good dies. The stacks with $n-1$ good dies directly contribute to the yield if a good die is stacked on them. Note that after matching, stacks that had $n-2$ good dies will have $n-1$ good dies in the next step.
- *Match The Worst*: To maximize the compound yield, each matching step targets stacks with the most faulty dies that are still repairable. Thus, the first matching step is based on stacks with $n-r$ good dies, thereafter, stacks with $n-r-1$ good dies, etc. The process stops when all r redundant layers are matched.

In the coming sections, we only consider the *Match The Best* method as both methods report similar results. We denote the yield after matching as $Y_{\mathrm{M,BP}}$ for this method.

8.2 Simulation Results

Similarly as for the disjoint yield improvements methods, both the yield and cost components are going to be explored. We define the cost $C_{\mathrm{COM}}(s)$ of a 3D-SIC using the combined approach in a similar way as we did for wafer matching, but now with stack size s. The following equations describe these cost.

$$C_{\mathrm{COM}}(s) = C_{\mathrm{COM,m}}(s) + C_{\mathrm{COM,t}}(s) + C_{\mathrm{COM,p}}(s) \quad (32)$$

$$C_{\mathrm{COM,m}}(s) = s \cdot C_{\mathrm{w}} + (s-1) \cdot C_{\mathrm{3D}} \quad (33)$$

$$C_{\mathrm{COM,t}}(s) = C_{\mathrm{COM,t,pre}}(s) + C_{\mathrm{COM,t,post}}(s) + C_{\mathrm{COM,t,final}}(s) \quad (34)$$

$$C_{\mathrm{COM,t,pre}}(s) = s \cdot d \cdot t_{\mathrm{int}} \quad (35)$$

Table 6 Relative cost improvement using the combined method for various n and r

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Abs. yield	85.00	69.38	56.63	46.23	37.73	30.80
$r=1$	**17.11**	29.70	43.02	57.06	71.51	86.40
$r=2$	n.a.	**36.94**	57.49	80.37	105.12	131.91
$r=3$	n.a.	n.a.	**61.57**	88.42	118.86	153.16
$r=4$	n.a.	n.a.	n.a.	**90.77**	**135.51**	**161.32**

Bold entries show the optimal values

Table 7 Relative cost improvement using the combined method for various n and r

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
$r=1$	58.47	12.91	**−7.80**	**−20.86**	−30.16	−37.37
$r=2$	n.a.	39.11	3.30	−17.79	**−32.06**	**−42.51**
$r=3$	n.a.	n.a.	18.81	−9.19	−27.76	−41.05
$r=4$	n.a.	n.a.	n.a.	1.31	−21.10	−36.96

Bold entries show the optimal values

Table 8 Cost reduction: combined wafer matching and layer redundancy

n	r	$\frac{C^G_{\mathrm{COM}}(n,s) - C^G(n)}{C^G(n)}$ (%)	$\frac{C^G_{\mathrm{COM}}(n,s) - G^G_{\mathrm{WM}}(n)}{G^G_{\mathrm{WM}}(n)}$ (%)	$\frac{C^G_{\mathrm{COM}}(n,s) - C^G_{\mathrm{LR}}(n,s)}{C^G_{\mathrm{LR}}(n,s)}$ (%)
2	1	12.91	15.88	−9.94
3	1	−7.80	−3.46	−5.68
4	1	−20.86	−15.09	−3.89
5	2	−32.06	−24.95	−5.62
6	2	−42.51	−34.31	−6.60

$$C_{\mathrm{COM,t,post}}(s) = Y_{\mathrm{M,BP}} \cdot (s-1) \cdot d \cdot t_{\mathrm{int}} \tag{36}$$

$$C_{\mathrm{COM,t,final}}(s) = Y_{\mathrm{M,BP}} \cdot Y_{\mathrm{INT}}^{s-1} \cdot \{(s-1) \cdot d \cdot t_{\mathrm{int}} + s \cdot d \cdot t_{\mathrm{die}}\} \tag{37}$$

$$C_{\mathrm{COM,p}}(n) = Y_{\mathrm{M,BP}} \cdot Y_{\mathrm{INT}}^{s-1} \cdot d \cdot C_{\mathrm{packaging}} \tag{38}$$

8.2.1 Yield Improvement

The yield improvement using the combined method, $Y_{\mathrm{COM}}(n,s)$, is directly obtained from simulation of the two-step algorithm described in the previous section. Table 6 shows the relative yield improvement realized as compared with yield $Y(n)$ of random stacking (without layer redundancy); the absolute value of $Y(n)$ is given in the 'Abs. yield' row. Inspecting the table reveals the following:

- Overall, the yield gain of the combined method outperforms that of layer redundancy (see Table 1) up to 64 %.
- Similarly as for layer redundancy, the memory yield improves irrespective of the considered stack size and number of redundant layers. Again, the yield improvement becomes significant as the stack size increases; this is because the occurrence probability of faulty layers increases.
- When using layer redundancy only, the addition of more redundant layers do not always result in better yield improvement. However, here it is the case for combined method; combining layer redundancy with wafer matching results in additional benefits that are larger than the yield loss due to stacking of extra layers.

8.2.2 Cost Improvement

To fairly evaluate the cost of this combined technique, both additional cost components for manufacturing and testing must be included. We define the cost improvement $C^G_{\mathrm{COM}}(n,s)$ as the cost of a good stacked IC using the combined approach.

$$\frac{C^G_{\mathrm{COM}}(n,s)}{C^G(n)} = \frac{C_{\mathrm{COM}}(s)}{C(n)} \cdot \frac{Y(n)}{Y_{\mathrm{COM}}(n,s)} \tag{39}$$

The relative cost change of this equation is depicted in Table 7. The combined method is interesting for $n \geq 3$ used with appropriate number of redundant layers r. The cost improves with larger stack sizes.

8.3 Comparison

In this last section, we compare the combined technique with the two stand-alone yield improvement techniques. The results of this comparison are shown in Table 8. The table contains five columns. The first column gives the considered stack size, the second column shows the number of redundant layers used for the combined method, the third column the yield improvement of the combined technique over no yield improvement scheme (i.e., random stacking without layer redundancy), the last two columns the yield improvement of the combined technique over the stand-alone versions. The table shows that for $n > 3$ the combined technique outperforms both layer redundancy and wafer matching. Thus, the combined approach is the best yield improvement technique to use.

9 Conclusion

This paper introduces the concept of layer redundancy and investigates it as a scheme to improve the compound yield of 3D stacked memories. It proposes an analytical model to evaluate the yield improvement due to layer redundancy.

Simulation results show that layer redundancy not only outperforms wafer matching (as a yield improvement scheme), but also realize a significant yield improvement, especially for larger stack size. For example, for a stack size of six layers and a die yield of 85 %, a relative yield improvement of 118.79 % is obtained using two redundant layers, while this is 14.03 % with wafer matching. The additional cost due to redundancy pays off; the cost of producing a good 3D stacked memory chip reduces with 37.68 % when using layer redundancy and only with 12.48 % when using wafer matching. Moreover, the results show that the benefits of layer redundancy become extremely significant for

lower die yields. Finally, we combined both methods technique to obtain even better improvements; e.g., for the six layered stack, the cost reduced from 38.45 to 42.51 %.

References

1. Adams RD (2003) High performance memory testing—design principles. In: Fault modeling and self-test. Kluwer Academic
2. Anigundi R, Hongbin S, Jian-Qiang L, Rose K, Tong Z (2009) Architecture design exploration of three-dimensional (3D) integrated DRAM. In: Quality of electronic design, pp 86–90
3. Bushnell M, Agrawal V (2000) Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. Wiley-VCH, Weinheim
4. Davis WR, Wilson J, Mick S, Xu J, Hua H, Mineo C, Sule AM, Steer M, Franzon PD (2005) Demystifying 3D ICs: the pros and cons of going vertical. IEEE Des Test Comput 22(8):498–510
5. de Vries DK (2005) Investigation of gross die per wafer formulas. IEEE Trans Semicond Manuf 18(1):136–139
6. Garrou P (2008) Christopher Bower and Peter Ramm. In: Handbook of 3D integration. Wiley-VCH
7. ITRS Report Yield Enhancement 2009 Edition. http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_Yield.pdf
8. Jiang L, Ye R, Xu Q (2010) Yield enhancement for 3D-stacked memory by redundancy sharing across dies. In: IEEE/ACM international conference on computer-aided design, pp 230–234
9. Kahng U et al (2010) 8 Gb 3-D DDR3 DRAM using through-silicon-via technology. IEEE J Solid-State Circuits 45:111–119
10. Kawano M et al (2006) A 3D packaging technology for 4 Gbit stacked DRAM with 3 Gbps data transfer. In: International electron devices meeting, pp 1–4
11. Kim E-K, Sung J (2008) Yield challenges in wafer stacking technology. In: Microelectronics reliability, pp 1102–1105
12. Lim K et al (2001) Bit line coupling scheme and electrical fuse circuit for reliable operation of high density DRAM. In: Symposium on VLSI circuits digest of technical papers, pp 33–34
13. Loh GH (2008) 3D-stacked memory architectures for multi-core processors. In: International symposium on computer architecture, pp 453–464
14. Marinissen EJ, Zorian Y (2009) Testing 3D chips containing through-silicon vias. In: International test conference, pp 1–11
15. Patti RS (2006) Three-dimensional integrated circuits and the future of system-on-chip designs. Proc IEEE 94(6):1214–1224
16. Puttaswamy K, Loh GH (2009) 3D-integrated SRAM components for high-performance microprocessors. IEEE Trans Comput 58(10):1369–1381
17. Reda S, Smith G, Smith L (2010) Maximizing the functional yield of wafer-to-wafer 3-D integration. IEEE Trans Very Large Scale Integr Syst 17(9):1357–1362
18. Reed P, Yeung G, Black B (2005) Design aspects of a microprocessor data cache using 3D die interconnect technology. In: International conference on integrated circuit design and technology, pp 15–18
19. Reese EA, Spaderna DW, Flannagan ST, Tsang F (1981) A 4K × 8 dynamic RAM with self-refresh. IEEE J Solid-State Circuits 16(5):479–487
20. Singh E (2011) Exploiting rotational symmetries for improved stacked yields in W2W 3D-SICs. In: VLSI test symposium, pp 32–37
21. Smith G, Smith L, Hosali S, Arkalgud S (2007) Yield considerations in the choice of 3D technology. In: IEEE international symposium on semiconductor manufacturing, pp 1–3
22. Taouil M, Hamdioui S (2011) Layer redundancy based yield improvement for 3D wafer-to-wafer stacked memories. In: European test symposium, pp 45–50
23. Taouil M, Hamdioui S, Beenakker K, Marinissen EJ (2010) Test cost analysis for 3D die-to-wafer stacking. In: Asian test symposium, pp 435–441
24. Taouil M, Hamdioui S, Verbree J, Marinissen EJ (2010) On maximizing the compound yield for 3D wafer-to-wafer stacked ICs. In: IEEE international test conference, pp 1–10
25. Tsai Y-F, Wang F, Xie Y, Vijaykrishnan N, Irwin MJ (2008) Design space exploration for 3-D cache. IEEE Trans Very Large Scale Integr Syst 16(4):444–455
26. Tummala R (2008) Fundamentals of microsystems packaging. McGraw-Hill Professional
27. Verbree J, Marinissen EJ, Roussel P, Velenis D (2010) On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking. In: IEEE European test symposium, pp 36–41
28. Wee J-K et al (2000) An antifuse EPROM circuitry scheme for field programmable repair in DRAMs. IEEE J Solid-State Circuits 35:1408–1414
29. Zhang T, Wang K, Feng Y, Song X, Duan L, Xie Y, Cheng X, Lin Y-L (2010) A customized design of DRAM controller for on-chip 3D DRAM stacking. In: IEEE Custom Integrated Circuits Conference (CICC), 19–22 Sept 2010, pp 1–4

Mottaqiallah Taouil received his MSc with honors from the Delft University of Technology (TUDelft), Delft, the Netherlands. He is currently pursuing a PhD at the Computer Engineering Lab of the same university in. His research interests include Reconfigurable Computing, Embedded Systems, VLSI Design & Test, Built-In-Self-Test, 3D stacked ICs, 3D Architectures, (3D) Design for Testability, (3D) Yield analysis and 3D Memory Test structures.

Said Hamdioui received his MSEE and PhD degrees (both with honors) from Delft University of Technology (TUDelft), Delft, The Netherlands. He is currently an associate professor at Computer Engineering Lab of TUDelft. Prior to joining TUDelft, Hamdioui worked for Microprocessor Products Group at Intel Corporation (in Santa Clara and Folsom, Califorina), for IP and Yield Group at Philips Semiconductors R&D (Crolles, France) and for DSP design group at Philips/NXP Semiconductors (Nijmegen, The Netherlands). He is the recipient of European Design Automation Association (EDAA) Outstanding Dissertation Award 2001, for his work on memory test techniques that have a wide-spread proliferation in the chip design industry; he is also the winner of the IEEE Nano and Nano Korea award at IEEE NANO 2010—Joint Symposium with Nano Korea 2010. He was nominated for The Young Academy (DJA) of the Royal Netherlands Academy of Arts and Sciences (KNAW) in 2009. His research interests include dependable nano-computing and VLSI Design & Test (defect/fault tolerance, reliability, security, nano-architectures, Design-for-Testability, Built-In-Self-Test, 3D stacked IC test, etc.). He has published one book and over 100 technical papers.

Is TSV-based 3D Integration Suitable for Inter-die Memory Repair?

Mihai Lefter, George R. Voicu, Mottaqiallah Taouil, Marius Enachescu, Said Hamdioui and Sorin D. Cotofana
Delft University of Technology, Delft, The Netherlands
E-mail: M.Lefter, G.R.Voicu, M.Taouil, M.Enachescu, S.Hamdioui, S.D.Cotofana @tudelft.nl

***Abstract*—In this paper we address lower level issues related to 3D inter-die memory repair in an attempt to evaluate the actual potential of this approach for current and foreseeable technology developments. We propose several implementation schemes both for inter-die row and column repair and evaluate their impact in terms of area and delay. Our analysis suggests that current state-of-the-art TSV dimensions allow inter-die column repair schemes at the expense of reasonable area overhead. For row repair, however, most memory con gurations require TSV dimensions to scale down at least with one order of magnitude in order to make this approach a possible candidate for 3D memory repair. We also performed a theoretical analysis of the implications of the proposed 3D repair schemes on the memory access time, which indicates that no substantial delay overhead is expected and that many delay versus energy consumption tradeoffs are possible.**

I. Introduction

Recent enhancements in Integrated Circuits (ICs) manufacturing process enable the fabrication of three dimensional stacked ICs (3D-SICs) based on Through-Silicon-Vias (TSVs) as die-to-die (D2D) interconnects, which further boost the trends of increasing transistor density and performance. 3D-SIC is an emerging technology, that, when compared with planar ICs, allows for smaller footprint, heterogeneous integration, higher interconnect density between stacked dies, and latency reduction mostly due to shorter wires [1].

3D memories have been proposed ever since the technology was introduced, one of the reason being their regular structure that allows them to be easily folded accross bitlines/wordlines and spread over multiple layers in a 3D embodiment [2]. Moreover, the typical area of a System on a Chip (SoC) is memory dominated, and, as the ITRS roadmap predicts that the trend of memory grow continues [3], it is expected that memories will play a critical role in 3D-SICs as most of the layers in the stack are likely to be allocated for storage.

As technology keeps shrinking towards meeting the requirements of increased density, capacity, and performance, IC circuits, memory arrays included, are more prone to degradation mechanism [4], and different sorts of defects during the manufacturing process [5]. In addition, the utilization of the still in its infancy 3D stacking technology increases the risk of low yield. To deal with this issue several works proposed inter-die memory repair, i.e., sharing redundant elements (rows/columns) between layers, in an attempt to increase the compound yield of memories [6], [7], [8], [9], [10], [11].

Up until now, all the work targeting inter-die memory repair primarily discussed the idea in principle, with no real implications being studied. The proposed approaches have been only evaluated via fault injection simulations and the obtained repair rate improvements form an upper bound. In order to achieve inter-die repair, a certain infrastructure has to be embedded into the memory such that spares can be made available to memory arrays in need that are located on remote dies. The added infrastructure must not affect the normal operation of the memory and may incur certain penalties in terms of area and/or delay which have not be studied.

In this paper we build upon previous work proposals and we further investigate the real implications of inter-die memory repair based on redundancy sharing. We first provide a classification of the possible access scenarios to memory arrays stacked in a 3D memory cube. Next, we propose several implementation schemes both for inter-die row and column repair in which we detail the circuit infrastructure required to support these access scenarios. For each scheme we propose the infrastructure, highlight its advantages and disadvantages, and discuss its impact on memory area and delay.

The area overhead is mostly dependent on the TSV size rather than on the extra logic. From our analysis it results that current state-of-the-art TSV dimensions allow inter-die column repair schemes with reasonable area overhead. For row repair, however, most memory configurations require TSV dimensions to scale down with at least one order of magnitude to make this approach applicable in practical 3D memory systems. We also performed a theoretical analysis of the implications of the proposed 3D repair schemes on the memory access time. Assuming a 20ps TSV delay our analysis indicates that for row repair the overhead is negligible and for column repair it can be in the same order of magnitude. This indicates that no substantial delay overhead is expected and that many delay versus energy consumption tradeoffs are possible.

The remaining of the paper is organized as follows. Section II briefly describes general memory repair techniques and related work regarding 3D memory repair. Section III defines the 3D memory repair architecture and the associated framework for inter-die redundancy. Section IV introduces the circuit infrastructure necessary to support inter-die memory repair. Section V considers various trade-offs and cost overhead in terms of area and delay. Finally, Section VI concludes the paper.

II. Redundancy Based Memory Repair

State of the art memory repair relies on the addition of several redundant resources to the memory arrays. These resources do not affect the interface or capacity of the memory, but can be later on utilized to substitute memory cells affected by, usual, permanent errors. Based on the physical placement of the spare elements we can broadly distinguish two types, that are not excluding one another, of memory redundancy: (i) *external redundancy*, in which a special smaller memory, external to the initial one is present, and where, based on a fault table, bad addresses are remapped by a Built-In Repair

978-3-9815370-0-0/DATE13/©2013 EDAA

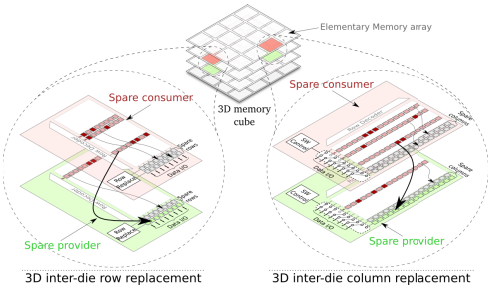


Fig. 1: 3D inter-die memory repair - general idea.

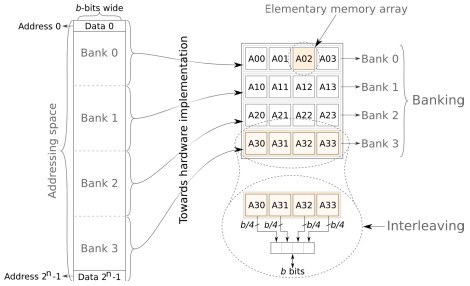


Fig. 2: Memory partitioning.

Analysis (BIRA) unit [12]; and (ii), *internal redundancy*, in which spare elements in the form of redundant rows and/or columns, are placed inside the memory alongside the normal columns and/or rows.

In this paper we consider *internal redundancy* only. Here, the mechanisms involved for row and column repair are quite different. For *row replacement*, detected faulty row addresses are stored in special registers. Whenever the memory is accessed, the incoming address is first compared with those stored in the special registers to check if a defective row is to be accessed. If this is the case, the output of the comparator disables the row decoder and activates the spare wordline. For *column replacement*, in general, a column switching mechanism is present to isolate the faulty column and to forward data from non-defective cells [13].

To create extra opportunities for memory repair various inter-die approaches have been proposed. In [7] a Die-to-Die (D2D) stacking flow algorithm is presented which assumes that each die is beforehand locally repaired such that the number of available (not utilized for local repair) spares are made available as inputs for the global repair algorithm. This method for inter-die column replacement is suitable only for the particular case where arrays are simultaneously accessed with the same address. A similar D2D stacking approach is considered in [11] where the die stacking flow is modeled as a bipartite graph maximal matching problem. Several global D2D matching algorithms without local repair first are introduced and compared in [8]. An interesting approach is introduced in [9] where the authors propose to recycle irreparable dies (i.e., dies with arrays that are not repairable if only local spares are considered) in order to create good working memories.

III. 3D Inter-die Memory Repair Architecture

The considered memory arrangement resembles a memory cube, as depicted in Fig. 1. The cube employs 3D *array stacking* with the identical memory arrays being equipped with redundant rows and/or columns. In this organization, a situation may arise in which arrays with insufficient redundancy are in the vertical proximity of arrays that still have unutilized redundant elements. Supporting the replacement of faulty cells by using redundant resources from arrays from other dies, i.e., inter-die spare replacement, results in extended memory reparability rates [8]. This can be observed on the lower part of Fig. 1, where the top arrays have utilized all their available spare rows/columns (two in this case) and still have one faulty row uncovered. However, the bottom arrays can provide the necessary spare rows/columns to replace the faulty ones on the top arrays to make the memory defect free.

We define the arrays that have available spare resources as *spare providers* and arrays which make use of the externally available spare resources as *spare consumers*. For the 3D memory repair to function correctly the consumer must be able to retrieve/store data from/on the provider in a transparent manner, i.e., the provider must be able to function normally, despite its spares being accessed by a neighboring die. In addition, it is important that the inter-die repair infrastructure does not disrupt the functionality of the memory cube when no repair takes place. Therefore, the required infrastructure that assures the memory repair mechanism is highly dependent on the exact internal structure of the memory arrays.

In order to balance area, delay, and power tradeoffs, a large memory is usually constructed in a hierarchical manner and is composed out of several banks, with each bank being further divided in several arrays. An example is presented in Fig. 2 where the partitioning employs banking and interleaving. Each bank can be accessed either concurrently with independent addresses, or sequentially, where one bank is accessed while the rest remain idle. For interleaving, however, all the subarrays of a bank are concurrently accessed with the same address.

As the internal organization of the memory cube is defined at design time, a fixed memory partitioning implies three exclusive situations in which two memory arrays, a *provider-consumer* pair, can be accessed: (i) **Idle provider** - the two arrays are located in different banks that are never concurrently accessed; we use the term idle to denote that the two arrays are never accessed at the same time; from the *consumer*'s perspective this is equivalent with the *provider* being always idle; (ii) **Busy provider with different access pattern** - the two arrays are located in different banks that are concurrently accessed with independent addresses; (iii) **Busy provider with same access pattern** - the two arrays are part of the same bank with interleaving, therefore the accessing address is the same.

We add that, although our proposal is general and can in principle be applied to more than two adjacent dies, in this paper we consider that inter-die replacement is performed between exclusive pairs of adjacent dies. The reasons behind this restriction are as follows: (i) the infrastructure overhead grows with the number of dies involved in the spare sharing process, and, (ii) two dies spare replacement is enough to sustain a satisfactory yield [8], since the die yield has a high value after repair. In the next section we introduce the infrastructure for the above identified *provider-consumer* repair schemes.

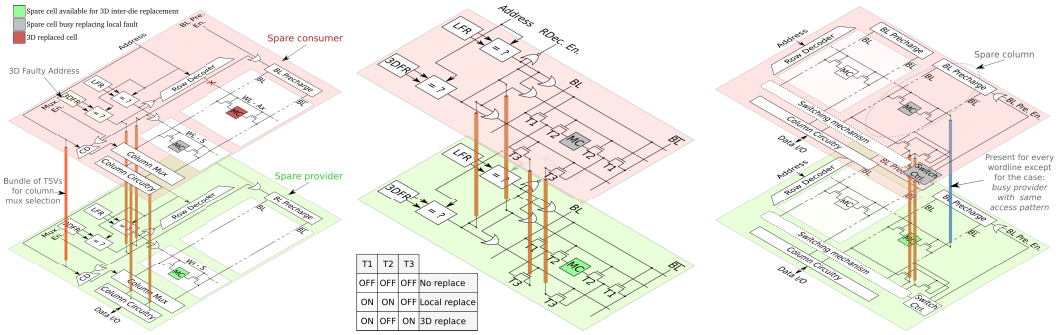


Fig. 3: Inter-die row (a,b) and column (c) replacement infrastructure.

IV. 3D Inter-die Memory Repair Infrastructure

In this section we detail the inter-die repair schemes for each of the three *provider-consumer* pair scenarios introduced in Section III for row and column replacement.

A. Inter-die Row Replacement

1) Idle provider: Fig. 3a depicts the situation for the case in which the *provider* is idle. On the *consumer* side, the local spare row is already allocated and another faulty row needs to be replaced remotely. A register is required to store its address (3DFR - 3D Fault Register), in a similar fashion as in the local replacement scheme. Furthermore, a comparator and several logic gates are introduced in the design to disable the local row decoder and to activate the spare wordline on the *provider* side whenever the incoming address is equal to the value stored in 3DFR. We propose to place the data TSVs after the column multiplexer. This requires the column address to be transfered through TSVs and the column decoder (CD) on the *provider* to be enabled. In this manner fewer TSVs are required when compared to the case when TSVs are placed for every bitline.

2) Busy provider with different access pattern: When both *consumer* and *provider* can be accessed in parallel the constraints imposed to the inter-die memory repair interface are tighter, making the infrastructure more complex. In particular, when an inter-die replacement occurs, both *consumer* and *provider* need to use the *provider*'s bitlines, giving rise to a conflict. For this reason the data TSVs cannot be placed after the column muxes and extra transistors (denoted by T2 and T3) are required in every spare memory cell, as in Fig. 3b.

3) Busy provider with same access pattern: A particular case of *provider* and *consumer* parallel access arises when their address is the same (i.e., in an interleaving organization of memory banks, see Section III). In contrast with the previous scheme, the infrastructure can be reduced in terms of logic. However, each spare cell still needs to be augmented with 4 extra transistors and 2 TSVs. Thus, even if we assume that future TSV manufacturing process will be greatly improved to a negligible size, the cell area almost doubles.

B. Inter-die Column Replacement

The general infrastructure required for inter-die column replacement depicted in Fig. 3c comprises all the cases introduced in Section III. The common part for all the cases consists of the TSV pair utilized for bitline value transmission. They are enabled by the switching control block, which needs to be adapted to control also the inter-die replacement mechanism.

A special TSV is required for every wordline whenever the *provider* is busy accessing a different address, or when it is idle, in order to assert the required wordline for the *consumer*. When the *provider* is busy it is also mandatory to decouple the *provider's* wordline such that no bitline conflict arises because of multiple wordlines assertion. For brevity this action is not represented in Fig. 3c. For the case in which the *provider* is idle, the TSV required for the wordline activation may be discarded if the *provider*'s row decoder is enabled. However, this requires the consumer's address to be driven onto the TSVs. Nevertheless, the gain is a significant TSV reduction.

The easiest and most convenient inter-die column replacement scheme in terms of TSV requirements is by far when the *consumer* and the *provider* are busy accessing the same address. Here, the same wordline is asserted in both arrays and no bitlines conflicts occur.

V. Discussion

In this section we discuss the overhead of the 3D inter-die memory repair schemes in terms of area and delay.

Area represents a sensitive issue in memory design and the memory cell is particulary the subject of severe scaling. SRAM bit cell has followed Moores's law, with an area shrinking rate of about $1/2$ for every generation, reaching $0.081\ \mu m^2$ for the $22\ nm$ technology [14]. This rate is expected to last even in the realm of post-CMOS devices [15]. TSVs dimensions are predicted to scale down too, but not that steep as SRAM bit cells. The predictions from [16] suggest a gradually decreasing trend with a shrinking ratio of about $1/4$ for every 3 years, reaching a minimum diameter of $0.8\ \mu m$ and a pitch of $1.6\ \mu m$ by 2018. Nowadays manufactured TSVs have a diameter bewteen 3 and $10\ \mu m$ and a pitch of about $10\ \mu m$ [17], [18], [19]. From their large size it is clear that TSVs represent the major contributor to the 3D memory repair area overhead.

Table I presents the TSV requirements for all the scenarios introduced in Section III. The scenarios that have the least number of TSVs are "idle *provider*" for row redundancy and "busy *provider* with same access" for column redundancy. All the other scenarios require a large number of TSVs that make them absolutely impractical. Even for the row redundancy with the "idle *provider*" scenario the practicality is problematic. Fig. 4 depicts the area of one redundant row and its required

TABLE I: TSV REQUIREMENTS FOR PROPOSED SCHEMES

Memory access scenarios	Number of TSVs
Row replacement	
Idle *provider*	$2 \times spares + 2 \times dw + cd\_bits$
Busy *provider* with different address	$spares \times (2 + 2 \times columns)$
Busy *provider* with same address	$spares \times (2 + 2 \times columns)$
Column replacement	
Idle *provider*	$2 \times spares + log_2(rows)$
Busy *provider* with different address	$2 \times spares + rows$
Busy *provider* with same address	$2 \times spares$

* *dw* = data width (data I/O); *cd_bits* = column decoder input bits.

TSVs. The redundant row area is drawn for different technology nodes using memory widths varying between 128 and 2048 bits. The TSV area is independent of the technology node and is calculated for a TSV pitch between 0.5 and 3.5 μm and a memory data width of 32 bits. Given that, inter-die redundancy may be profitable only if the redundant row area is smaller than the TSV area. Fig. 4 clearly suggests that for a large TSV pitch inter-die redundancy becomes impractical.

It is interesting be to find the required TSV pitch for which inter-die row replacement becomes advantageous. For example, in case the column width is 512 and the data output width is 32, the TSV pitch must be at most 534 nm. For the worst case considered configuration, with a column width of 512 and data output width of 64, TSV pitch needs even to scale further down to 388 nm. Therefore, current TSV sizes in the order of 3 μm need to be shrinked severely for inter-die redundancy to be beneficial for a wide range of memory configurations.

The access time (T_{N2D}) for a normal memory read operation (Eq. (1)) is determined by: address decoding (T_{dec}), wordline generation (T_{WL}), bitlines discharge (T_{BL}), column multiplexing (T_{mux}), and data sensing (T_{SA}). If row redundancy is present and the redundant row is accessed the access time changes to T_{R2D} (Eq. (2)), because the access goes through the comparator (T_{cmp}) instead of the decoder. For 3D row redundancy, extra time is required to transfer data to the *consumer* through the TSVs (T_{TSV}), resulting in T_{R3D} (Eq. (3)). The time overhead for 3D row redundancy (D_{OR}) can be computed as in Eq. (4).

For 3D inter-die column redundancy, the access time increases as in Eq. (6). Thus, there is always a delay overhead (D_{OC}) due to TSV propagation and switching time.

$$T_{N2D} = T_{dec} + T_{WL} + T_{BL} + T_{mux} + T_{SA} \quad (1)$$

$$T_{R2D} = T_{cmp} + T_{WL} + T_{BL} + T_{mux} + T_{SA} \quad (2)$$

$$T_{R3D} = T_{R2D} + 2 \times T_{TSV} \quad (3)$$

$$D_{OR} = \frac{max(T_{N2D}, T_{R3D}) - max(T_{N2D}, T_{R2D})}{max(T_{N2D}, T_{R2D})} \quad (4)$$

$$T_{C2D} = T_{N2D} + T_{switching} \quad (5)$$

$$T_{C3D} = T_{C2D} + T_{switching} \quad (6)$$

$$D_{OC} = \frac{T_{switching} + T_{TSV}}{T_{C2D}} \quad (7)$$

As the delay of a TSV is in the order of 20 ps [20], we expect the following to hold true: $T_{R2D} < T_{R3D} < T_{N2D}$. Therefore, no delay penalty for row repair is expected. For column repair however, the following inequation holds: $T_{N2D} < T_{C2D} < T_{C3D}$. The overhead is determined by the delay of switching muxes and a TSV ($T_{switching} + T_{TSV}$) which is expected to be minimal.

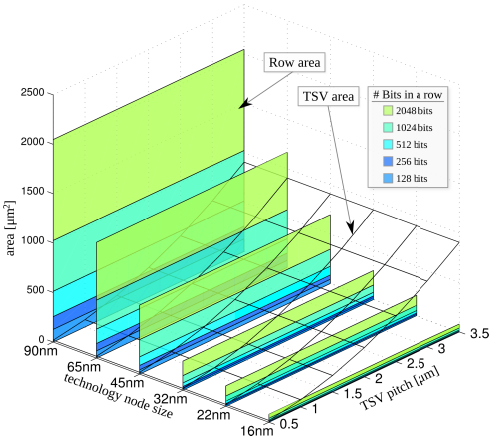


Fig. 4: TSV area overhead vs. row area for 32-bit data I/O.

VI. CONCLUSIONS

In this paper, we presented a study of inter-die repair schemes for TSV based 3D-SICs, i.e., of using repair in the vertical dimension.The paper provided an overview of general repair schemes and subsequently, proposed a memory framework for inter-die redundancy based on a *provider-consumer* pair scheme. Our analysis suggests that for state-of-the-art TSV dimensions inter-die column-based repair schemes could result in yield improvements at a reasonable area overhead. For row repair, however, most memory configurations require TSV dimensions to scale down at least with one order of magnitude to be utilized in 3D memory systems.

REFERENCES

[1] P. Garrou, *Handbook of 3D integration : technology and applications of 3D integrated circuits*. Weinheim: Wiley-VCH, 2008.
[2] K. Puttaswamy *et al.*, "3D-Integrated SRAM components for high-performance microprocessors," *TC*, 2009.
[3] "ITRS - System Drivers," Tech. Rep., 2011. http://www.itrs.net
[4] S. Rusu *et al.*, "Trends and challenges in VLSI technology scaling towards 100nm," in *VLSI Design / ASPDAC*, 2002.
[5] S. R. Nassif, "The light at the end of the CMOS tunnel," *ASAP*, 2010.
[6] R. Anigundi *et al.*, "Architecture design exploration of three-dimensional (3D) integrated DRAM," in *ISQED*, 2009.
[7] C. Chou *et al.*, "Yield-enhancement techniques for 3D random access memories," in *VLSI-DAT*, 2010.
[8] L. Jiang *et al.*, "Yield enhancement for 3D-stacked memory by redundancy sharing across dies," in *ICCAD*, 2010.
[9] Y.-F. Chou *et al.*, "Yield enhancement by bad-die recycling and stacking with though-silicon vias," *TVLSI*, 2011.
[10] C.-W. Wu *et al.*, "On test and repair of 3D random access memory," in *ASPDAC*, 2012.
[11] S. Lu *et al.*, "Yield enhancement techniques for 3-dimensional random access memories," *Microelectronics Reliability*, 2012.
[12] N. Axelos *et al.*, "Efficient memory repair using cache-based redundancy," *TVLSI*, 2011.
[13] M. Horiguchi *et al.*, *Nanoscale Memory Repair*. Springer, 2011.
[14] K. Smith *et al.*, "Through the looking glass: Trend tracking for ISSCC 2012," *M-JSSC*, 2012.
[15] H. Iwai, "Roadmap for 22nm and beyond," *Microelectronic Eng.*, 2009.
[16] "ITRS - Interconect," Tech. Rep., 2011. http://www.itrs.net
[17] C. L. Yu *et al.*, "TSV process optimization for reduced device impact on 28nm CMOS," in *TVLSI*, 2011.
[18] G. Katti *et al.*, "3D stacked ICs using cu TSVs and die to wafer hybrid collective bonding," in *IEDM*, 2009.
[19] H. Chaabouni *et al.*, "Investigation on TSV impact on 65nm CMOS devices and circuits," in *IEDM*, 2010.
[20] D. H. Kim *et al.*, "Through-silicon-via-aware delay and power prediction model for buffered interconnects in 3D ICs," in *SLIP*, 2010.

Appendix B

Publications - Cost Model

This chapter presents the publications on cost modeling. The following papers are included:

B1: **M. Taouil** and S. Hamdioui, "On Optimizing Test Cost for Wafer-to-Wafer 3D-Stacked ICs," *7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Tunis, Tunisia, May 2012, pp. 1–6.

B2: **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking," *19th IEEE Asian Test Symposium (ATS)*, Shanghai, China, Dec. 2010, pp. 435–441.

B3: **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "How Significant will be the Test Cost Share for 3D Die-to-Wafer Stacked-ICs?" *6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Athens, Greece, April 2011, pp. 1–6.

B4: **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Impact on the Overall Die-to-Wafer 3D Stacked IC Cost," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 1, pp. 15-25, Feb. 2012.

B5: **M. Taouil** and S. Hamdioui, "Stacking Order Impact on Overall 3D Die-to-Wafer Stacked-IC Cost," *14th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Cottbus, Germany, April 2011, pp. 335–340.

B6: **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," *6th IEEE International Design and Test Workshop (IDT)*, Beirut, Lebanon, Dec. 2011, pp. 24–29.

B7: **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Using 3D-COSTAR for 2.5D Test Cost Optimization," *IEEE International 3D Systems Integration Conference (3DIC)*, San Fransisco, CA, USA, Oct. 2013, pp. 1–8.

B8: **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Impact of Mid-Bond Testing in 3D Stacked ICs," *16th IEEE Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, New York, NY, USA, Oct. 2013, pp. 178–183.

B9: **M.Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Quality versus Cost Analysis for 3D Stacked ICs," *32nd IEEE VLSI Test Symposium (VTS)*, Napa, CA, USA, April 2014, pp. 1–6.

B10: E.J. Marinissen, B. de Wachter, K. Smith, J. Kiesewetter, **M.Taouil**, and S. Hamdioui, "Direct Probing on Large-Array Fine-Pitch Micro-Bumps of a Wide-I/O Logic-Memory Interface," *International Test Conference (ITC)*, Seattle, WA, Oct. 2014, pp. 1–10.

2012 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

On Optimizing Test Cost for Wafer-to-Wafer 3D-Stacked ICs

Mottaqiallah Taouil Said Hamdioui

Computer Engineering Laboratory
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: {M.Taouil, S.Hamdioui}@tudelft.nl

***Abstract*— The increasing demand for more sophisticated ICs with more functionality mostly was realized by downscaling and increasing the number of transistors. A technology that promises further increase of transistor density (in addition with heterogeneous integration, better performance and less power dissipation at a smaller footprint) is the three-dimensional stacked ICs (3D-SICs). Several stacking approaches are under development to manufacture such 3D-SICs. Wafer-to-Wafer (W2W) stacking seems the most favorable approach when high manufacturing throughput, thinned wafers and small die handling is required. However, efficient and optimal test approaches to satisfy the required quality are still subject to research. Each manufactured 3D-SIC undergoes a test and therefore optimizing test cost will have a large overall impact. This paper discusses test cost optimization for W2W 3D-SICs. It first introduces a framework covering different test flows for 3D W2W ICs. Test flows that include pre-bond tests can benefit from wafer matching; in wafer matching a software algorithm is used to increase the compound yield by stacking wafers with similar fault distributions. Subsequently, the paper proposes a cost model to evaluate and estimate the impact of test flows on the overall 3D-SIC cost. Our simulation results show that test flows with pre-bond testing in general significantly reduce the overall cost. These test flows benefit mostly from the yield increase due to wafer matching.**

Keywords: *W2W, 3D W2W test flows, pre-bond testing, wafer matching*

I. Introduction

The increasing demand for more functionality on ICs has been met by the semiconductor industry adhering to Moore's law. Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs), which are electrically interconnected by Through Silicon Vias (TSV). This opened up new research directions that could be investigated to continue the trend of performance increase. A TSV based 3D-SIC is an emerging technology that provides a smaller footprint, higher interconnect density between stacked dies, higher performance and lower power consumption due to shorter wires as compared to planar ICs [1]. Moreover, heterogeneous integration in 3D-SICs allows dies to be manufactured with dissimilar processing and technology nodes; for example, memory layers can be stacked on a processor.

The key manufacturing steps in assembling 3D-SICs are the stacking and the bonding of dies. The three existing bonding methods are Die-to-Die (D2D), Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) bonding [2]. High alignment accuracy is feasible in D2D and D2W bonding, but it impacts the throughput negatively. In D2D and D2W bonding, Known Good Die (KGD) stacking can be applied to prevent faulty dies from being stacked [2]. W2W stacking allows for (a) high manufacturing throughput due to single wafer alignment, and (b) thinned wafers and small die handling. Due to their regularity, memories and FPGAs are very attractive to be used in W2W stacking. However, the major drawback of W2W stacking is the low compound yield especially with increased number of stacked layers.

Increasing compound yield in W2W stacking has been addressed recently by some authors [3–6]; all of them use wafer matching; i.e., a technique in which wafers with similar die fault distributions are stacked. To use wafer matching, wafers must be tested before bonding them; i.e., pre-bond test. Using appropriate test strategies/flows will therefore have a large impact on the compound yield. This topic is also discussed in [5,6]. However, the works presented by the authors have many limitations. For instance, they consider only three test flows; in addition, not all cost components were considered when the overall 3D-SIC cost was determined, etc.

This paper explores the whole space of test flows for W2W 3D-SICs and their impact on the overall cost and compound yield, while considering all cost components such as manufacturing, test, packaging etc. The main contributions of this paper are:

- A classification of 3D W2W test flows.
- An analytical model that formulates the cost of 3D-SICs for W2W stacked ICs.
- Analysis of the impact of test flows on the compound yield and overall cost.

The remainder of this paper is organized as follows. Section II discusses the related prior work in wafer matching and 3D-SIC testing. Section III discusses the test framework. Section IV presents the cost model. Section V describes the performed experiments. The simulation results are analyzed in Section VI. Finally, Section VII concludes this paper.

II. Related Prior Work

This section discusses the prior work in wafer matching and 3D-SIC testing in Sections II-A and II-B respectively.

978-1-4673-1928-7/12/$31.00 ©2012 IEEE

2012 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

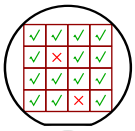


Fig. 1. Pre-bond tested wafer with good and bad dies.

A. Wafer matching

Wafer matching is a technique to improve the compound yield by stacking wafers with the same or similar die fault distributions (fault maps). This technique has been addressed recently by some authors [3–6]; different wafer matching techniques have been introduced.

The authors in [3–5] use static repositories to perform the wafer matching, while in [6] the authors use running repositories. In [6], the authors consider different matching criteria as well. In a static repository, wafers are only replenished after the whole cassette is empty, while in a running repository selected wafers are immediately replenished. In addition, [4] uses the symmetrical structure of a wafer to increase the matching combination (by rotating wafers).

Wafer matching necessitates pre-bond tests to obtain the fault map distributions of the dies on the wafer. Figure 1 shows an example of a wafer with sixteen dies, where two dies have been identified faulty during pre-bond testing. After collecting several wafers of each layer of the stack in different repositories, matching between the repositories can take place.

The pre-bond test cost only pays off in case sufficient compound yield is realized. This yield improvement can be significant in case of a large stack size or low die yield [5]. However, this yield improvement decreases for higher die yield. For example, for a stack size of two layers with a die yield of 85% and 1278 dies per wafer, wafer matching is able to increase the compound yield from 72.3% (for random stacking) to 73.1% [6]. This dilemma motives us to analyze the cost trade-off between pre-bond test cost and yield increase for the different test flows.

B. Testing

Optimizing test cost is a challenge that can significantly contribute to the overall cost reduction. Choosing an optimal and efficient test flow requires the analysis of all possible flows using an appropriate test cost model. Research on this topic is still in its infancy stage and very limited work is published [7–9]. In [7], the author considered a manufacturing cost model for 3D monolithic memory integrated circuits; cost improvement of 3D with respect to 2D (for different 3D stack sizes) was modeled. In [8], the authors developed a 3D-cost model to determine the optimal stack size for a given 3D-SICs circuit, where they restricted the variable parameters to only die yield and die size. In [9], the authors proposed a 3D cost model for Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) stacking. However, none of these published work is able to model the impact of the test cost on the overall 3D-SIC cost since none of them considers the different test moments and test flows. In our previous work [10], a basic cost model considering the impact of different test flows on the overall 3D-SIC cost was presented. A refined version of such a model, where many limitation are addressed, is presented in [11]. However, both were limited to D2W stacking in which a freedom exist to perform Known Good Die (KGD) stacking.

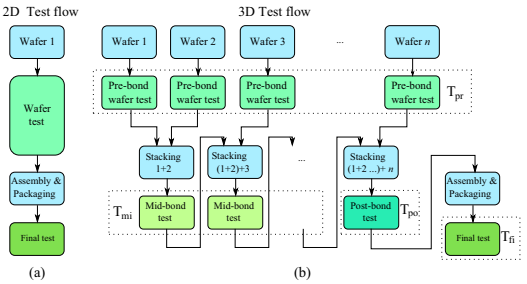


Fig. 2. 2D versus 3D D2W test flows

III. W2W Test Framework

In this section, we derive a test framework consisting of test flows for W2W stacked 3D-SICs. First, Section III-A describes the possible test moments in time. Thereafter, test flows are compiled into a framework in Section III-B by applying different tests at the considered test moments.

A. 3D Test Moments

For conventional testing of 2D ICs, two types of tests can be defined (as shown in Figure 2(a) [12]): a wafer test and a final test. A wafer test screens out faulty ICs prior to assembly and packaging in order to prevent unnecessary packaging costs, while a final test guarantees the quality of the packaged chip to reduce test escapes. A trade-off between the additional wafer test costs versus savings in packaging cost determines the applicability of this test. Furthermore, the test decision is based on the manufacturing yield and fault coverage. In case the yield is high enough, the test can be skipped or performed at low cost (i.e., low fault coverage).

For 3D SICs, additional tests -such as partial created stack tests- be defined. Figure 1(b) shows the natural test moments during the manufacturing of 3D-SICs. Four test moments can be distinguished in time, as depicted in Figure 2(b) and explained next.

1) T_{pr}: n pre-bond wafer tests, since there are n layers to be stacked. T_{pr} tests prevent faulty dies entering the stack. Two different types of test can be applied here. Traditional functionality of the chip can be tested for, but also preliminary TSV tests can be applied (in case of via-first [13]) as well.
2) T_{mi}: $n-2$ mid-bond tests applicable for partial created stacks. In this case, either dies, interconnects formed by the TSVs between them, a combination of the former two or none of them can be tested. Good tested dies in the pre-bond test phase could get corrupted

2012 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

TABLE I
3D TEST FLOWS

Flow	Pre-bond	Post-bond	Final
t1	-	-	int → die
t2a	-	int	int → die
t2b	-	die	int → die
t2c	-	int → die	int → die
t3	yes	-	int → die
t4a	yes	int	int → die
t4b	yes	die	int → die
t4c	yes	int → die	int → die

during the stacking process as a consequence of e.g., die thinning, and bonding [14].

3) T_{po}: one post-bond test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Here, both dies and interconnects between them can be tested for.
4) T_{fi}: one final test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied at this test moment as well.

Note that in total $2{\cdot}n$ different test moments can be identified versus 2 test moments for planar ICs. A 3D test flow can be defined as a combination of tests applied at the four test moments.

However, in this paper, mid-bond tests T_{mi} are ignored as dies are stacked based on the wafer level. Intermediate tests can not prevent faulty dies to be stacked as the case is for D2W stacking.

B. W2W Testflows

From the test moments of the previous section, 8 test flows are derived and depicted in Table I. The first column denotes the name of the particular test flow. The second column specifies whether a pre-bond test is performed or not. This pre-bond test consists of either a die, TSV or die and TSV test. The more sophisticated the pre-bond test, the more faulty dies can be identified. This increases the effectiveness of wafer matching at a higher test cost.

The third column presents the performed test during post-bond. Here, the option exist to skip this test, or to test for interconnects, dies, or both of them. In case both interconnects and dies are tested, the symbol → is used to denote the test sequence order, i.e., for int → die interconnects are tested prior dies. We do no consider the die → int for three reasons: (1) testing of dies is assumed to be much more expensive (more test vectors), (2) prior to test the dies in the 3D stack, interconnects that access that die must be tested, (3) to obtain a manageable space of test flows.

The last column of the table specifies the final test after IC packaging. The applied test in this stage determines the quality, test escapes of the product. We assume that a full test is performed in this final test phase.

IV. COST MODEL

Obviously, in order to determine the most cost-effective test flow the test cost should be specified. However, this is by far not enough to produce a fair comparison of test flows. Other cost classes have to be specified as input requirements as they have a large impact on the overall cost as well. We consider three classes: manufacturing, test and packaging as depicted in Figure 3.

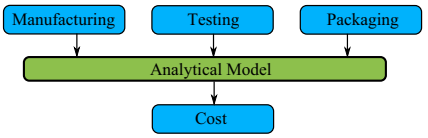


Fig. 3. Cost Model Interface.

- Manufacturing cost: it covers two types of parameters and are related to cost and yield. The most obvious ones related to 3D are the die size/cost and stacking operation. The stack yield is determined by the yield of the dies that enter the stack, the yield of the interconnects between the dies and the yield of the stacking operation.
- Test cost: this is related to the required cost associated with (a) pre-bond test, (b) mid-bond test, (c) post-bond test and (d) final test as defined previously. A test consists of two parts, a test for interconnects between the stacked dies and the dies themselves. The vertical interconnects are new in the stack and testing them after stacking seems rational.
- Packaging cost: the assembly and packaging cost.

The parameters used to define these classes are described in Section V-A. In the remainder of this section we define the cost evaluation.

The cost per good 3D-SIC C_{GD} can be defined by:

$$C_{GD} = \frac{\sum_{i=1}^{n} C_{die,i} + \sum_{i=1}^{n-1} C_{3D,i} + C_t + C_p}{Y_s} \quad (1)$$

Here, $C_{die,i}$ represents the manufacturing cost of the die on layer i; in total there are n stacked layers. The parameter $C_{3D,i}$ denotes the stacking cost for a 3D-SIC. Note that an n-layered stack only requires n-1 stacking operations. C_t represents the test cost, C_p the packaging cost and Y_s is the overall stack yield per 3D-SIC.

Equation 1 can be written into:

$$C_{GD} = \frac{n \cdot C_w + (n-1) \cdot \gamma \cdot C_w + r_p \cdot \beta \cdot C_w + C_t}{Y_s \cdot d} \quad (2)$$

Here, $\gamma = \frac{C_{3D}}{C_w}$ the ratio between the 3D stacking and wafer cost, $\beta = \frac{C_p}{C_w}$ the ratio between packaging and the wafer cost. Here, C_w is the cost per wafer (we assume this is equal for each layer as in memories), r_p the fraction of 3D-SICs that are packaged per stacked wafer set, C_p the packaging cost per 3D-SIC, C_t the test cost. Finally, Y_s and d represent the overall yield of the 3D-SIC and the number of dies per wafer respectively.

V. CASE-STUDY

A. Experiment setup

a) Manufacturing: The manufacturing class includes parameters related to the manufacturing of 3D-SICs such

2012 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

as wafer cost, costs required for wafer processing, TSV fabrication and 3D stacking/bonding. However, it includes also parameters related to the die and stack yield.

For wafers and their processing, we used the cost models of [15] and [16]; the total price of a 300 mm wafer is estimated at approximately $2779. The model in [15] considers a variety of costs, including installation, maintenance, lithography and material. For TSV fabrication, the work of EMC-3D consortium [17] is used; the cost to fabricate 5 μm TSVs in a single wafer is assumed to be $190 and these cost are additive to the wafer cost. We assume the cost of manufacturing TSVs to be 60% of the total 3D cost [18].

The die yield is based on the stacking process in [5], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. The work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50\ mm^2$, the number of Gross Dies per Wafer (GDW) are estimated to be d=1278 [19]. With the negative binomial formula for yield, a die yield of $Y_D = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [20]. For the stack size we assume a default stack size n=2. The stacking yield is composed out of two parameters: the interconnect (TSV) yield Y_{INT} and the stacked-die yield Y_{SD}. In our simulations, the interconnect yield Y_{INT} is considered to be 97%. For the good dies that enter the stack, a small probability exists that they get corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 99%, similar as in [5].

The compound yield of a 3D-SIC can be formulated as follows in case no wafer matching is used:

$$Y_s = Y_D^n \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)} \tag{3}$$

In case wafer matching is used, this expression can formulated by

$$Y_s = Y(n,k) \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)} \tag{4}$$

where $Y(n,k)$ the compound yield of the dies after being matched using a repository with k wafers. In case the repository size is k=1, $Y(n,k) = Y_d^n$.

b) Test: The test class consists of parameters that are related to the test cost of dies and interconnects in the stack and to the test flows.

To estimate the test cost per die, the model in [20] is used; it includes depreciation, maintenance and operating cost and assumes five ATE machines operating simultaneously. The derived test cost equals t_{die}=3.82 $cent/second per die. Assuming a test time of 6 seconds per die, the test cost will be t_{int}=$0.23 per die. To estimate the interconnect test cost, a ratio of 1:100 between the test time of dies and interconnects is assumed (as in [5]).

The cost related to each test flow depends on the number of tests that are performed. The test cost for each test flow is the sum of the test costs in the pre-bond (t_{pr}), in the post-bond (t_{po}) and final phase (t_{fi}), hence C_t = t_{pr} + t_{po} + t_{fi}. Table II shows this cost. For example, in test flow t1a only a final test is applied. Here, all the interconnects are tested at a cost equal to $n_i = (n-1) \cdot d \cdot t_{int}$. After the interconnects are tested, only the non-faulty dies are further tested at a cost equal to $Y_{INT}^{(n-1)} \cdot n_d$. Here, $n_d = n \cdot d \cdot t_{die}$ presents the test cost for all the dies. As a second example, consider the test flow t3a that contains a pre-bond test. All dies are tested at a cost of n_d in this phase. Due to wafer matching, it already know that the yield of the dies that enter the stack equals $Y(n,k)$. Therefore, in the final test stage only the interconnects of the good stacks have to be tested at a cost $Y(n,k) \cdot n_i$. After new faulty interconnects have been detected the cost to test the remain dies equals $Y(n,k) \cdot Y_{INT}^{(n-1)} \cdot n_d$.

TABLE II
TEST COST

flow	t_{pr}	t_{po}	t_{fi}
t1a	-	-	$n_i + Y_{INT}^{(n-1)} \cdot n_d$
t2a	-	n_i	$Y_{INT}^{(n-1)} \cdot (n_i + n_d)$
t2b	-	n_d	$Y_D^n \cdot Y_{SD}^{(n-1)} \cdot \left\{n_i + Y_{INT}^{(n-1)} \cdot n_d\right\}$
t2c	-	$n_i + Y_{INT}^{(n-1)} \cdot n_d$	$Y_D^n \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)} \cdot \{n_i + n_d\}$
t3a	n_d	-	$Y(n,k) \cdot \left\{n_i + Y_{INT}^{(n-1)} \cdot n_d\right\}$
t4a	n_d	$Y(n,k) \cdot n_i$	$Y(n,k) \cdot Y_{INT}^{(n-1)} \cdot (n_i + n_d)$
t4b	n_d	$Y(n,k) \cdot n_d$	$Y(n,k) \cdot Y_{SD}^{(n-1)} \cdot \left\{n_i + Y_{INT}^{(n-1)} \cdot n_d\right\}$
t4c	n_d	$Y(n,k) \cdot \left\{n_i + Y_{INT}^{(n-1)} \cdot n_d\right\}$	$Y(n,k) \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)} \cdot \{n_i + n_d\}$

TABLE III
PACKAGING COST

Test flow	Pre-bond	r_p
t1a	no	1
t2a	no	$Y_{INT}^{(n-1)}$
t2b	no	$Y_D^n \cdot Y_{SD}^{(n-1)}$
t2c	no	$Y_D^n \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)}$
t3a	yes	$Y(n,k)$
t4a	yes	$Y(n,k) \cdot Y_{INT}^{(n-1)}$
t4b	yes	$Y(n,k) \cdot Y_{SD}^{(n-1)}$
t4c	yes	$Y(n,k) \cdot Y_{SD}^{(n-1)} \cdot Y_{INT}^{(n-1)}$

c) Packaging: The packaging cost forms a significant fraction of the overall cost and depends on the used technique [21]. In this paper, we assume the packaging cost to be 50% of the wafer cost. The overall packaging cost depends on the number of packaged ICs which depends on the selected test flow. For example, in test flow t1a where only a final test is applied, all ICs are packaged (i.e., the packaging ratio r_p=1). Table III summarizes the packaging ratios of each test flow. The table consists of 3 columns; the first column depicts the test flow, the second column shows whether it is pre-bond tested or not, and the last column contains the ratio of packaged 3D-SICs. The ratio of packaged 3D-SICs depends on both the yield and the applied test. For example, test flow t3a contains a pre-bond test and a final test. The yield of the dies that enter the stack equals $Y(n,k)$ after applying wafer matching and thus some of the faulty stacks are known at this time. Since no other tests are performed only the fraction of 3D-SICs that are considered good are packaged (i.e., $Y(n,k)$). If for example also an interconnect test is performed in the post-bond test (test flow t4a), then 3D-SICs with faulty interconnects can also be detected and prevented from being packaged, leading to a packaging ratio of $Y(n,k) \cdot Y_{INT}^{(n-1)}$.

2012 International Conference on Design & Technology of Integrated Systems in Nanoscale Era

TABLE IV
YIELD WAFER MATCHING VS RANDOM W2W STACKING

wafer matching	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
no	66.67	54.43	44.45	36.29	29.63
yes	67.61	56.25	47.12	39.68	33.54

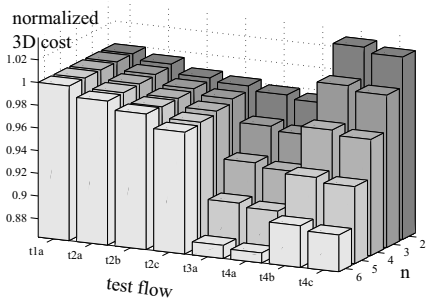


Fig. 4. Normalized 3D cost versus stack size.

B. Experiments

In this subsection, we describe the experiments performed using the test flows of Table I and there cost calculation calculation of Section IV. The parameters considered so far are the default values for each experiment. In addition, the following experiments have been conducted:

1) **Impact of stack size:** In this experiment, the impact of different test flows will be investigated while considering different stack sizes $2 \leq n \leq 6$. Table IV shows the stack yield belonging to this stack size with and without wafer matching [6]. These yields do not include the stacked-die and interconnect yield.
2) **Impact of die yield:** A similar experiment as the previous one, but now by having a fixed stack size of n=2, and variable die yield Y_D: 60%$\leq Y_D \leq$90%
3) **Impact of stacking yield:** In this case, the default process parameters are used (e.g., n=2, Y_D=81.65%, etc.), but the stacking yield is varied; this yield consists of interconnect yield Y_{INT} and stacked-die yield Y_{SD}: 91%$\leq Y_{INT}, Y_{SD} \leq$99%
4) **Impact of packaging cost:** To simulate a different packaging cost we consider $0.2 \leq \beta \leq 0.8$, while fixing all the other parameters to their default values.

The results of the experiments are described in the next section.

VI. SIMULATION RESULTS

In this section, the simulation results are presented. The impact of different test flows are analyzed for each experiment.

A. Impact of stack size

Figure 4 depicts the relative overall 3D-SIC cost of the test flows for a stack size between $2 \leq n \leq 6$. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1 for each stack size. The following conclusions can be drawn from the figure:

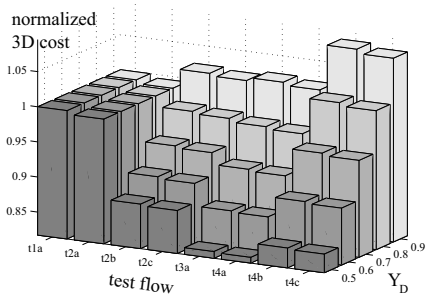


Fig. 5. Normalized 3D cost versus die yield.

- Test flows with pre-bond tests (e.g., t3a and t4a) can reduce the overall cost. The larger n, the larger this reduction.
- Test flow t4a is the most cost-effective test flow irrespective of n.
- Test flows t2a, t2b, t2c have a marginal impact on the cost reduction irrespective of n. The difference with the remaining test flows is due to the yield increase achieved by wafer matching (see Table IV).
- Re-testing dies in the post-bond phase for t4b and t4c only adds to the cost when compared to t4a. Due to a high stacking yield, re-testing of dies is not beneficial.

B. Impact of die yield

Figure 5 depicts the relative 3D cost of the test flows with a die yield varying between 50%$\leq Y_D \leq$90% for the default parameters. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1. From the figure we conclude the following.

- Test flows with pre-bond tests significantly reduce the overall cost for die yields lower than 90%. The lower the die yield the larger the reduction (except for t2a since this test flow does not test for dies during the pre-bond and post-bond phase).
- Test flow t2a has a marginal impact on the cost, irrespective of the die yield. This is not the case for t2b and t2c, as they both test for dies in the post-bond phase. The lower the die yield, the more faulty ICs are detected prior to packaging.
- Similar conclusions can be drawn as those from Figure 4 for the test flows enabled with pre-bond testing. Applying a pre-bond test (thus wafer matching), and testing only for the interconnects during the post-bond phase results in most cases into the overall lowest cost.

C. Impact of stacking yield

The stacking yield consists of the interconnect yield and stacked die yield. Due to space shortage only the results of the stacked die yield experiment are shown.

Figure 6 depicts this experiment and shows the overall 3D cost versus stacked die yield for the test flows. The 3D cost of the flows are normalized to the cost of TF1 for each stacking yield.

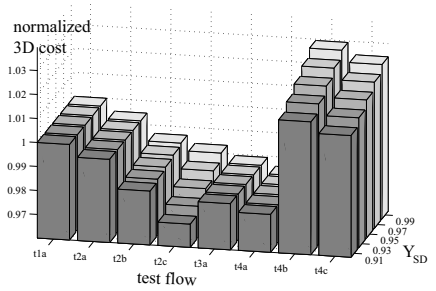


Fig. 6. Normalized 3D cost versus stacked die yield.

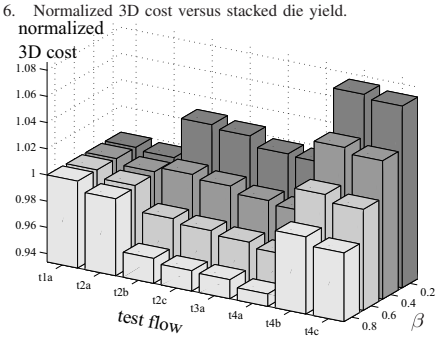


Fig. 7. Normalized 3D cost versus packaging cost.

From the figure we conclude that t2c and t4a are the most cost-effective test flows. If Y_{SD} is very high (i.e., 99%), then t4a perfroms best. However, when Y_{SD} reduces t2c becomes most cost efficient. In this case, the benefit of wafer matching reduces due to a larger number of stack faults.

D. Impact of packaging cost

Figure 7 depicts the relative 3D cost of the test flows for $0.2 \leq \beta \leq 0.8$. From the figure we conclude the following.

- For very low packaging costs (β=0.2), test flows that test dies in the pre-bond phase (t3a, t4a, t4b and t4c) and and mid-bond phase (t2b, t2c, t4b and t4) negatively impact the 3D cost. The cost of testing the dies is not preventing enough faulty 3D-SICs to be packaged. For a low packaging cost it is more advantageous to skip die tests in the pre-bond and post-bond phase.
- For increasing β, test flow t4a becomes the most efficient one. This test flow significantly reduce the overall cost for $\beta \geq 0.8$.
- Test flow t2a again has a marginal impact on the cost, irrespective of the packaging cost. The high interconnect yield and the testing of interconnects only during the post-bond test impacts the 3D cost minimally.

In order to optimize the overall test cost, the appropriate test flow should be selected to reduce the 3D-SIC cost. Therefore, cost modeling is very important.

VII. Conclusion

This paper investigated the impact of several 3D test flows on the total 3D cost in W2W stacking. It introduced a framework of test flows for 3D-SIC testing; each test flow is based on a combination of tests applied at three test moments, i.e., the pre-bond wafer test, the post-bond test and the final test. A model that considers manufacturing, test and packaging cost is presented in order to evaluate the impact of different test flows on the overall cost.

The simulation results showed that the pre-bond testings is extremely important in order to reduce overall cost. The benefit of having pre-bond tests is a yield increase due to wafer matching. In most of the cases, this approved to be beneficial. In the post-bond test phase, primarily interconnect test are of relevance. In some cases, also die tests proved to be cost effective. The final test phase included both tests for interconnects and dies.

The conclusion of the paper indicates that in order to manufacture 3D-ICs at optimum cost for W2W stacking, any DFT has to consider not only the infrastructure for pre-bond tests, but also take into consideration to test for interconnects during the post-bond phase.

References

[1] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Desig Test on Computers*, vol. 22, no. 6, pp. 498-510, 2005.
[2] P. Garrou, Christopher Bower and Pater Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.
[3] S. Reda, G. Smith and L. Smith, "Maximizing the Functional Yield of Wafer-to-Wafer 3-D Integration", IEEE Transactions on Very Large Scale Integration Systems, Vol 17, Issue 9, pp. 1357-1362, 2010.
[4] E. Singh, "Exploiting rotational symmetries for improved stacked yields in W2W 3D-SICs", IEEE VLSI Test Symposium, pp. 32-37, 2011.
[5] J. Verbree, E.J. Marinissen, P. Roussel and D. Velenis, "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking", IEEE European Test Symposium, pp. 36-41, May 2010.
[6] M. Taouil, S. Hamdioui, J. Verbree and E.J. Marinissen, On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs, IEEE International Test Conference, pp. 1-10, 2010.
[7] A.J. Walker, "A Manufacturing Cost Model for 3-D Monolithic Memory Integrated Circuits", *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, no. 2, pp. 268-275, 2009.
[8] P. Mercier, S.R. Singh, K. Iniewski, B. Moore and P. O'Shea, "Yield and Cost Modeling for 3D Chip Stack Technologies", *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 357-360, 2006.
[9] Y. Chen et al. "Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 471-476, 2010.
[10] M. Taouil, S. Hamdioui, K.Beenakker and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking ", *Asian Test Symposium*, pp. 435-441, 2010.
[11] M. Taouil. S. Hamdioui and E.J. Marinissen, "Test Cost Modeling for 3D-Stacked ICs", *IEEE International Workshop on Testing 3D stacked IC*, 2011.
[12] Erik Jan Marinissen and Yervant Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference, 2009*, Nov. 2009.
[13] P. Chen, C. Wu and D. Kwai, "On-Chip TSV testing for 3D IC before bonding using sense amplification", *Asian Test Symposium (ATS)*, pp. 450-455, 2009.
[14] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computer*, vol 25, no. 5, pp. 26-35, Oct. 2009.
[15] Sematech Wafer Cost Comparison Calculator, http://ismi.sematech.org/modeling/agreements/wafercalc.htm
[16] J. Chappell, What costs most in 300mm? As materials management becomes more complex, FOUP becomes first line of defense, http://findarticles.com/p/articles/mi_m0EKF/ is_24_48/ai_87145967/
[17] P. Siblerud, Emc-3d consortium develops process and cost model for interconnect thru-silicon-via or (iTSV^{TM}) structures", 2008. http://emc3d.org/documents/pressReleases/2008/EMC3D_iTSV_CoO_PressRelease_final_Sept4_2008.pdf
[18] D. Velenis et al., Impact of 3D design choices on manufacturing cost, IEEE International Conference on 3D System Integration, pp. 1-5, Sept 2009.
[19] D. K. de Vries, Investigation of Gross Die Per Wafer Formulas, IEEE Transactions on Semiconductor Manufacturing, vol. 18, no. 1, pp. 136-139, 2005.
[20] M. Bushnell and V. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, Frontiers in Electronic Testing, Vol. 17 Springer, 2000.
[21] R. Tummala, "Fundamentals of Microsystems Packaging", *McGraw-Hill Professional*, 2008.

2010 19th IEEE Asian Test Symposium

Test Cost Analysis for 3D Die-to-Wafer Stacking

Mottaqiallah Taouil[1] Said Hamdioui[1] Kees Beenakker[2] Erik Jan Marinissen[3]

[1]Computer Engineering Lab [2]DIMES Technology Center
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, S.Hamdioui, C.I.M.Beenakker}@tudelft.nl

[3]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
erik.jan.marinissen@imec.be

Abstract

The industry is preparing itself for three-dimensional stacked ICs (3D-SICs); a technology that promises heterogeneous integration with higher performance and lower power dissipation at a smaller footprint. Several 3D stacking approaches are under development. From a yield point of view, Die-to-Wafer (D2W) stacking seems the most favorable approach, due to the ability of Known Good Die stacking. Minimizing the test cost for such a stacking approach is a challenging task. Every manufactured chip has to be tested, and any tiny test saving per 3D-SIC impacts the overall cost, especially in high-volume production. This paper establishes a cost model for D2W SICs and investigates the impact of the test cost for different test flows. It first introduces a framework covering different test flows for 3D D2W ICs. Subsequently, it proposes a test cost model to estimate the impact of the test flow on the overall 3D-SIC cost. Our simulation results show that (a) test flows with pre-bond testing significantly reduce the overall cost, (b) a cheaper test flow does not necessary result in lower overall cost, (c) test flows with intermediate tests (performed during the stacking process) pay off, (d) the most cost-effective test flow consists of pre-bond tests and strongly depends on the stack yield; hence, adapting the test according the stack yield is the best approach to use.

Keywords: *3D test flow, 3D test cost, Die-to-Wafer stacking, 3D manufacturing cost, Through-Silicon-Via.*

I. Introduction

The popularity of 3D Stacked ICs (3D-SICs) is rising among industry and research institutes [1–8]. 3D-SICs are emerging as one of the main contesters to continue the trend of Moore's Law. Currently, a number of methods have been proposed to implement the interconnection of stacked dies [1]. One of the most promising and perhaps the most reliable way to achieve this is with *Through-Silicon Vias (TSVs)*. TSVs are holes going through the chip silicon substrate filled with a conducting material. They enable short interconnections in 3D-SICs.

The prospects of the research [1–8] show many 3D-SICs benefits compared to planar dies [2], and include (a) improved performance due to short TSVs that connect IPs on different layers, (b) heterogeneous integration (for example, DRAM memory can be manufactured in separate layers), and (c) a better form factor and package volume density due to vertical stacking.

3D-SICs with TSVs can be manufactured using three different stacking approaches: Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) or Die-to-Die (D2D) stacking [2]. In W2W, complete wafers are stacked and bonded together. The major benefit of W2W is the high manufacturing throughput and the ability to handle small dies. In D2D, a high yield can be obtained due to Known Good Die (KGD) stacking [2], but the throughput is expected to be less. The manufacturing throughput in D2W settles between D2D and W2W, and results in similar yields as in D2D due to the same ability of KGD stacking. This paper focuses on D2W stacking as it is the most relevant stacking approach in industry.

To guarantee high 3D-SIC product quality at lower cost, appropriate test flows need to be developed. For example, in D2W stacking dies may not only require testing before they are stacked (i.e., pre-bond testing), but also during and after stacking (post-bond test). The question arises, whether it is justifiable to perform a pre-bond test as well as a post-bond test after each created temporary stack; i.e. are the dies still functionally operating and are the TSVs created properly. Sources of die failures during stacking could be introduced by thinning, bonding and TSV failures including misalignment and opens [9]. If it is known beforehand that a particular stack is corrupted, silicon, TSV and stacking costs can be prevented for the successive die that has to be stacked in D2W stacking. This paper investigates the impact of different test flows on the overall

1081-7735/10 $26.00 © 2010 IEEE
DOI 10.1109/ATS.2010.80

3D cost in D2W stacking. The emphasis is on the impact of the different test flows, rather than on the analysis of the impact of different manufacturing processes. The contributions of this paper are the following.

- A new framework that covers different test flows.
- A cost model for 3D D2W-stacked 3D-SICs.
- An investigation of the impact of different 3D D2W test flows on the overall 3D cost.

The remainder of the paper is organized as follows. Section II presents the test flow framework. Section III introduces the cost model used for the evaluation of the various test flows. Section IV discusses our simulation results; it first describes the parameters of the experiments, and thereafter presents the experimental results. Section V concludes the paper.

II. Test Flow Framework

This section first defines a test flow for 3D-SICs by extending the 2D test flow. Thereafter, it provides a framework for 3D test flows.

A. 2D versus 3D Test Flow

A conventional 2D test flow for planar wafers is depicted in Figure 1(a) [10]. Here, usually two *test moments* are applicable; i.e., a wafer test prior to packaging and the final test after packaging. The wafer test can be cost-effective when the yield is low, since it prevents unnecessary assembly and packaging costs. The goal of the final test is to guarantee the required quality of the final packaged chip. For 3D-SICs, four test moments can be distinguished in time as depicted in Figure 1(b). We categorize all different test moments in these four test phases, as given next:

1) T_{pb}: n *pre-bond* wafer tests for each individual die on the wafer (n is the number of stacked layers). T_{pb} tests prevent faulty dies from being stacked. Besides testing for dies, TSVs (in case of via-first [2]) can be tested for as well. Although the bonding is not performed yet, capacitance tests can detect some faulty TSVs [11].
2) T_{in}: n-2 *intermediate* tests applicable during the intermediate stacking and bonding. In this case, either the dies, the interconnects, their combination or none of them can be tested for. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g, die thinning, or bonding [9]. In the simulation model of our test flows, first the interconnects are tested and thereafter the dies in bottom up order (in case both are tested for); if a fault is detected in the interconnects, then there is no need to test the dies as the SIC will be faulty anyway. The reason for this particular order is that the test cost for interconnects is considered cheaper, as will be explained in Section III.
3) T_{pr}: one *pre-packaging* test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be seen as a way to prevent unnecessary assembly and packaging cost.
4) T_{po}: one final *post-packaging* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied here as well.

Note that in total there are $2 \cdot n$ different test moments.

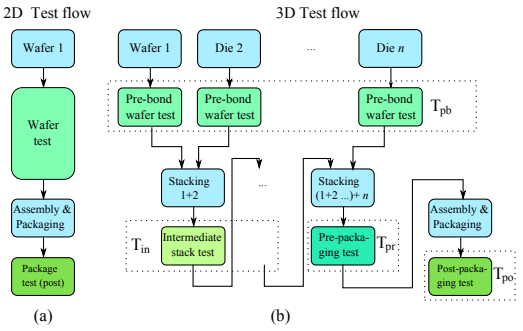


Fig. 1. 2D versus 3D D2W test flows.

Depending on either one or more companies are involved in the manufacturing of 3D-SICS, different requirements can be set for the pre-bond wafer test quality [12]. If the wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place, the pre-bond test quality is subject to optimization. This mean, there is not only the option to perform pre-bond testing or not, but also to perform pre-bond testing at a higher or lower test quality. Faulty undetected dies can be detected in a later stadium, e.g., in higher quality post-packaging tests. Similarly, a high quality pre-packaging test (Known Good Stacks test) can be applied.

B. 3D Test Flow Framework

The test flow framework for 3D D2W stacking can be extracted from the test flow moments depicted in Figure 1(b). Depending on whether no or at least one test is performed at each possible test moment, we can distinguish 2^{2n} possible test flows out of $2n$ test moments. This number will further increase if we consider that tests

TABLE I. Test flow framework

T_{pb}	T_{in}	T_{pr}			
		$d_t i_t$	$d_t i_a$	$d_a i_t$	$d_a i_a$
n	n	–	–	–	TF1
n	i_t	–	–	TF2	–
n	d_t	–	TF3	–	–
n	$d_t i_t$	TF4	–	–	–
y	n	–	–	–	TF5
y	i_t	–	–	TF6	–
y	d_t	–	TF7	–	–
y	$d_t i_t$	TF8	–	–	–

"–" denotes non-applicable

of each phase may target different faults; e.g., if we assume that T_{in} may test (1) one or more dies, (2) one or more interconnects, (3) a combination of (1) and (2), or (4) none, then the number of possibilities for T_{in} will be 4^{n-2}. This increases the number of test flows from 2^{2n} to 2^n (T_{pb}) $\times 4^{n-2}$ (T_{in}) $\times 2$ (T_{pr}) $\times 2$ (T_{po}) $= 2^{3n-2}$. It is clear that considering all 'theoretical' possible test flows will result in an unmanageable space. Therefore realistic assumptions have to be made in order to create a clear overview (without loss of generality) for the work presented in this paper. Our assumptions consist of the following.

1) During stacking, it is assumed that only the *top* two dies could get corrupted since these dies are most susceptible to the stacking/bonding steps like heating, thinning, pressure, and TSV-related defects.
2) Each test flow has to guarantee that a 3D-SIC is fault free before it is packaged to prevent unnecessary costs. The test phases 'T_{pb}+T_{in}+T_{pr}' test each die and each interconnect of the SIC at least once.
3) For the T_{in} test phase, the *same* test content is assumed to be applied among all $n-2$ test moments.
4) The tests performed during T_{po} are assumed to be the same for all test flows.

Because of Assumption 1, T_{in} will test only for one of the following:

- Only for the *top dies* (d_t= dies top)
- Only for the *interconnect* between the top dies (i_t= interconnect top).
- For both the *top dies* and *top interconnects* ($d_t i_t$).
- none (n)

This results into $T_{in} \in \{d_t, i_t, d_t i_t, n\}$.

Table I contains the test flow framework of all possible test flows based on the above assumptions. The first column denotes the two possibilities for T_{pr} (pre-bond test), either it is performed ('y') or not ('n'). The second column gives the four possible values of $T_{in} \in \{d_t, i_t, d_t i_t, n\}$. The second row of the rest of the columns list the different possible values of T_{pr} required in combination with T_{pb} and T_{in} to satisfy Assumption 2; these are:

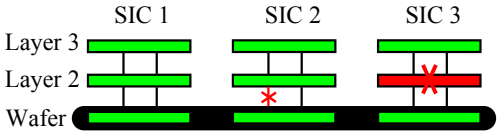


Fig. 2. Faults during stacking

- $d_t i_t$: test for both top dies and top interconnects.
- $d_t i_a$: test for top dies and *all* interconnects.
- $d_a i_t$: test for *all* dies and top interconnects.
- $d_a i_a$: test for *all* dies and *all* interconnects.

Each possible test flow is given a name in the table; e.g., TF1 denotes a test flow based on no T_{bp}, no T_{in} and $T_{pr} = d_a i_a$. There are in total eight test flows, i.e., TF1 to TF8. The entries with '–' denote non-applicable combinations, as they do not satisfy Assumption 2 or more tests are applied than required by Assumption 1.

The framework of test flows clearly indicates that an appropriate 3D DfT test architecture has to support independent testing of dies and interconnects, both for intermediate and final stacks. In [13], an architecture providing these functionalities is proposed.

In order to provide more insight into the different test flows and their impact on the total cost of 3D-SICs, we consider the example shown in Figure 2. It consists of three SICs with n=3 layers each. For simplicity, it is assumed that all dies in the pre-bond phase were manufactured with 100% yield and that two faults occurred during stacking of Layer 2 on the bottom layer, one in SIC2 and the other one in SIC3. In SIC2, a fault occurred in the interconnects between the bottom die (i.e, Layer 1) and the die at Layer 2 (e.g., due to TSV failures), while in SIC3 a fault occurred in Layer 2 (e.g., because of thinning). It is assumed that during the intermediate and pre-packaging tests, first interconnects are tested, followed by the dies in bottom up order.

Table II shows the impact of four test flows TF1, TF2, TF3 and TF4 on three different cost factors: manufacturing, test, and packaging. Each entry in the table is composed out of three numbers, associated with SIC1, SIC2 and SIC3 respectively, followed by their sum. The costs are explained next.

The manufacturing cost is considered to include the number of used dies (the second column of the table) and the number of stacking operations performed (the third column of the table). For example, in TF1 only T_{pr}=$d_a i_t$ is performed (see Table I); therefore this will result in: (a) stacking of three dies per 3D-SIC, hence 3+3+3=9 dies, and (b) two stacking operations per SIC, thus a total of 2+2+2=6.

TABLE II. Impact of Test Flows

TF	Manufacturing cost		Test cost					Packaging cost
	#dies	#stacking operations	T_{pb}	T_{in}		T_{pr}		#packaged SICs
			#dies	#inter	#dies	#inter	#dies	
TF1	3+3+3=9	2+2+2=6	0+0+0=0	0+0+0=0	0+0+0=0	2+1+2=5	3+0+2=5	1+0+0=1
TF2	3+2+3=8	2+1+2=5	0+0+0=0	1+1+1=3	0+0+0=0	1+0+1=2	3+0+2=5	1+0+0=1
TF3	3+3+2=8	2+2+1=5	0+0+0=0	0+0+0=0	2+2+2=6	2+1+0=3	2+2+0=3	1+0+0=1
TF4	3+2+2=7	2+1+1=4	0+0+0=0	1+1+1=3	2+0+2=4	1+0+0=1	2+0+0=2	1+0+0=1

The test cost is categorized according to the test phases defined in Section II-A; i.e., pre-bond wafer tests T_{pb}, intermediate tests T_{in}, pre-packaging tests T_{pr} and post-packaging tests T_{po}. Note that T_{po} is not included in the table as we assumed that post-packaging tests are the same for all test flows (Assumption 4). Except for the T_{pb} phase, each test phase distinguishes between tests for interconnects and tests for dies. Consider test flow TF4, which performs the following tests (see also Table I):

- No pre-bond test (i.e., T_{pb}=n): no tests are executed and therefore no pre-bond tests for the three SICs are performed.
- Intermediate tests consisting of (a) tests for top dies and (b) tests for top interconnects (i.e., T_{in}=$d_t i_t$). Note that there is $n-2=1$ test moment. Hence, in this phase TF4 tests for the interconnects between the bottom layer and Layer 2 of each SIC, resulting in $1+1+1=3$ tests. In addition, TF4 tests for two bottom dies of SIC1 (i.e., the first two layers), no die from SIC2 (since the interconnect found to be faulty during i_t tests) and two bottom dies of SIC3 resulting into $2+0+2=4$ tests.
- Pre-package tests consisting of testing top dies and top interconnects of the SIC (T_{pr}=$d_t i_t$). In this phase, TF4 tests only for the top interconnects and the two top dies of SIC1, not those of SIC2 and SIC3 as they are already considered faulty after the intermediate tests were applied. This results into a total test of one interconnect and two dies during this phase.

The packaging cost is given in the last column of Table II. Because of Assumption 2, the packaging cost are the same for all the four test flows. Only SIC1 will be packaged, while the other two SICs will be discarded.

III. 3D Cost Model

To evaluate the impact of the different test flows on the overall 3D-SIC cost, an appropriate cost model is built. Figure 3 shows the block diagram of the cost model; it considers three major inputs:

- Manufacturing cost: It includes wafer cost, costs required for wafer processing, TSVs and 3D stacking.
- Test cost: The cost related to testing of dies and interconnects. Test flows have a large impact on this cost since they determine when and what to test for.

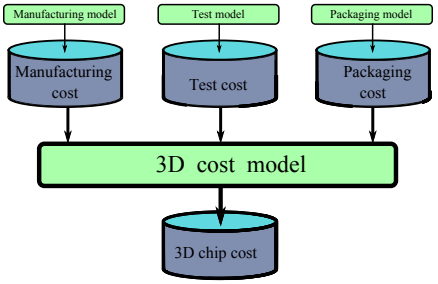


Fig. 3. Test cost model 3D D2W Stacking.

- Packaging cost: The cost to package stacked 3D-SICs.

The cost model calculates the overall 3D cost per test flow. In addition, it also determines the share of the test cost as compared to the overall cost. In fact, the model performs more elaborated and comprehensive calculations of those explained in the example of Section II-B (shown in Figure 2 and Table II). The model collects statistical data (in our case based on 1000 wafers) while considering the different costs. The monitored data includes e.g., the number of used dies, the number of stacking/bonding operations, the number of packaged SICs, the number of tests performed (for dies and interconnect), etc.

Since the purpose of this work is to investigate the impact of different test flows rather than to observe the impact of different manufacturing processes (e.g., transistor feature size, TSV via-first or via-last, Face-to-Face or Back-to-Face bonding orientation, the number of TSVs etc.), the manufacturing costs are assumed to be constant, as discussed in Section IV. However, the test cost strongly depends on other parameters like die yield, interconnect yield, stacking yield, number of stacked layers, etc. Section IV provides more details about our experiment.

In the rest of this section, more details about the three major inputs of the cost model are given.

a) Manufacturing Cost: It includes wafer cost, costs required for wafer processing, TSV fabrication and 3D stacking/bonding. For wafers and their processing, we used the cost models of [14] and [15]; the total price of a 300 mm wafer is estimated at approximately $2779.

The model in [14] considers a variety of costs, including installation, maintenance, lithography and material. For TSV fabrication, the work of EMC-3D consortium [16] is used; the cost to fabricate 5 μm TSVs in a single wafer is assumed to be \$190 and these cost are additive to the wafer cost. To estimate the cost of the 3D stacking/bonding process, the 3D cost model in [17,18] is used.

b) Test Cost: This cost is related to testing of dies and interconnects. To estimate the test cost per die, the model in [19] is used; it includes depreciation, maintenance and operating cost and assumes five ATE machines operating simultaneously. The derived test cost equals 3.82 \$cent/second per die. Assuming a test time of 6 seconds per die, the test cost will be \$0.23 per die. To estimate the interconnect test cost, a ratio of 1:100 between the test time of dies and interconnects is assumed (as in [20]).

c) Packaging Cost: The packaging cost for 3D SICs used in our model is based on oral conversations with Boschman BV [21] and DIMES [22]. The costs are comprehensive and include machine, maintenance, labor and material cost.

IV. Case Study

In this section, the test flows of Table I are analyzed and evaluated based on the cost model of Figure 3. Section IV-A defines the parameters considered in our experiments, while Section IV-B presents the results.

A. Model Parameters

The impact of the test cost on the overall 3D cost depends on several parameters, e.g., stack size, die yield, number of dies per wafer, stacking yield, interconnect yield, packaging yield, fault coverage, etc. Due to space limitations, in this paper we restrict ourselves to the impact of three main parameters (i.e., stack size n, die yield Y_D and stack yield Y_S) on the test and overall cost. Note that for each experiment, only one parameter is considered to be variable, while the others are set to fixed values. These fixed values are derived from our reference process, which is described next.

In our reference process, the die yield is based on the stacking process in [20], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. The work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50\ mm^2$ and a 300 mm wafer, the number of Gross Dies per Wafer (GDW) can be estimated to 1278 [23]. With the negative binomial formula for yield, a die yield of $Y_D = (1+\frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [19]. For the stack size we assume a default stack size n=5.

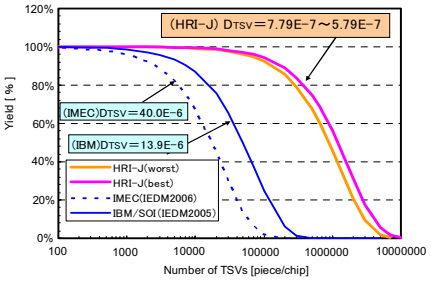


Fig. 4. TSV yield based on a Poisson Distribution [24].

The stacking yield is considered to be composed out of two parameters: the TSV interconnect yield Y_{TSV} and the stacked-die yield Y_{SD}. Figure 4 is used to estimate Y_{TSV} [24]. It shows the TSV yield decrease as a function of the number of TSVs per chip for three manufacturers. In our simulations, the TSV yield Y_{TSV} is assumed to be 95%. Dies that enter the stack could get corrupted during stacking. This is modeled by the stacked-die yield Y_{SD} and is assumed to be 95% as well. Several research works assume a complete stack yield of approximately 95% [20,25].

As already mentioned, three main parameters are considered to be variable in our experiment; these are:

1) **Stack size.** The stack size is considered to vary between $2 \leq n \leq 6$.
2) **Die yield.** The die yield assumes to take values between $60\% \leq Y_D \leq 90\%$.
3) **Stacking yield.** Here, we assume both the Y_{TSV} and Y_{SD} to take values of 93% and 99%.

B. Results

In this section, the simulation results are presented. First, the impact of the test flows on the overall 3D D2W cost is covered, followed by the impact of the test flows on the share of test cost.

1) Impact of Test Flows on Overall Cost: To evaluate the impact on the overall cost, the simulation is performed three times: (1) by varying the stack size while keeping the wafer and stack yield constant, (2) by varying the die yield while keeping the stack size and stack yield constant, and (3) by varying the stack yield while keeping the stack size and die yield constant.

a) Varying the stack size: Figure 5 depicts the relative 3D cost of the test flows for $2 \leq n \leq 6$. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1 for each stack size. For n=2, test flows TF1, TF2, TF3 and TF4 result in equal cost; the same thing applies to test flows TF5, TF6, TF7 and TF8. The reason is that

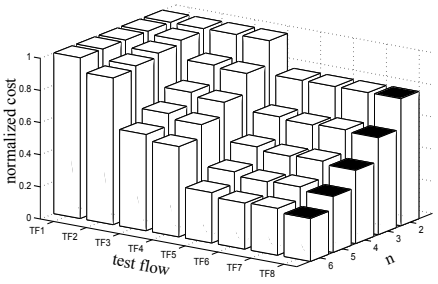


Fig. 5. Normalized test cost for the test flows by considering different stack sizes.

in this case, the test flows are the same. The following conclusions can be drawn from the figure:

- Test flows with pre-bond tests significantly reduce the overall cost. The larger n, the larger the reduction.
- TF8 is the most cost-effective test flow irrespective of n. The bars with black tops represent the test flows with the lowest costs per layer.
- TF2 has a marginal impact on the cost reduction irrespective of n. This is because TF2 neither performs pre-bond tests nor die tests during the intermediate phase. This is not the case for TF3 and TF4, as they both test for dies in the intermediate phase.
- While test flow TF2 results in higher cost than test flow TF3, the reverse occurs for the test flows TF6 and TF7. Note that TF1 and TF3 are similar to TF6 and TF7, respectively, except that the TF6 and TF7 also include pre-bond testing. In case of TF6 and TF7 only good dies will be stacked. Hence, it is cost-wise cheaper to test the interconnects (TF6) than to re-test the dies (TF7) during the intermediate phase. Nevertheless, testing both interconnects and dies during the intermediate phase is the most cost-effective test flow (i.e., TF8).

b) Varying the die yield: Figure 6 depicts the relative 3D cost of the test flows with a die yield varying between $60\% \leq Y_D \leq 90\%$. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1. The stack size is fixed to n=5 and the interconnect and stacked-die yield are set both to 95%. From the figure we conclude the following.

- Test flows with pre-bond tests significantly reduce the overall cost. The lower the die yield the larger the reduction (except for TF2 since this test flow does not test for dies during the pre-bond and intermediate phases).
- TF2 has a marginal impact on the cost, irrespective of the die yield. This is not the case for TF3 and TF4, as they both test for dies in the intermediate phase.
- Similar conclusions can be drawn as those from

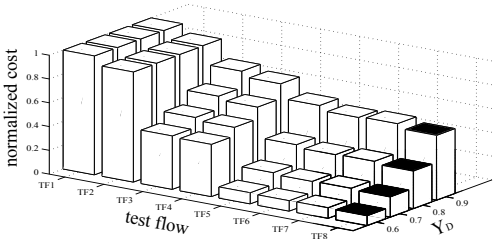


Fig. 6. Normalized test cost for the test flows by considering variable die yield.

Figure 5 for the test flows enabled with pre-bond testing. It is cheaper to test for interconnects only (TF6) than to test for dies only (TF7) during the intermediate test phase. Nevertheless, testing both for interconnects and dies during the intermediate phase is the most cost-effective test flow (i.e., TF8).

c) Varying the stack yield: Figure 7 depicts the overall 3D cost versus stacked yield (i.e., interconnect Y_{TSV} and stacked-die Y_{SD}) for the test flows. In the figure, Y_{TSV} and Y_{SD} are set both to 93% and 99%. The 3D cost of the flows are normalized to the cost of TF1 where Y_{TSV}=Y_{SD}=93%. The bars with black tops presents test flows with the least impact on the overall cost per stacking yield. For example, for a stack yield of [Y_{TSV}, Y_{SD}] = [0.99, 0.99], TF6 is the most cost-effective test flow.

From the figure we conclude that TF6 and TF8 are the most cost-effective test flows. If Y_{SD} is very high (i.e., 99%), then TF6 is the best as it tests only for interconnect. However, in case Y_{SD}=93%, TF8 performs better, since it tests for dies during the intermediate phase. Therefore, it is able to prevent unnecessary stacking of dies in faulty partial stacks.

2) Impact of Test Flows on Test Cost: The relative impact of the test cost on the overall cost is depicted in Figure 8 for the reference process. There are two bars per test flow. The first bar presents the overall cost normalized to TF1, while the second bar presents the ratio of test cost with respect to the overall cost. The figure clearly shows that a cheap test flow does not necessary result in lower overall cost. For example, while TF8 reduces the overall cost with 11% as compared to TF5, the share of test cost of TF8 is 35% higher than that of TF5.

V. Conclusion

This paper investigates the impact of several 3D test flows on the total 3D cost in D2W stacking. It introduces a framework of test flows for 3D testing; each flow is based on a combination of tests applied at four test moments, i.e., the pre-bond wafer test, the intermediate stack test, the

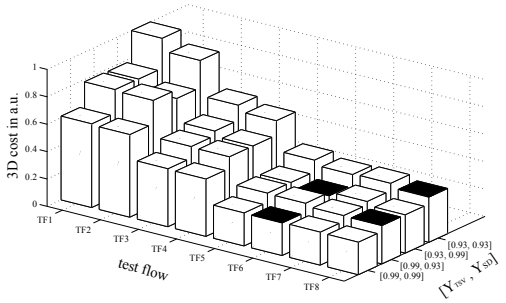


Fig. 7. Normalized test cost for the test flows by considering variable stack yield.

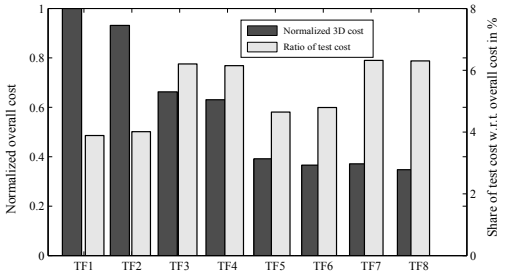


Fig. 8. Test cost versus overall cost.

pre-package test and the post-package test. An appropriate cost model (considering manufacturing, test and packaging cost) is introduced in order to evaluate the impact of different test flows on the overall cost. Different stack sizes, die yield, and stack yield are considered for the evaluation.

The simulation results show that test flows with pre-bond testing significantly reduces the overall cost. Test flows with the intermediate tests enabled with interconnect tests outperform the rest. Moreover, a cheaper test flow does not necessary results in lower overall 3D-SIC cost. The best cost-effective test flow consists of the pre-bond and strongly depends on the stack yield. This requires the adaptation of the test flow during the yield learning of the 3D-SIC process manufacturing. Moreover, test architectures should provide access to all dies as well as all interconnects of the SIC in order to be able to perform intermediate tests.

References

[1] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Design Test on Computers*, vol. 22, no. 6, pp. 498-510, 2005.

[2] P. Garrou, Christopher Bower and Peter Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.

[3] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, vol. 94, no. 6, 2006.

[4] G. Loh et al. "Processor Design in 3D Die-Stacking Technologies", *IEEE Micro*, vol. 27, no. 3, pp. 31-48, 2007.

[5] K. Puttaswamy et al. "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369-1381, 2009.

[6] K. Puttaswamy et al. "Processor Design in 3D Die-Stacking Technologies", *IEEE Transactions on Computers*, vol. 27, no. 3, pp. 31-48, 2007.

[7] Y-F. Tsai et al. "Design Space Exploration for 3-D Cache", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 444-455, 2008.

[8] T. Thorolfsson et al. "Comparative Analysis of Two 3D Integration Implementations of a SAR Processor", *IEEE International Conference on 3D System Integration*, pp. 1-4 Oct. 2009.

[9] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computer*, vol 25, no. 5, pp. 26-35, Oct. 2009.

[10] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference (ITC)*, pp.1-11, 2009.

[11] P. Chen, C. Wu and D. Kwai, "On-Chip TSV Testing for 3D IC before Bonding Using Sense Amplification", *Asian Test Symposium (ATS)*, pp. 450-455, 2009.

[12] E. J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs", *Design, Automation and Test in Europe (DATE)*, pp. 1689-1694, 2010.

[13] E.J. Marinissen, J. Verbree and M. Konijnenburg "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs", *IEEE VLSI test symposium (VTS)*, pp. 269-274, 2010.

[14] "Sematech Wafer Cost Comparison Calculator", http://ismi.sematech.org/modeling/agreements/wafercalc.htm

[15] J. Chappell, "What costs most in 300mm? As materials management becomes more complex, FOUP becomes first line of defense", http://findarticles.com/p/articles/mi_m0EKF/ is_24_48/ai_87145967/

[16] P. Siblerud, "Emc-3d consortium develops process and cost model for interconnect thru-silicon-via or (iTSV^{TM}) structures", 2008. http://emc3d.org/documents/pressReleases/2008/EMC3D_iTSV_CoO_PressRelease_final_Sept4_2008.pdf

[17] X. Dong and Y. Xie, "System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs) ", *Design Automation Conference (ASP-DAC)*, pp. 234-241, 2009.

[18] X. Dong, "Web-Based 3D Cost Analysis Tool", http://www.cse.psu.edu/ xydong/3dcost.html

[19] M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Wiley-VCH, Weinheim, Germany, 2000.

[20] J. Verbree et al. "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking", IEEE European Test Symposium, pp. 269-274, 2010.

[21] Boschman Technology, http://www.boschman.nl/

[22] Delft Institute of Microsystems and Nanoelectronics, DIMES. http://www.dimes.tudelft.nl/

[23] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas.", IEEE Transactions on Semiconductor Manufacturing, vol. 18, no. 1, pp. 136-139, 2005.

[24] N. Miyakawa, "A 3D Prototyping Chip Based on a Wafer-level Stacking Technology", *Design Automation Conference (ASP-DAC)*, pp. 416-420, 2009

[25] E. Beyne, "3D Integration Crossing IC technology, Packaging and Design Barriers", *Semicon West 2008, TechXPOT, Test Assembly & Packaging*, http://www.semiconwest.org/cms/groups/public/documents/web_content/ctr_024376.pdf

How significant will be the test cost share for 3D Die-to-Wafer stacked-ICs?

Mottaqiallah Taouil Said Hamdioui Erik Jan Marinissen

Computer Engineering Lab
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, S.Hamdioui}@tudelft.nl

IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
{erik.jan.marinissen}@imec.be

***Abstract*— Several challenges must be overcome before high volume production of the 3D Stacked-ICs (3D-SIC) can be realized. A key challenge is to guarantee the required product quality at minimal overall cost. Testing, which is an integral part of 3D-IC manufacturing, should be performed in such way that its cost contribution is optimal. This paper investigates the impact of different test moments for pre-bond and post-bond stacks (resulting into different test flows) on the overall cost of die-to-wafer (D2W) 3D-SICs. The investigation is carried out for a wide range of die yields and stack sizes. Moreover, a breakdown of the cost into manufacturing, test and packaging costs offers a more detailed picture of the 3D overall cost. Our simulation results show that overall cost in D2W stacking strongly depends on the selected test flow; test flows with pre-bond and post-bond tests show a higher test cost share, but a significant reduction in the overall 3D-SIC cost. In addition, the cost breakdown for our reference process reveals that the manufacturing cost is most dominant (between 76% and 85%), followed by test (between 13% and 19%). Moreover, the results show that the share of test and packaging decreases as the manufacturing becomes mature and the yield increases, and that both manufacturing and test cost share increases, while the packaging cost share decreases for higher stack sizes.**

Keywords: *3D test flow, 3D test cost, Die-to-Wafer stacking, 3D manufacturing cost, Through-Silicon-Via.*

I. Introduction

Recent enhancements in process development enable the fabrication of three dimensional stacked ICs (3D-SICs) [1]. A 3D-SIC consists of a pile of two or more vertical stacked ICs electrically interconnected by Through Silicon Vias (TSV). TSVs are holes that go through the silicon substrate filled with a conducting material. 3D technology opens up new research directions that could be investigated to continue the trend of performance increase. For example, it can lead to a smaller footprint, higher interconnect density between stacked dies, higher chip transistor density and lower power consumption due to shorter wires as compared to planar ICs, while possibly using heterogeneous dies [1–7].

Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) and Die-to-Die (D2D) bonding [1] are the existing methods that could be employed in order to manufacture 3D-SICs. W2W bonding leads to highest throughput, as dies are processed in parallel at wafer level, and makes the manufacturing of tiny dies feasible [1]; however, it suffers from low compound yield [8,9]. Regarding yield, D2W and D2D are superior, due to the opportunity to apply Known-Good-Die (KGD) testing [1]. D2D bonding suffers from low throughput as the die stacking is based on individual pairs. This paper focuses on D2W stacking as it is currently the most relevant stacking approach in industry.

The manufacturing of 3D-SICs did not reach a mature stage yet and several challenges have to be overcome before it can be realized. One of these challenges is testing and its associated cost. Testing for defects is required in order to satisfy the required product quality. Due to testing, test flows for 3D-SICs have to consider several test moments. The first test moments relates to pre-bond testing and it targets traditional defects that may occur during processing of planar wafers, possibly augmented with preliminary TSV tests. The good tested dies in the *pre-bond* test phase could get corrupted during the stacking. Typical sources of die failures during stacking include the processing steps involved in thinning, bonding, as well as TSV failures such as misalignments and opens [10]. If it is known beforehand that a particular stack is corrupted, silicon, stacking and bonding costs can be prevented for the successive dies that have to be stacked. This requires partial stack or intermediate tests, which form the second test phase. The number of intermediate tests, both for interconnects as well as dies, increases significantly with the stack size. The third test moment consist of a pre-packaging test and can prevent unnecessary loss in assembly and packaging cost. Finally, a post packaging test ensures the final quality of the outgoing product. To guarantee high 3D-SIC product quality at low cost, appropriate test flows need to be developed that take the different test phases (e.g. pre-bond testing, post-bond testing etc.) into consideration.

This paper investigates the impact of *different* test flows on the overall cost for D2W based 3D-SICS by considering different stack sizes and die yields. For a given yield and/or number of stacked layers, an appropriate test flow has to be used to optimize the cost. In addition, this paper analyses the impact of the test cost on the overall 3D-SIC cost and breaks down this cost into test, manufacturing and packaging cost in order to estimate the test share. In this work, a cost model for 3D D2W-stacked ICs presented in our previous work [11] will be used.

The remainder of the paper is organized as follows. Section II introduces the test flow framework. Section III briefly reviews the cost model to be used in this work. Section IV presents the simulation results; it first describes the parameters of the experiments, and thereafter discusses the experimental results. Section V concludes the paper.

978-1-61284-899-0/11 $26.00 © IEEE

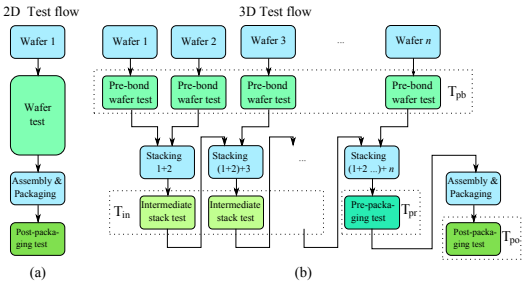


Fig. 1. 2D versus 3D D2W test flows.

II. Test Flow Framework

A test flow consist of a sequence of applied tests at different time steps during the manufacturing of the 3D-SIC. Section II-A defines these test moments and specifies for each one what exactly can be tested for. Subsequently, a framework of various test flows is obtained by applying different tests at different test moments. The framework is presented in Section II-B.

A. 2D versus 3D Test Flow

A conventional 2D test flow for planar wafers is depicted in Figure 1(a) [12]. Here, usually two *test moments* are applicable; i.e., a wafer test prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low, since it prevents unnecessary assembly and packaging costs. The goal of the final test is to guarantee the final quality of the packaged chip. During the manufacturing of a 3D-SIC, additional test points can be defined for each partial created stack. At each test point a distinction can be made between die tests and interconnect tests. Die tests ensure the functionality of individual dies, while interconnect tests ensure functional TSVs. For 3D-SICs, four test moments can be distinguished in time as depicted in Figure 1(b), and explained next.

1) T_{pb}: n *pre-bond* wafer tests, since there are n layers to be stacked. T_{pb} tests prevent faulty dies entering the stack. Besides die test, preliminary TSV interconnect tests can be applied (in case of via-first [1]) as well. An example of a preliminary test that detects some faulty TSVs could be a capacitance test [13].
2) T_{in}: n-2 *intermediate* tests applicable during the intermediate stacking and bonding. In this case, either the dies, the interconnects, their combination or none of them can be tested for. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g., die thinning, and bonding [10]. In the simulation model of our test flows, first the interconnects are tested and thereafter the dies in bottom up order (in case both are tested for); if a fault is detected in the interconnects, then there is no need to test the dies as the SIC will be faulty anyway. The reason for this particular order is that the test cost for interconnects is considered cheaper, as will be explained in Section III.
3) T_{pr}: one *pre-packaging* test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Both dies and interconnects can be tested for.
4) T_{po}: one final *post-packaging* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied here as well.

Note that in total there are $2 \cdot n$ different test moments.

Depending on either one or more companies are involved in the manufacturing of 3D-SICS, different requirements can be set for the pre-bond wafer test quality [14]. If the wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place, the pre-bond test quality is subject to optimization. This means, there is not only the option to perform pre-bond testing or not, but also to perform pre-bond testing at a higher or lower test quality. Faulty undetected dies can be detected in a later stadium, e.g., in higher quality post-packaging tests. Similarly, a high quality pre-packaging test (Known-Good-Stacks test) can be applied.

B. 3D Test Flow Framework

The test flow framework for 3D D2W stacking can be extracted from the test moments depicted in Figure 1(b). Depending on whether no or at least one test is performed at each possible test moment, we can distinguish 2^{2n} possible test flows out of $2n$ test moments. This number will further increase if we consider that tests of each phase may target different faults; e.g., if we assume that T_{in} may test (1) one or more dies, (2) one or more interconnects, (3) a combination of (1) and (2), or (4) none, then the number of possibilities for T_{in} will be 4^{n-2}. This increases the number of test flows from 2^{2n} to 2^n (T_{pb}) $\times 4^{n-2}$ (T_{in}) $\times 2$ (T_{pr}) $\times 2$ (T_{po}) $= 2^{3n-2}$. It is clear that considering all 'theoretical' possible test flows will result in an unmanageable space. Therefore, realistic assumptions have to be made in order to create a clear overview (without loss of generality) for the work presented in this paper. Our assumptions consist of the following.

1) A linear stacking approach is assumed, i.e., dies are stacked sequentially in a bottom-up approach starting from the bottom wafer. During stacking, it is assumed that only the *top* two dies and the interconnect between them could be corrupted; they are assumed to be defect-prone to stacking/bonding steps like heating, thinning, pressure.
2) All die tests are identical; a similar assumption applies to all interconnect tests.
3) Each test flow has to guarantee that a 3D-SIC is fault free before it is packaged to prevent unneces-

TABLE I
TEST FLOW FRAMEWORK

test flow	T_{pb}	T_{in}	T_{pr}
TF1	n	n	$i_a d_a$
TF2	n	i_t	$i_a d_t$
TF3	n	i_t	$i_t d_a$
TF4	n	$i_t d_t$	$i_t d_t$
TF5	y	n	$i_a d_a$
TF6	y	i_t	$i_a d_t$
TF7	y	i_t	$i_t d_a$
TF8	y	$i_t d_t$	$i_t d_t$

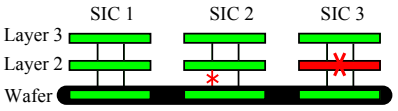


Fig. 2. Examples of defects (x) occurring during stacking.

sary assembly and packaging cost. The test phases 'T_{pb}+T_{in}+T_{pr}' test each die and each interconnect of the SIC at *least* once.

4) The final post-packaging test in T_{po} is a complete test, i.e., all dies and interconnects are tested.

Because of Assumption 1, T_{in} will test only for one of the following:

- Only for the *top dies* (d_t= dies top).
- Only for the *interconnect* between the top dies (i_t= top interconnect).
- For both the *top interconnect* and *top dies* ($i_t d_t$).
- none (n).

This results into $T_{in} \in \{d_t, i_t, i_t d_t, n\}$.

Table I shows the test flow framework of all possible test flows based on the above assumptions. The first column denotes the two possibilities for T_{pr} (pre-bond test), either it is performed ('y') or not ('n'). The second column gives the four possible values of $T_{in} \in \{d_t, i_t, i_t d_t, n\}$. The last column lists the different possible values of T_{pr}. In order to satisfy Assumption 3 (a fault-free 3D-SIC prior to packaging) T_{pr} is limited to the following values:

- $i_t d_t$: test for *top* interconnect and *top* dies.
- $i_t d_a$: test for *top* interconnect and *all* dies.
- $i_a d_t$: test for *all* interconnects and *top* dies.
- $i_a d_a$: test for *all* interconnects and *all* dies.

Each possible test flow is given a name in the table; e.g., TF1 denotes a test flow based on no T_{bp}, no T_{in} and $T_{pr} = i_a d_a$. There are eight test flows in total, i.e., TF1 to TF8.

To provide more insight into the different test flows and their impact on the total 3D-SIC cost, we consider the example shown in Figure 2. It consists of three SICs with n=3 layers each. For simplicity, it is assumed that all dies in the pre-bond phase were manufactured with 100% yield and that two faults occurred during stacking of Layer 2 on the bottom layer, one in SIC2 and one in SIC3. In SIC2, a fault occurred in the interconnects between the bottom die (i.e, Layer 1) and the die at Layer 2 (e.g., due to misaligned TSVs), while in SIC3 a defect occurred in Layer 2 (e.g., due to thinning). It is assumed that during the intermediate and pre-packaging tests, first interconnects are tested, followed by the dies in bottom up order. The framework of test flows clearly indicates that an appropriate 3D DfT test architecture has to support independent testing of dies and interconnects, both for intermediate and final stacks. In [15], an architecture providing these functionalities is proposed.

In order to provide more insight into the different test flows and their impact on the total cost of 3D-SICs, we consider the example shown in Figure 2. It consists of three SICs with n=3 layers each. For simplicity, it is assumed that all dies in the pre-bond phase were manufactured with 100% yield and that two faults occurred during stacking of Layer 2 on the bottom layer, one in SIC2 and the other one in SIC3. In SIC2, a fault occurred in the interconnects between the bottom die (i.e, Layer 1) and the die at Layer 2 (e.g., due to TSV failures), while in SIC3 a fault occurred in Layer 2 (e.g., because of thinning). It is assumed that during the intermediate and pre-packaging tests, first interconnects are tested, followed by the dies in bottom up order.

Table II shows the impact of four test flows TF1, TF2, TF3 and TF4 on three different cost factors: manufacturing, test, and packaging. Each entry in the table is composed out of three numbers, associated with SIC1, SIC2 and SIC3 respectively, followed by their sum. The costs are explained next.

The manufacturing cost is considered to include the number of used dies (the second column of the table) and the number of stacking operations performed (the third column of the table). For example, in TF1 only T_{pr}=$i_a d_a$ is performed (see Table I); therefore this will result in: (a) stacking of three dies per 3D-SIC, hence 3+3+3=9 dies, and (b) two stacking operations per SIC, thus a total of 2+2+2=6.

The test cost is categorized according to the test phases defined in Section II-A; i.e., pre-bond wafer tests T_{pb}, intermediate tests T_{in}, pre-packaging tests T_{pr} and post-packaging tests T_{po}. Note that T_{po} is not included in the table as we assumed that post-packaging tests are the same for all test flows (Assumption 4). Except for the T_{pb} phase, each test phase distinguishes between tests for interconnects and tests for dies. Consider test flow TF4 which performs the following tests (see also Table I):

- No pre-bond test (i.e., T_{pb}=n): no tests are executed and therefore no pre-bond tests for the three SICs are performed.
- Intermediate tests consisting of (a) tests for top dies and (b) tests for top interconnects (i.e., T_{in}=$i_t d_t$). Note that there is $n-2$=1 test moment. Hence, in this phase TF4 tests for the interconnects between the bottom layer and Layer 2 of each SIC, resulting in $1+1+1=3$ tests. In addition, TF4 tests for two bottom dies of SIC1 (i.e., the first two layers), no die from SIC2 (since the interconnect found to be faulty during i_t tests) and two bottom dies of SIC3 resulting into $2+0+2=4$ tests.
- Pre-package tests consisting of testing top dies and

TABLE II
IMPACT OF TEST FLOWS

TF	Manufacturing cost		Test cost					Packaging cost
	#dies	#stacking operations	T_{pb}	T_{in}		T_{pr}		#packaged SICs
			#dies	#inter	#dies	#inter	#dies	
TF1	3+3+3=9	2+2+2=6	0+0+0=0	0+0+0=0	0+0+0=0	2+1+2=5	3+0+2=5	1+0+0=1
TF2	3+2+3=8	2+1+2=5	0+0+0=0	1+1+1=3	0+0+0=0	1+0+1=2	3+0+2=5	1+0+0=1
TF3	3+3+2=8	2+2+1=5	0+0+0=0	0+0+0=0	2+2+2=6	2+1+0=3	2+2+0=3	1+0+0=1
TF4	3+2+2=7	2+1+1=4	0+0+0=0	1+1+1=3	2+0+2=4	1+0+0=1	2+0+0=2	1+0+0=1

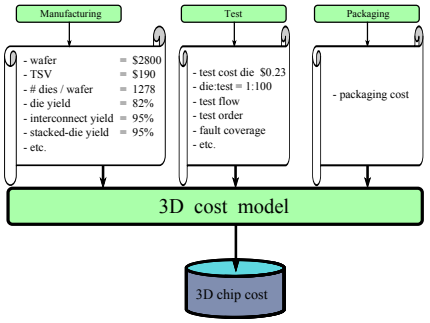


Fig. 3. Test cost model 3D D2W Stacking.

top interconnects of the SIC (T_{pr}=$i_t d_t$). In this phase, TF4 tests only for the top interconnects and the two top dies of SIC1, not those of SIC2 and SIC3 as they are already considered faulty after the intermediate tests were applied. This results into a total test of one interconnect and two dies during this phase.

The packaging cost is given in the last column of Table II. Because of Assumption 2, the packaging costs are the same for all the four test flows. Only SIC1 will be packaged, while the other two SICs will be discarded. Choosing the test flow resulting in optimal overall cost needs the evaluation of all possible test flows using an appropriate generic cost model; the latter is given in the next section.

III. 3D COST MODEL

The cost model for the evaluation of the test flows is explained in more detail in [11]. Figure 3 shows the block diagram of the cost model; it consists of three inputs:

- Manufacturing: It consist of all parameters related to 3D-SIC manufacturing; these are e.g., wafer cost, costs required for wafer processing, TSVs and 3D bonding and thinning.
- Test: This consists of all parameters related to DFT and test such as the cost related to testing dies and interconnects, DFT area overhead etc. Test flows have a large impact on this cost since they determine when and what to test for.
- Packaging: The cost of 3D-SIC packaging.

The value of the parameters related to manufacturing, test and packaging cost used in this work are depicted in Figure 3.

IV. CASE STUDY

In this section, we measure the impact of the test flows in Table I by applying the cost model of Figure 3. We investigate not only the impact of the test flows on the overall cost, but also how it is composed out of test, manufacturing and packaging cost for different die yields and stack sizes. The input parameters required by the model are defined in Section IV-A. The results are presented and discussed in Section IV-B.

A. Simulation Parameters

Several parameters influence the performance of the test flows in terms of cost. These parameters include die yield, stack size, number of dies per wafer, stack yield, packaging yield, fault coverage, etc. The selected parameters for our reference process are described next. The reference process describes the default simulation parameters.

The die yield is based on the stacking process in [8], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. The work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50\ mm^2$ and a 300 mm wafer, the number of Gross Dies per Wafer (GDW) can be estimated to be 1278 [17]. With the negative binomial formula for yield, a die yield of $Y_D = (1+\frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [16]. For the stack size we assume a default stack size n=5. The stacking yield is composed out of two parameters: the interconnect yield Y_{INT} and the stacked-die yield Y_{SD}. In our simulations, the TSV yield Y_{INT} is assumed to be 95%. For the good dies that enter the stack, a small probability exists that they get corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 95% as well. Several research works assume a complete stack yield of approximately 95% [8,18].

B. Results

Our simulation results consist of several experiments. First, the impact of the test flows on the overall 3D D2W cost is examined for the reference process. Thereafter, the composition of this overall cost is split in manufacturing, test and packaging cost in order to determine the share of test cost. The last experiment analyzes the effect of the test flow on both test cost and test cost share by considering different die yields and stack sizes.

Impact on overall cost

Figure 4 shows the overall cost (including test, manufacturing and packaging cost) normalized to TF1 for the 8 test

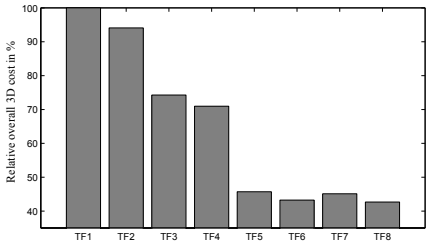


Fig. 4. Normalized 3D cost for n=5.

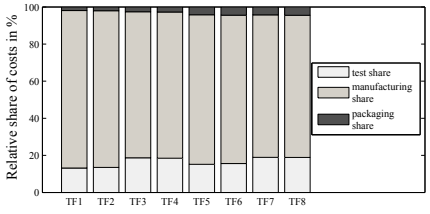


Fig. 5. Share of manufacturing, test and packaging cost in percentage.

flows. For example, TF3 results in an overall cost which is 74.27% of TF1. The figure shows the importance of pre-bond testing, i.e., TF1 up to TF4 are cost wise more expensive than test flows TF5 up to TF8. Since the stack yield is assumed to be much higher than the die yield, test flow TF3 (test for dies during the intermediate phase) results in lower cost than TF2 (test for interconnects only during the intermediate phase). The reverse occurs for the test flows TF6 and TF7. Note that TF1 and TF3 are similar to TF6 and TF7, respectively, except that TF6 and TF7 also include pre-bond testing. In case of TF6 and TF7, only good dies will be stacked. Hence, it is cost-wise cheaper to test the interconnects (TF6) than to re-test the dies (TF7) during the intermediate phase. Nevertheless, testing both interconnects and dies during the intermediate phase is the most cost-effective test flow (i.e., TF8). Test flow TF8 is able to reduce the cost with 57.3% compared to TF1 (that considers only pre-packing tests) and 6.7% as compared to test flow TF5 (that contains pre-bond and pre-packaging tests).

Cost of test share

Figure 5 plots the breakdown of the 3D cost for each test flow, based on the cost model of Figure 3. For each test flow, the shares of test, manufacturing and packaging are depicted. From the figure we conclude the following.

- The manufacturing cost is relatively the most dominant cost factor for each test flow and lies between 76% and 85% of the total cost. However, the overall 3D-SIC cost strongly depends on the selected test flow, as can be concluded from Figure 4.
- The share of test cost is between 13% and 19% depending on the test flow. Test flows containing die tests during intermediate stacking result in relatively higher test cost share as compared with the rest. For instance, test flow TF3, TF4, TF7 and TF8 result in a test cost share of about 19%.
- The share of packaging cost is between 2% and 5%, and it is higher for test flows resulting in lower overall 3D cost. For example, test flow TF1 consist of a packaging cost share of 2%, while this increases to 5% for TF8.
- A higher test cost share does not necessarily result in highest overall cost.

Impact of die yield and stack size

The cost breakdown in manufacturing, test and packaging cost, under different die yields and stack sizes is shown in Figure 6 and 7 respectively. In Figure 6, the impact of different die yields in the range of 30% up to 90% are depicted for the reference process. For each experiment the overall costs are normalized to TF1.

The figure clearly shows that irrespective of the die yield, test flows with pre-bond tests (TF5 to TF8) always result in the lowest overall cost; the latter becomes significant for lower die yield. In addition, the figure reveals that TF8 results into the lowest overall cost in all cases, and that the test cost and packaging cost shares increase as the yield increases. The figure also clarifies the importance of intermediate tests; test flows with intermediate tests result in lower cost. For example, TF8 results in 7% lower overall cost as compared to TF5; note that TF8 and TF5 are the same except that TF8 also consist of intermediate tests.

The figure clearly shows the importance of pre-bond testing, as TF5 to TF8 result in the lowest overall cost is of significant importance, if the die yield is low. The test flow TF8, results in all cases in the lowest overall yield. The test share increases for higher die yields.

Figure 7 shows a similar experiment, but for a variable stack size. The figure indicates again the importance of pre-bond and intermediate testing, especially for larger stack sizes. For n=2, TF1, TF2, TF3 and TF4 result in the same cost, a similar remark holds for the remaining test flows. The figure shows that the share of packaging cost decreases as the stack size increases, while the test share increases with larger stack sizes. For test flow TF8, the test share is 15.4% for a stack size of n=2, while this ratio increases to 20.6% for a stack size of n=6. This increase in test cost for larger stack sizes is due to an increased number of applied tests, while at the same time the number of fault-free 3D-SICs reduces. It is worth noting that although TF8 has the highest test cost share, it results in the lowest overall 3D-cost.

V. Conclusion

This paper investigated the impact of several 3D test flows on the total 3D cost in D2W stacking. It introduced a framework of test flows for 3D testing; each flow is based on a combination of tests applied at four test moments, i.e., the pre-bond wafer test, the intermediate stack test, the pre-package test and the post-package test. A cost model that considers manufacturing, test and packaging cost is presented

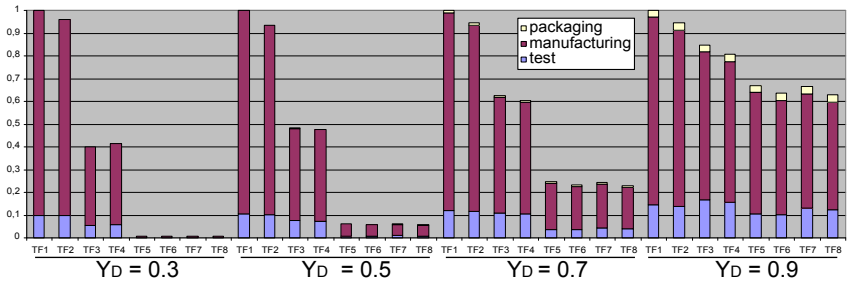


Fig. 6. Cost breakdown for variable die yield.

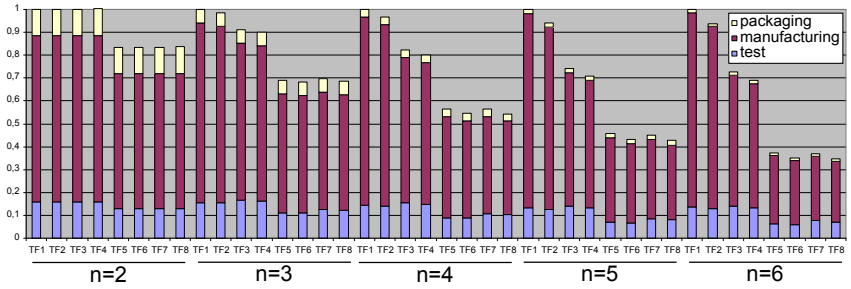


Fig. 7. Cost breakdown for variable stack size.

in order to evaluate the impact of different test flows on the overall cost.

The simulation results showed that the manufacturing cost is the most dominant in 3D stacking and strongly depends on the selected test flow. In addition, they revealed that test flows with pre-bond testing significantly reduced the overall cost. Intermediate tests contributed also to further cost savings. Although the share of test cost increases for such flows, the overall cost is significantly reduced. The cost saving increase with lower die yields and larger stack sizes. The conclusion of the paper indicates that in order to manufacture 3D-ICs at optimum cost, any DFT has to consider not only the infrastructure for pre-bond tests, but also for intermediate tests for both dies and interconnects.

REFERENCES

[1] P. Garrou, Christopher Bower and Peter Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.

[2] 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 on Computers*, vol. 22, no. 6, pp. 498-510, 2005.

[3] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, vol. 94, no. 6, 2006.

[4] G.H. Loh, Y. Xie, and B. Black, "Processor Design in 3D Die-Stacking Technologies", *IEEE Micro*, vol. 27, no. 3, pp. 31-48, 2007.

[5] K. Puttaswamy and G.H. Loh, "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369-1381, 2009.

[6] Y-F. Tsai, F. Wang, Y. Xie; N. Vijaykrishnan and M.J. Irwin, "Design Space Exploration for 3-D Cache", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 444-455, 2008.

[7] T. Thorolfsson, S. Melamed, G. Charles and P.D.Franzon, "Comparative Analysis of Two 3D Integration Implementations of a SAR Processor", *IEEE International Conference on 3D System Integration*, pp. 1-4, 2009.

[8] J. Verbree, E.J. Marinissen, P.Roussel and D. Velenis, "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking", IEEE European Test Symposium, pp. 269-274, 2010.

[9] M. Taouil, S. Hamdioui, J. Verbree and E.J. Marinissen, "On maximizing the compound yield for 3D Wafer-to-Wafer stacked ICs", *IEEE International Test Conference (ITC)*, pp. 1-10, 2010.

[10] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computer*, vol 26, no. 5, pp. 26-35, 2009.

[11] M. Taouil, S. Hamdioui, K. Beenakker and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking", *IEEE Asian Test Symposium*, pp. 435-441, 2010.

[12] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference*, pp.1-11, 2009.

[13] P. Chen, C. Wu and D. Kwai, "On-Chip TSV Testing for 3D IC before Bonding Using Sense Amplification", *Asian Test Symposium* , pp. 450-455, 2009.

[14] E. J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs", *Design, Automation & Test in Europe Conference & Exhibition*, pp. 1689-1694, 2010.

[15] E.J. Marinissen, J. Verbree and M. Konijnenburg "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs", *28th VLSI Test Symposium* , pp. 269-274, 2010.

[16] M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Wiley-VCH, Weinheim, Germany, 2000.

[17] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas.", IEEE Transactions on Semiconductor Manufacturing, vol. 18, no. 1, pp. 136-139, 2005.

[18] E. Beyne, "3D Integration Crossing IC technology, Packaging and Design Barriers", *Semicon West 2008, TechXPOT, Test Assembly & Packaging*, http://www.semiconwest.org/cms/groups/public/documents/web_content/ctr_024376.pdf

J Electron Test (2012) 28:15–25
DOI 10.1007/s10836-011-5270-3

Test Impact on the Overall Die-to-Wafer 3D Stacked IC Cost

Mottaqiallah Taouil · Said Hamdioui · Kees Beenakker · Erik Jan Marinissen

Received: 23 January 2011 / Accepted: 8 November 2011 / Published online: 8 December 2011

Abstract One of the key challenges in 3D Stacked-ICs (3D-SIC) is to guarantee high product quality at minimal cost. Quality is mostly determined by the applied tests and cost trade-offs. Testing 3D-SICs is very challenging due to several additional test moments for the mid-bond stacks, i.e., partially created stacks. The key question that this paper answers is what is the best test flow to be used in order to optimize the overall cost while realizing the required quality? We first present a framework covering different test flows for 3D Die-to-Wafer (D2W) stacked ICs. Thereafter, we present a cost model that allows us to evaluate these test flows. The impact of different test flows on the overall 3D-SIC cost for several die yields and stack sizes are investigated; a breakdown of the cost into test, manufacturing and packaging cost is also provided. Our simulation results show that both the test cost and the overall cost in D2W stacking strongly depends on the selected test flow; test flows with pre-bond and mid-bond stacking tests (performed during the stacking process) show a higher test cost share, but significantly reduce the overall 3D-SIC cost.

Keywords 3D test flow · 3D test cost · Die-to-Wafer stacking · 3D manufacturing cost · Through-Silicon-Via

Responsible Editor: Y. Zorian

M. Taouil ( ) · S. Hamdioui · K. Beenakker
Faculty of EE, Mathematic and CS, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: M.Taouil@tudelft.nl

S. Hamdioui
e-mail: S.Hamdioui@tudelft.nl

K. Beenakker
e-mail: C.I.M.Beenakker@tudelft.nl

E. J. Marinissen
IMEC vzw, 3D Integration Program, Kapeldreef 75, 3001 Leuven, Belgium
e-mail: erik.jan.marinissen@imec.be

1 Introduction

The potential benefits that 3D Stacked ICs (3D-SICs) offer is leading to an escalation of research and work both in academy and industry [6, 9, 11, 14–16, 20, 21]. The feasibility to stack dies allows long wires that normally cover long distances to be mapped on Trough-Silicon-Vias (TSVs). TSVs are holes that go through the silicon substrate filled with a conducting material. TSVs reduce the interconnect distance between stacked dies. This lowers the latency and power dissipation in such connections. Moreover, the incorporation of possibly heterogeneous dies results in a high transistor density at a smaller footprint. The ability to place the TSVs anywhere on the surface of the chip allows the establishment of high bandwidth communication between dies [6].

Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) and Die-to-Die (D2D) bonding [9] are the existing methods that could be employed in order to manufacture 3D-SICs. W2W bonding leads to highest throughput, as dies are processed in parallel at wafer level, and makes the manufacturing of tiny dies feasible [9]. Regarding yield, D2W and D2D are superior, due to the opportunity to apply Known-Good-Die (KGD) testing [9].

This paper focuses on D2W stacking as it is currently the most relevant stacking approach in industry.

Testing for manufacturing defects is required to satisfy the required product quality. In addition to the traditional defects that may occur during processing of planar wafers, new faults inherent to the 3D processes have to be considered. Good tested dies in the *pre-bond* test phase could get corrupted during the stacking. Typical sources of die failures during stacking include the processing steps involved in thinning, bonding, as well as TSV failures such as misalignments and opens [10]. If it is known beforehand that a particular stack is corrupted, silicon, stacking and bonding costs can be prevented for the successive dies that have to be stacked. The number of test moments, both for interconnects as well as dies, increases significantly during stacking. Pre-bond tests prevent corrupted dies from entering the stack, while post-bond tests verify the correctness of the dies and interconnects for the stack. To guarantee high 3D-SIC product quality at low cost, appropriate test flows need to be developed that take the different test phases (e.g. pre-bond testing, post-bond testing, etc.) into consideration.

This paper introduces a framework of test flows and analyzes the impact of such test flows on the overall cost of D2W based 3D-SIC. An appropriate cost model is developed to accurately evaluate the impact of the test flows while considering different process parameters such as stack size, die yield, etc.

The remainder of the paper is organized as follows. Section 2 introduces the test flow framework. Section 3 describes the cost model. Section 4 describes the simulation setup. Section 5 presents the simulation results and discusses them. Section 6 concludes the paper.

2 Test Flow Framework

This section presents first the differences between 2D and 3D test flows and shows that for 3D many test moments are possible. These test moments are thereafter compiled into a framework of test flows.

2.1 2D Versus 3D Test Flow

A conventional 2D test flow for planar wafers is depicted in Fig. 1a [13]. Here, usually two *test moments* are applicable; i.e., a wafer test prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low, since it prevents unnecessary assembly and packaging costs. The goal of the final test is to guarantee the final quality of the packaged chip. During the manufacturing of a 3D-SIC, additional test points can be defined for each partial stack. At each test point a distinction can be made between die tests and interconnect tests. Die tests ensure the functionality of individual dies, while interconnect tests ensure functional TSVs between dies. For 3D-SICs, four test moments can be distinguished in time as depicted in Fig. 1b, and explained next.

1. T_{pr}: n *pre-bond* wafer tests, since there are n layers to be stacked. T_{pr} tests prevent faulty dies entering the stack. Besides die test, preliminary TSV interconnect tests can be applied (in case of via-first [9]) as well. An example of a preliminary test that detects some faulty TSVs could be a capacitance test [5].
2. T_{mi}: n-2 *mid-bond* tests applicable for partial created stacks. In this case, either the dies, the interconnects, their combination or none of them can be tested. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g., die thinning, and bonding [10]. In the simulation model of our test flows, first the interconnects are tested and thereafter the dies in bottom up order (in case both are tested for); if a fault is detected in the interconnects, then there is no need to test the dies as the 3D-SIC will be faulty anyway. The reason for this particular test order is that the test cost for interconnects is considered cheaper, as will be explained in Section 3.
3. T_{po}: one *post-bond* test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Both interconnects and dies can be tested.
4. T_{fi}: one *final* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied at this test moment as well.

Note that in total there are $2 \cdot n$ different test moments.

Depending on whether one or more companies are involved in the manufacturing of 3D-SICS, different requirements can be set for the pre-bond wafer test quality [12]. If the wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place, the pre-bond test quality is subject to optimization. This means, there is not only the option to perform pre-bond testing or not, but also to perform pre-bond testing at a higher or lower test quality. Faulty

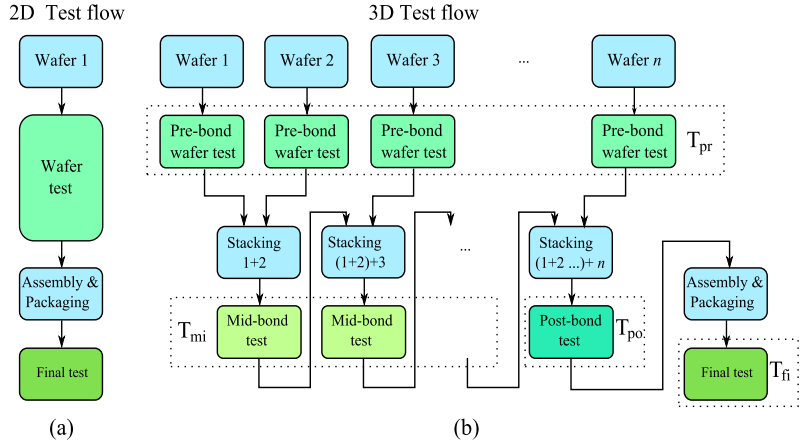


Fig. 1 2D versus 3D D2W test flows

undetected dies can be detected in a later stage, e.g., in higher quality final tests. Similarly, high quality mid- or post-bond tests (Known-Good-Stack tests) can be applied.

2.2 3D Test Flow Framework

The test flow framework for 3D D2W stacking can be extracted from the test flow moments depicted in Fig. 1b. Depending on whether no or at least one test is performed at each possible test moment, we can distinguish 2^{2n} possible test flows out of $2n$ test moments. This number will further increase if we consider that tests at each phase may target different faults; e.g., if we assume that T_{mi} may test (1) one or more interconnects, (2) one or more dies, (3) a combination of (1) and (2), or (4) none, then the number of possibilities for T_{mi} will be 4^{n-2}. This increases the number of test flows from 2^{2n} to $2^n\,(T_{pr}) \times 4^{n-2}\,(T_{mi}) \times 2\,(T_{po}) \times 2\,(T_{fi}) = 2^{3n-2}$. It is clear that considering all 'theoretical' possible test flows will result in an unmanageable space. Therefore, realistic assumptions have to be made in order to create a clear overview (without loss of generality) for the work presented in this paper. Our assumptions consist of the following.

1. A linear stacking approach is assumed, i.e., dies are stacked sequentially in a bottom-up approach starting from the bottom wafer. During stacking, it is assumed that only the *top* two dies and the interconnect between them could be corrupted; they are assumed to be defect-prone to stacking/bonding steps like heating, thinning, pressure.
2. All die tests are identical; a similar assumption applies to all interconnects.
3. Each test flow has to guarantee that a 3D-SIC is fault free before it is packaged to prevent unnecessary assembly and packaging cost. The test phases 'T_{pr}+T_{mi}+T_{po}' test each die and each interconnect of the SIC at *least* once.
4. The final test in T_{fi} is a complete test, i.e., all dies and interconnects are tested.

Because of Assumption 1, T_{mi} will test only for one of the following:

- Only for the *interconnect* between the top dies (i_t = top interconnect).
- Only for the *top dies* (d_t = dies top).
- For both the *top interconnect* and *top dies* ($i_t d_t$).
- none (n).

This results into $T_{mi} \in \{i_t, d_t, i_t d_t, n\}$.

Table 1 shows the test flow framework of all possible test flows based on the above assumptions. The first column denotes the two possibilities for T_{pr} (pre-bond test), either it is performed ('y') or not ('n'). The second column gives the four possible values of $T_{mi} \in \{i_t, d_t, i_t d_t, n\}$. The last column lists the different

Table 1 Test flow framework

Test flow	T_{pr}	T_{mi}	T_{po}
TF1	n	n	$i_a d_a$
TF2	n	i_t	$i_a d_t$
TF3	n	i_t	$i_t d_a$
TF4	n	$i_t d_t$	$i_t d_t$
TF5	y	n	$i_a d_a$
TF6	y	i_t	$i_a d_t$
TF7	y	i_t	$i_t d_a$
TF8	y	$i_t d_t$	$i_t d_t$

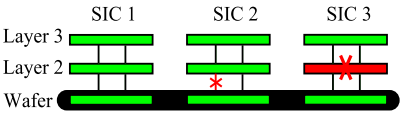


Fig. 2 Examples of defects (x) occurring during stacking

possible values of T_{po}. In order to satisfy Assumption 3 (a fault-free 3D-SIC prior to packaging) T_{po} is limited to the following values:

- $i_t d_t$: test for *top* interconnect and *top* dies.
- $i_t d_a$: test for *top* interconnect and *all* dies.
- $i_a d_t$: test for *all* interconnects and *top* dies.
- $i_a d_a$: test for *all* interconnects and *all* dies.

Each possible test flow is given a name in the table; e.g., TF1 denotes a test flow based on no T_{pr}, no T_{mi} and $T_{po} = i_a d_a$. There are eight test flows in total, i.e., TF1 to TF8.

To provide more insight into the different test flows and their impact on the total 3D-SIC cost, we consider the example shown in Fig. 2. It consists of three SICs with $n = 3$ layers each. For simplicity, it is assumed that all dies in the pre-bond phase were manufactured with 100% yield and that two faults occurred during stacking of Layer 2 on the bottom layer, one in SIC2 and one in SIC3. In SIC2, a fault occurred in the interconnects between the bottom die (i.e., Layer 1) and the die at Layer 2 (e.g., due to misaligned TSVs), while in SIC3 a defect occurred in Layer 2 (e.g., due to thinning). It is assumed that during the mid-bond and post-bond tests, first interconnects are tested, followed by the dies in bottom up order.

Table 2 shows the impact of four test flows TF1, TF2, TF3 and TF4 on three different cost factors: manufacturing, test, and packaging. Each entry in the table is composed of four numbers, associated with SIC1, SIC2 and SIC3 respectively, followed by their sum. The manufacturing, test and packaging costs for the three 3D-SICs are explained next.

The manufacturing cost is considered to include the number of used dies (the second column of the table) and the number of stacking operations that are performed (the third column of the table). For example, in TF1 only $T_{po} = i_a d_a$ is performed (see Table 1); therefore this will result in: (a) stacking of three dies per 3D-SIC, hence $3 + 3 + 3 = 9$ dies, and (b) two stacking operations per SIC, thus a total of $2 + 2 + 2 = 6$ stacking operations.

The test cost is classified according to the test phases defined in Section 2.1; i.e., pre-bond wafer tests T_{pr}, mid-bond tests T_{mi}, post-bond tests T_{po} and final tests T_{fi}. Note that T_{fi} is not included in the table as we assumed that final tests are the same for all test flows (Assumption 4). Except for the T_{pr} phase, each test phase distinguishes between tests for interconnects and tests for dies. Consider test flow TF4 which performs the following tests (see also Table 1):

- No pre-bond test (i.e., $T_{pr} = n$): no tests are executed and therefore no pre-bond tests for the three SICs are performed.
- Mid-bond tests consisting of (a) test for top interconnect and (b) tests for top dies (i.e., $T_{mi} = i_t d_t$). Note that there is $n - 2 = 1$ test moment. Hence, in this phase TF4 tests for the interconnects between the bottom layer and Layer 2 of each SIC, resulting in $1 + 1 + 1 = 3$ tests. In addition, TF4 tests for two bottom dies of SIC1 (i.e., the first two layers), no dies in SIC2 (since the interconnect found to be faulty during i_t tests) and the two bottom dies of SIC3 resulting into $2 + 0 + 2 = 4$ tests.
- Post-bond tests consisting of testing top dies and top interconnects of the SIC ($T_{po} = i_t d_t$). In this phase, TF4 tests only for the top interconnects and the two top dies of SIC1, not those of SIC2 and SIC3 as they are already considered faulty after the mid-bond tests were applied. This results in a total test of one interconnect and two dies during this phase.

The packaging cost is given in the last column of Table 2. Because of Assumption 3, the packaging cost is the same for all the four test flows. Only SIC1 will be packaged, while the other two SICs will be discarded.

Table 3 summarizes the cost required to manufacture and test the three 3D-SICs. The table clearly shows the

Table 2 Impact of test flows

TF	Manufacturing cost		Test cost					Packaging cost
	#dies	#stacking operations	T_{pr} #dies	T_{mi} #inter	#dies	T_{po} #inter	#dies	#packaged SICs
TF1	$3+3+3=9$	$2+2+2=6$	$0+0+0=0$	$0+0+0=0$	$0+0+0=0$	$2+1+2=5$	$3+0+2=5$	$1+0+0=1$
TF2	$3+2+3=8$	$2+1+2=5$	$0+0+0=0$	$1+1+1=3$	$0+0+0=0$	$1+0+1=2$	$3+0+2=5$	$1+0+0=1$
TF3	$3+3+2=8$	$2+2+1=5$	$0+0+0=0$	$0+0+0=0$	$2+2+2=6$	$2+1+0=3$	$2+2+0=3$	$1+0+0=1$
TF4	$3+2+2=7$	$2+1+1=4$	$0+0+0=0$	$1+1+1=3$	$2+0+2=4$	$1+0+0=1$	$2+0+0=2$	$1+0+0=1$

Table 3 Manufacturing versus test trade-off

Test flow	Manufacturing cost		Test cost	
	#dies	#stacking operations	#dies	# interconnects
TF1	9	6	5	5
TF2	8	5	5	5
TF3	8	5	9	3
TF4	7	4	6	4

cost trade-off between manufacturing and testing. For example, TF1 requires the manufacturing of nine dies and needs six stacking operations at a test cost of testing five dies and five interconnects. On the other hand, TF4 requires the manufacturing of seven dies and needs four stacking operations, at a test cost of six dies and four interconnects. Choosing the test flow resulting in optimal overall cost needs the evaluation of all possible test flows using an appropriate generic cost model; the latter is given in the next section.

3 Cost Model

To evaluate the impact of the different test flows on the overall 3D-SIC cost, an appropriate generic cost model is built. Figure 3 shows a diagram of this cost model; it considers three major input classes [19]:

- Manufacturing: this consists of all parameters related to 3D-SIC manufacturing process such as wafer cost, costs required for wafer processing, TSVs and 3D bonding and thinning, the number of dies per wafer, die yield etc.
- Test: This consists of all parameters related to DFT, test and test flows such as cost related to testing dies and interconnects. Test flows have a large impact on this cost since they determine when and what to test for.
- Packaging: The cost of 3D-SIC packaging.

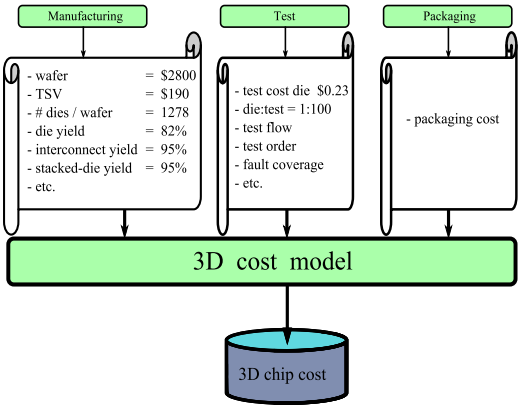


Fig. 3 Test cost model 3D D2W Stacking

The cost model is able to evaluate each test flow and calculates the overall 3D cost per test flow. In addition, it also determines the share of the test cost as compared to the overall cost. In fact, the model performs more *elaborate* and *comprehensive* calculations and analysis of those explained in the example of Section 2.2. The model has, for example, the ability to evaluate parallel testing of dies and it can handle more test flows than those described in Table 1. The model collects statistical data (in our case based on 1,000 wafers) while considering the different costs. The monitored data includes e.g., the number of used dies, the number of stacking/bonding operations, the number of packaged SICs, the number of tests performed (for dies and interconnect), etc.

Since the purpose of this work is to investigate the impact of different test flows rather than to observe the impact of different manufacturing processes (e.g., transistor feature size, TSV via-first or via-last, Face-to-Face or Back-to-Face bonding orientation, the number of TSVs etc.), the manufacturing costs are assumed to be constant; these will be discussed in Section 4.1. However, the test cost strongly depends on other parameters like die yield, interconnect yield, stacking yield, number of stacked layers, etc. These parameter are described in Section 4.2.

4 Simulation Setup

In order to appropriately perform simulations, different input parameters of the cost model have to be defined. These parameters are classified into fixed and variable ones.

4.1 Fixed Parameters

The fixed parameters of each of the input classes are given next.

Manufacturing Cost It includes wafer cost, costs required for wafer processing, TSV fabrication and 3D stacking/bonding. For wafers and their processing, we used the cost models of [17] and [4]; the total price of a 300 mm wafer is estimated at approximately $2,779. The model in [17] considers a variety of costs, including installation, maintenance, lithography and material. For TSV fabrication, the work of EMC-3D consortium [18] is used; the cost to fabricate 5 μm TSVs in a single wafer is assumed to be $190 and these cost

are additive to the wafer cost. We assume the cost of manufacturing TSVs to be 60% of the 3D stacking process cost [22].

Test Cost This cost is related to tests and test flows. To estimate the test cost per die, the model in [3] is used; it includes depreciation, maintenance and operating cost and assumes five ATE machines operating simultaneously. The derived test cost equals 3.82 $cent/second per die. Assuming a test time of 6 seconds per die, the test cost will be $0.23 per die. To estimate the interconnect test cost, a ratio of 1:100 between the test time of dies and interconnects is assumed (as in [23]).

Packaging Cost The packaging cost for 3D SICs used in our model is based on oral conversations with Boschman BV [2] and DIMES [8]. The costs are comprehensive and include machine, maintenance, labor and material cost.

4.2 Variable Parameters

Several variables, either related to manufacturing or test, have a large impact on the overall cost picture of 3D-SICs. Examples of the former are die yield, stack size, number of dies per wafer, stack yield, etc; and examples of the latter are fault coverage, test order, etc. The default values of the parameters used in our cost model are described next and are depicted in Fig. 3. In the remainder of this paper, these default parameters (depicted in Fig. 3) are referred to as the reference process.

Manufacturing The die yield is based on the stacking process in [23], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. This work assumes a defect density of $d_0 = 0.5$ defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50$ mm^2, the number of Gross Dies per Wafer (GDW) are estimated to be 1,278 [7]. With the negative binomial formula for yield, a die yield of $Y_D = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha} = 81.65\%$ is expected [3]. For the stack size we assume a default stack size $n = 5$. The stacking yield is composed of two parameters: the interconnect (TSV) yield Y_{INT} and the stacked-die yield Y_{SD}. In our simulations, the interconnect yield Y_{INT} is considered to be 95%. For the good dies that enter the stack, a small probability exists that they get corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 95% as well. Several research works assume a complete stack yield of approximately 95% [1, 23].

Test The order of testing is performed sequentially, bottom-up, starting first with the interconnects followed by the dies. In this work, we consider only the eight test flows defined in Table 1 for evaluation and analysis. A fault coverage of 100% is assumed for both dies and interconnect.

5 Simulation Results

In this section, we measure the impact of the test flows defined in Table 1 by using the cost model depicted in Fig. 3. We investigate not only the impact of the test flows on the overall cost, but also the share of test cost as compared with test, manufacturing and packaging; this will be performed for different die yields and stack sizes. The following experiments have been conducted:

1. **Impact of stack size** In this experiment, the impact of different test flows and the share of test cost will be investigated while considering different stack sizes n: $2 \leq n \leq 6$.
2. **Impact of die yield** Similar experiment as the previous one, but now by having a fixed stack size of $n = 5$, and variable die yield Y_D: $60\% \leq Y_D \leq 90\%$.
3. **Impact of stack yield** In this case, the reference process is used (e.g., $n = 5$, $Y_D = 81.65\%$, etc.), but the stack yield is varied; this yield consists of interconnect yield Y_{INT} and stacked-die yield Y_{SD}: $91\% \leq Y_{INT}, Y_{SD} \leq 99\%$.

5.1 Impact of Stack Size

Figure 4 depicts the relative overall 3D-SIC cost of the test flows for a stack size between $2 \leq n \leq 6$. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1 for each stack size. For $n = 2$, test flows TF1, TF2, TF3 and TF4 result in equal cost; the same thing applies to test flows TF5, TF6, TF7 and TF8. The reason is that in this case, the test flows are the same

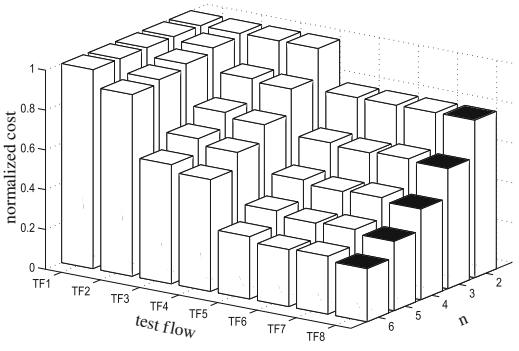


Fig. 4 Normalized overall cost for different stack sizes

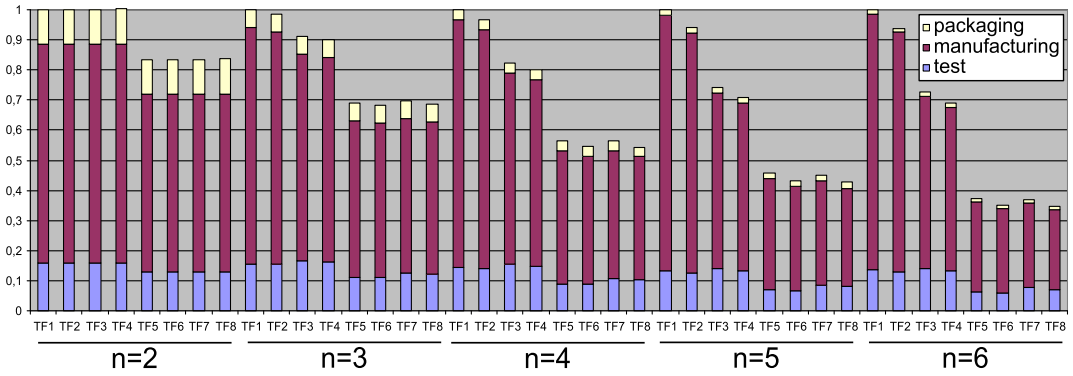


Fig. 5 Cost breakdown for different stack sizes

(as there are no mid-bond test moments). The following conclusions can be drawn from the figure:

- Test flows with pre-bond tests significantly reduce the overall cost. The larger the stack size n, the larger the reduction.
- TF8 is the most cost-effective test flow irrespective of n. The bars with black tops represent the test flows with the lowest costs per layer. For $n = 2$, TF5 until TF8 result in same cost.
- TF2 has a marginal impact on the cost reduction irrespective of n. This is because TF2 neither performs pre-bond tests nor die tests during the mid-bond phase. This is not the case for TF3 and TF4, as they both test for dies in the mid-bond phase.
- While test flow TF2 results in higher cost than test flow TF3, the reverse occurs for the test flows TF6 and TF7. Note that TF1 and TF3 are similar to TF6 and TF7, respectively, except that TF6 and TF7 also include pre-bond testing. In case of TF6 and TF7 only good dies will be stacked. Hence, it is cost-wise cheaper to test the interconnects (TF6) than to re-test the dies (TF7) during the mid-bond phase. Nevertheless, testing both interconnects and dies during the mid-bond phase is the most cost-effective test flow (i.e., TF8).

Figure 5 gives a different representation of Fig. 4, it breaks down the cost into manufacturing, test and packaging cost. In addition to the conclusions drawn from Figs. 4 and 5 shows that the share of packaging cost decreases as the stack size increases, while the test share increases with larger stack sizes. For test flow TF8, the test share is 15.4% for a stack size of $n = 2$, while this ratio increases to 20.6% for a stack size of $n = 6$. It is worth noting that although TF8 has the highest test cost share, it results in the lowest overal 3D-cost.

To get more insight into the impact of test flows and the cost break down, we will zoom on the case of the reference process. Figure 6 shows the overall cost normalized to TF1 for the eight test flows. TF3 results in an overall cost which is 74.27% of that of TF1. Since the stack yield is assumed to be much higher than the die yield, test flow TF3 (test for dies during the mid-bond phase) results in a lower cost than TF2 (test for interconnects only during the mid-bond phase). The reverse occurs for the test flows TF6 and TF7. Test flow TF8 is able to reduce the cost by 57.34% compared to TF1 (that considers only post-bond tests) and 6.7% as compared to test flow TF5 (that contains pre-bond and post-bond tests).

Figure 7 plots the breakdown of the 3D cost for the reference process. For each test flow, the shares of test, manufacturing and packaging are depicted. From the Figs. 7 and 5 the following can be concluded:

- The manufacturing cost is the most dominant cost factor for each test flow. However, the absolute

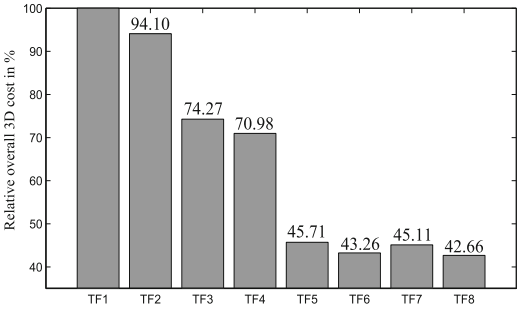


Fig. 6 Normalized 3D cost for the reference process

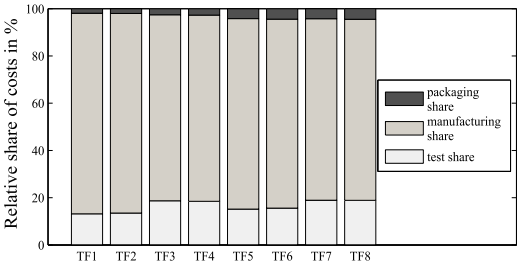


Fig. 7 Cost breakdown for the reference process

manufacturing cost depends strongly on the selected test flow.
- Test flows resulting in lower overall 3D cost do have a higher packaging cost share; this applies for test flows TF5 to TF8. This is because these test flows guarantee fault-free 3D-SICs before packaging.
- The share of test cost is between 13% and 19% depending on the test flow. Test flows containing die tests during the mid-bond phase result in a relatively higher test cost share as compared with the rest. For instance, test flow TF3, TF4, TF7 and TF8 result in a test cost share of about 19%.
- A higher test cost share does not necessarily result in higher overall cost.

5.2 Impact of Die Yield

Figure 8 depicts the relative 3D cost of the test flows with a die yield varying between $60\% \leq Y_D \leq 90\%$ for the reference process. Here, the 3D cost for each test flow is normalized to the 3D cost of TF1. From the figure we conclude the following.

- Test flows with pre-bond tests significantly reduce the overall cost. The lower the die yield, the larger the reduction (except for TF2 since this test flow does not test for dies during the pre-bond and mid-bond phases).
- TF2 has a marginal impact on the cost, irrespective of the die yield. This is not the case for TF3 and TF4, as they both test for dies in the mid-bond phase.
- Similar conclusions can be drawn as those from Fig. 4 for the test flows enabled with pre-bond testing. It is cheaper to test for interconnects only (TF6) than to test for dies only (TF7) during the mid-bond test phase. Nevertheless, testing for interconnects and dies during the mid-bond phase is the most cost-effective test flow (i.e., TF8).

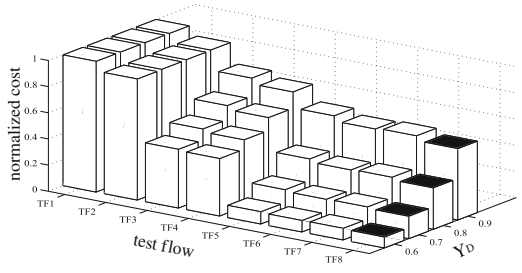


Fig. 8 Normalized cost for different die yields

Figure 9 gives the cost breakdown for the reference process and for $30\% \leq Y_D \leq 90\%$. For each Y_D, the overall costs are normalized to TF1. Within each bar, the share of test, manufacturing and packaging are depicted. The figure clearly reinforces the conclusions previously drawn from Fig. 8. For example, test flows with pre-bond tests (TF5 to TF8) result in the lowest overall cost irrespective of the value of the die yield; the cost difference with test flows without pre-bond test becomes more significant for lower yields. In addition, the figure reveals that TF8 results into the lowest overall cost in all cases, and that the test cost and packaging cost shares increases as the yield increases. The test and packaging share increase from 13 and 2%, respectively, for a die yield of 30%, to 20 and 5%, respectively, for a die yield of 90%. This figure also clarifies the importance of mid-bond tests; test flows with mid-bond tests result in lower cost. For example, TF8 results in 7% lower overall cost as compared to TF5; note that TF8 and TF5 are the same except that TF8 also consists of mid-bond tests.

5.3 Impact of Stack Yield

Figure 10 depicts the overall 3D cost versus stacked yield (i.e., interconnect Y_{INT} and stacked-die Y_{SD}) for the test flows. In the figure, Y_{INT} and Y_{SD} are set to either 91 and 99%. The 3D cost of the flows are normalized to the cost of TF1 for each different stack yield. The bars with black tops present test flows resulting in optimal overall cost per stacking yield. For example, for a stack yield of $[Y_{INT}, Y_{SD}] = [0.99, 0.99]$, TF6 is the most cost-effective test flow.

From the figure we conclude that TF6 and TF8 are the most cost-effective test flows. If Y_{SD} is very high (i.e., 99%), then TF6 is the best as it tests only for interconnect. However, in case $Y_{SD} = 91\%$, TF8 performs better, since it tests for dies during the mid-bond phase.

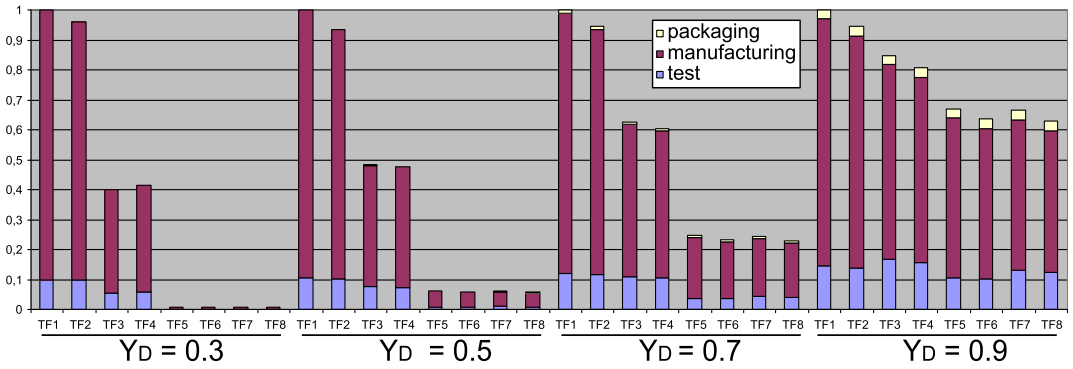


Fig. 9 Cost breakdown for variable die yield

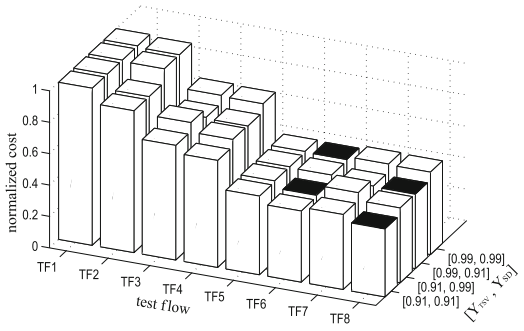


Fig. 10 Normalized overall cost for different stack yields

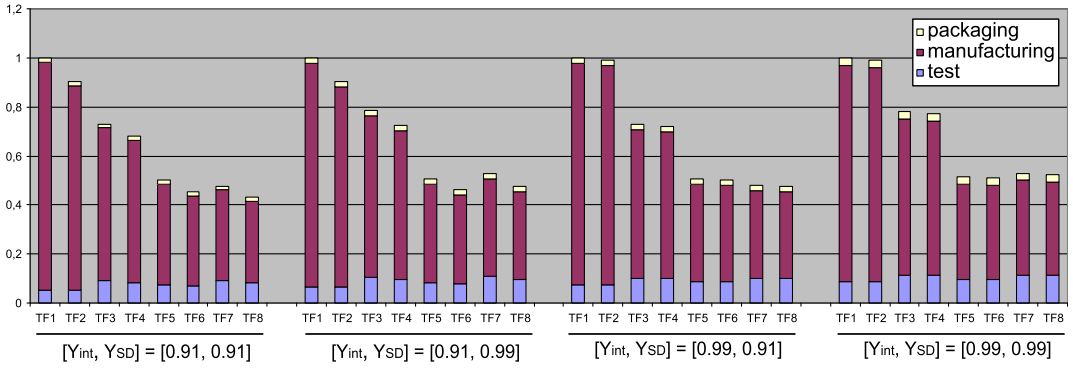


Fig. 11 Cost breakdown for variable die stack yield

Therefore, it is able to prevent unnecessary stacking of dies in faulty partial stacks.

Figure 11 shows the breakdown of the 3D cost. The higher the stack yield, the higher the test and packaging shares. For example, for TF8 the test and packaging shares are 19 and 4% respectively for a stack yield $[Y_{INT}, Y_{SD}] = [91, 91\%]$, while this increases to 21 and 6% for a stack yield of $[Y_{INT}, Y_{SD}] = [99, 99\%]$.

6 Conclusion

This paper investigated the impact of several 3D test flows on the total 3D cost in D2W stacking. It introduced a framework of test flows for 3D testing; each flow is based on a combination of tests applied at four test moments, i.e., the pre-bond wafer test, the mid-bond stack test, the post-bond test and the final test. A cost model that considers manufacturing, test and packaging cost is presented in order to evaluate the impact of different test flows on the overall cost.

The simulation results showed that the manufacturing cost is the most dominant in 3D stacking and strongly depends on the selected test flow. In addition, they revealed that test flows with pre-bond testing significantly reduced the overall cost. Mid-bond tests contributed to further cost savings. Although the share of test cost increases for such flows, the overall cost is significantly reduced. The cost saving increase with lower die yields and larger stack sizes. The conclusion of the paper indicates that in order to manufacture 3D-ICs at optimum cost, any DFT has to consider not only the infrastructure for pre-bond tests, but also for mid-bond tests for both dies and interconnects.

References

1. Beyne E (2008) 3D Integration Crossing IC technology, Packaging and Design Barriers, Semicon West, TechXPOT, Test Assembly & Packaging. http://www.semiconwest.org/cms/groups/public/documents/web_content/ctr_024376.pdf. Accessed 2010
2. Boschman Technology (2011) http://www.boschman.nl/. Accessed 2011
3. Bushnell M, Agrawal V (2000) Essentials of electronic testing for digital. Memory and mixed-signal VLSI circuits. Springer
4. Chappell J (2002) What costs most in 300 mm? As materials management becomes more complex, FOUP becomes first line of defense, http://findarticles.com/p/articles/mi_m0EKF/is_24_48/ai_87145967/. Accessed 2011
5. Chen P, Wu C, Kwai D (2009) On-chip TSV testing for 3D IC before bonding using sense amplification. Asian test symposium, pp 450–455
6. Davis WR et al (2005) Demystifying 3D ICs: the pros and cons of going vertical. IEEE Des Test Comput 22(6):498–510
7. de Vries DK (2005) Investigation of gross die per wafer formulas. IEEE Trans Semicond Manuf 18(1):136–139
8. Delft Institute of Microsystems and Nanoelectronics (2011) DIMES. http://www.dimes.tudelft.nl/. Accessed 2011
9. Garrou P (2008) Christopher Bower and Peter Ramm. Handbook of 3D Integration, Wiley-VCH
10. Lee H-HS, Chakrabarty K (2009) Test challenges for 3D integrated circuits. IEEE Des Test Comput 25(5):26–35
11. Loh G et al (2007) Processor design in 3D die-stacking technologies. IEEE Micro 27(3):31–48
12. Marinissen EJ (2010) Testing TSV-based three-dimensional stacked ICs. Design, automation and test in Europe, pp 1689–1694
13. Marinissen EJ, Zorian Y (2009) Testing 3D chips containing through-silicon vias. International test conference, pp 1–11
14. Patti RS (2006) Three-dimensional integrated circuits and the future of system-on-chip designs. Proc IEEE 94(6):1214–1224
15. Puttaswamy K et al (2007) Processor design in 3D die-stacking technologies. IEEE Trans Comput 27(3):31–48
16. Puttaswamy K et al (2009) 3D-integrated SRAM components for high-performance microprocessors. IEEE Trans Comput 58(10):1369–1381
17. Sematech Wafer Cost Comparison Calculator (2011) http://ismi.sematech.org/modeling/agreements/wafercalc.htm. Accessed 2011
18. Siblerud P (2008) Emc-3d consortium develops process and cost model for interconnect thru-silicon-via or (iTSV^{TM}) structures. http://emc3d.org/documents/pressReleases/2008/EMC3D_iTSV_CoO_PressRelease_final_Sept4_2008.pdf. Accessed 2011
19. Taouil M et al (2010) Test cost analysis for 3D die-to-wafer stacking. Asian Test Symposium, pp 435–441
20. Thorolfsson T et al (2009) Comparative analysis of two 3D integration implementations of a SAR processor. IEEE international conference on 3D system integration, pp 1–4
21. Tsai Y-F et al (2008) Design space exploration for 3-D cache. IEEE Trans Very Large Scale Integr (VLSI) Syst 16(4):444–455
22. Velenis D et al (2009) Impact of 3D design choices on manufacturing cost. IEEE international conference on 3D system integration, pp 1–5
23. Verbree J et al (2010) On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking. IEEE European test symposium, pp 269–274

Mottaqiallah Taouil received his MSc with honors in Computer Engineering from the Delft University of Technology (TUDelft), Delft, the Netherlands. He is currently pursuing a PhD at the same university in the Dependable Nano-computing group. His research interests include Reconfigurable Computing, Embedded Systems, VLSI Design & Test, Built-In-Self-Test, 3D stacked ICs, 3D Architectures, (3D) Design for Testability, (3D) Yield analysis and 3D Memory Test structures.

Said Hamdioui received the MSEE and PhD degrees (both with honors) from the Delft University of Technology (TUDelft), Delft, The Netherlands. He is currently a Associate Professor

at the Computer Engineering Lab. of TUDelft. Prior to joining TUDelft, Hamdioui worked for Microprocessor Products Group at Intel Corporation (in Santa Clara and Folsom, Califorina), for IP and Yield Group at Philips Semiconductors R&D (Crolles, France) and for DSP design group at Philips/ NXP Semiconductors (Nijmegen, The Netherlands). He is the recipient of European Design Automation Association (EDAA) Outstanding Dissertation Award 2001, for his work on memory test techniques that have a wide-spread proliferation in the chip design industry; he is also the winner of IEEE Nano and Nano Korea award at IEEE NANO 2010—Joint Symposium with Nano Korea 2010. He was nominated for The Young Academy (DJA) of the Royal Netherlands Academy of Arts and Sciences (KNAW) in 2009. His research interests include dependable nano-computing and VLSI Design & Test (defect/fault tolerance, reliability, security, nano-architectures, Design-for-Testability, Built-In-Self-Test, 3D stacked IC test, etc). He has published one book and over 100 technical papers. He serves on the editorial board of the Journal of Electronic Testing: Theory and Applications (JETTA).

Kees Beenakker was born in Leiden in 1948. After the gymnasium he studied chemistry and physics at Leiden university. In 1971 he got his M. Sc. and joined as an Ph. D. student the FOM-Institute for Atomic and Molecular Physics in Amsterdam. After he obtained his Ph. D. in 1974 he joined Philips Research Laboratories in Eindhoven. There he was involved in various research projects related to IC technology. In 1982 he moved to the Philips Semiconductor Division in Nijmegen to become head of the corporate assembly process and equipment development. In that position first intensive contacts were established with the microelectronics industry in the Far East. In 1987 he resigned at Philips and became cofounder of Eurasem, a European hi-rel IC assembly company. In 1989 Kees Beenakker joined Dimes and is since 1990 full professor at the faculty of EEMCS (Electrical Engineering, Mathematics and Computer Science). From 1990 till 2004 he was chairman of the ECTM laboratory. In 1999 he was appointed chairman of the department of Microelectronics and Computer Engineering. He is a member of the national Medea advisory committee, member of the scientific board of the Debije Institute, the ENIAC scientific council and board member of Boschman Technologies. He is a founder of the Else Kooi Foundation, the SAFE conference, the Tsing Hua-TU Delft training centre of microelectronics technology in Beijing and the Fudan-TU Delft International school of microelectronics in Shanghai. Since March 2006 he holds a honorary guest professorship at the Tsinghua University in Beijing. In March 2007 he was appointed scientific director of DIMES, the Delft institute of microsystems and nanoelectronics. Since June 2008 he is elected chaiman of the academic council of Point-One, the national initiative on nanoelectronics and embedded systems. His specific research interests include technology for thin films and integrated circuits.

Erik Jan Marinissen is Principal Scientist at IMEC vzw in Leuven, Belgium. Previously, he worked at NXP Semiconductors and Philips Research, both in Eindhoven, The Netherlands. Marinissen holds an MSc degree in Computing Science (1990) and a PDEng degree in Software Technology (1992), both from Eindhoven University of Technology. Marinissen's research interests include all topics in the domain of test and debug of micro-electronics. He is co-author of over 150 journal and conference papers and co-inventor on eight granted US and EP patent families. Marinissen is recipient of the ITC 2008 and ITC 2010 Most Significant Paper Awards and Best Paper Awards at the Chrysler-Delco-Ford Automotive Electronics Reliability Workshop 1995 and the IEEE International Board Test Workshop 2002. He served as Editor-in-Chief of IEEE Std. 1500. He is a founder of workshops on 'Diagnostic Services in Network-on-Chips' (DSNOC), '3D Integration', and 'Testing of Three-Dimensional Stacked Integrated Circuits' (3D-TEST). He serves on numerous conference committees, including ATS, ETS, DATE, ITC, and VTS, and on the editorial boards of IEEE Design & Test of Computers, IET Computers and Digital Techniques, and Springer's Journal of Electronic Testing: Theory and Applications (JETTA). Marinissen is a Fellow of IEEE and Golden Core Member of Computer Society.

Stacking Order Impact on Overall 3D Die-to-Wafer Stacked-IC Cost

Mottaqiallah Taouil Said Hamdioui

Computer Engineering Laboratory
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: {M.Taouil, S.Hamdioui}@tudelft.nl

***Abstract*— Three-dimensional Stacked IC (3D-SIC) is a promising technology gaining a lot of attention by industry. Such technology promises lower latency, lower power consumption and a smaller footprint as compared to planar ICs. Reducing the overall 3D-SIC manufacturing cost is a major challenge driving the industry. The process of stacking the dies together is an integral part of 3D-SIC manufacturing process; hence, it impacts the overall cost. This paper introduces out-of-order stacking and compares it with the conventional in-order stacking from cost point of view. In-order stacking restricts the stacking of the dies in a bottom-up sequential order, while out-of-order stacking poses no restrictions and the order is free as long as it is realistic. The simulation results show that out-of-order stacking ends up in lower cost than in-order stacking, and that the difference increases for larger stack sizes and lower stacking yield. For example, our case study shows that for a 3D-SIC with a stack size of 6 layers, out-of-order stacking outperforms the in-order one with up to 6% using the optimal test flow.**

Keywords: *3D test flow, 3D test cost, Die-to-Wafer stacking, 3D manufacturing cost, 3D stacking.*

I. Introduction

The potential benefits that 3D Stacked ICs (3D-SICs) offer is leading to an escalation of research and work in academy and industry [1–7]. The facility to stack dies allows long wires that normally cover long distances to be mapped on Trough-Silicon-Vias (TSVs). TSVs are holes that go through the silicon substrate filled with a conducting material. TSVs reduce the interconnect distance between stacked dies. This lowers the latency and power dissipation in such connections. Moreover, the incorporation of possibly heterogeneous dies results in a high transistor density at small footprint. The ability to place the TSVs anywhere on the surface of the chip allows to establish high bandwidth communication between the dies [1].

Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) and Die-to-Die (D2D) bonding [2] are the existing methods that could be employed in order to manufacture a 3D-SICs. W2W bonding leads to highest throughput, as the processing of the dies goes in parallel, and makes the manufacturing of tiny dies feasible [2]; however, it suffers from low compound yield [8,9]. Regarding yield, D2W and D2D are superior, due to the opportunity to apply Known-Good-Die (KGD) testing [2]. This paper focuses on D2W stacking as it is currently the most relevant stacking approach in industry.

Testing for manufacturing defects is required to satisfy the required product quality. In addition to the traditional defects that may occur during processing of planar wafers, new faults inherent to the 3D processes have to be considered. Good tested dies in the *pre-bond* test phase could get corrupted during the stacking. Typical sources of die failures during stacking include the processing steps involved in thinning, bonding, as well as TSV failures such as misalignments and opens [10]. The number of test moments, both for interconnects as well as dies, increases significantly during stacking [11]. To guarantee high 3D-SIC product quality at low cost, appropriate test flows need to be developed. For example, in D2W stacking dies may not only require testing before they are stacked (i.e., pre-bond testing), but also during and after stacking (post-bond testing).

In our previous work [11], we showed that there are many test flows that can be used to test a 3D-SIC manufactured using in-order stacking, and that each test flow result in a different overall 3D-SIC cost; in the in-order stacking approach, dies are stacked sequentially in a bottom-up approach starting from the bottom wafer. Each successive stacking operation involves the next layer of the stack. In this paper we will investigate out-of-order stacking and the impact of different test flows on the overall cost; out-of-order stacking removes the restriction of the sequential bottom-up stacking order and allows the dies to be stacked in any realistic order. A comparison of in-order and out-of-order stacking order is provided for different process parameters such as stack size, stack yield, etc.

In this paper, the test flows and cost model for in-order 3D D2W-stacked ICs presented in our previous work [11] are used as starting point. They are modified and extended to support cost modeling for out-of-order stacked 3D-SICs. This allows us to compare in-order and out-of-order stacking in terms of cost. The contribution of the paper is the following:

- Modification and expansion of the test flow framework and cost model for out-of-order stacking.
- The innovation of the in-order and out-of-order stacking concept for 3D-SICs.
- Test evaluation and comparison of in-order and out-of-order stacking.
- Test cost analysis for out-of-order stacking.

The remainder of the paper is organized as follows. Sec-

978-1-4244-9756-0/11/$26.00 ©2011 IEEE

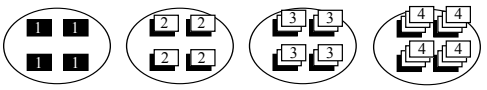


Fig. 1. In-order D2W stacking for a four layered 3D-SIC

tion II introduces in-order stacking and presents an alternative to this conventional method by allowing a free stacking order, i.e., out-of-order stacking. Section III describes the experimental setup and consists of three parts: (a) a brief description of the test framework, (b) a cost model that is used to evaluate out-of-order stacking, and (c) a description of the selected parameters for the experiments. Section IV presents the simulation results and analyzes them. Section V concludes the paper.

II. Classification of Stacking Order

The manufacturing process steps involved in the fabrication of 3D-SICs are still in its infant stage. The best design flow to consider is far from known and several processes are still under research [2]. Here, we are considering the impact of different stacking orders. First, Section II-A describes the more natural way of in-order stacking, while Section II-B introduces out-of-order stacking.

A. In-order 3D D2W stacking

A straightforward stacking methodology in manufacturing a 3D-SIC using D2W bonding is based on the sequential stacking of dies. Starting from a bottom wafer, each successive layer in the 3D-SIC is stacked in sequence. We denote this stacking methodology as in-order stacking. Figure 1 depicts this in-order stacking for a four layer 3D-SIC. The dies on the bottom wafer are depicted in black and the numbers denote the die index in the stack. First dies of layer 2 are stacked on the four dies of the bottom layer. After creating a partial stacked IC, the dies of layer 3 are stacked on them; etc. The in-order stacking approach suffers from some drawbacks, as described next.

Due to this nature of stacking, faults introduced in later stacking stages impact the cost severe, as larger partial stacks have to be thrown away. Changing the order of stacking may reduce the overall cost. A second drawback of in-order stacking is the excessive access to the bottom die to test the ICs. Note that the partial created stacks increase the number of test opportunities considerably if exhaustive testing is performed. In order to perform these intermediate tests, i.e. testing in partial created stacks, the access to the stack has to be performed through the bottom wafer and excessive probing of the same die might form a limitation [10]. For the example shown in Figure 1, the bottom wafer may be accessed five times for testing purposes during the 3D-SIC manufacturing if intermediate testing is performed. The first test that is applied is the wafer test (pre-bond test). After each created intermediate stack tests for both dies or interconnects can be applied (i.e., after the stacking of dies

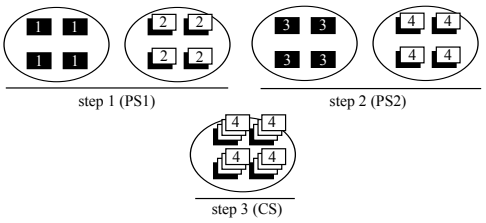


Fig. 2. Out-of-order D2W stacking for a four layered 3D-SIC

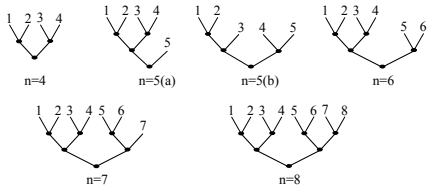


Fig. 3. Out-of-order stacking topologies for different stack sizes

from layer 2 and 3). Before packaging (after dies from layer 4 are stacked) and after packaging, similar tests for dies and interconnects can be applied. In the next section, out-of-order stacking is addressed and shows how it improves these drawbacks. It is assumed that each layer provides the Design for Testability (DFT) infrastructure such that it supports testing of intermediate stacks [12].

B. Out-of-order 3D D2W stacking

In out-of-order stacking, the order which the dies are stacked is modified. Consider for example the four layer 3D-SIC in Figure 2. Here, the stacking process consists of three steps; in the first step, the dies of the second layer are stacked on the bottom wafer to create a partial stack PS1. In the second step, the dies of the fourth layer are stacked on those of the third layer to create a second partial stack PS2. In the last step, the partial stack PS2 is stacked on partial stack PS1 to create the complete stack (CS). In all cases, the stacking of the dies is based on D2W stacking.

The number of times the bottom wafer has to be probed using this particular stacking order reduces by one, as testing the partial stack PS2 does not include the bottom die. The improvement increases with larger stack sizes. For example, for a stack size of eighth layers the number of times the bottom wafer is accessed is for the most extensive test flows ten times (1 pre-bond test, 7 intermediate stack tests, 1 pre-packaging and 1 post-packaging test), while for out-of-order stacking this is reduced to five accesses.

The biggest advantage most likely is the gain in cost reduction that is obtained due to the an unrestricted stacking order. As the partial stacks are stacked in out-of-order, the partial stacks that are tested faulty are on average of smaller size and thus can save cost in case detected faulty. Figure 3 shows binary trees representing the different stacking orders that are considered in this work. For example, for a stack size n=4, the stacking sequence is based on Figure 2. First,

dies from layer two are stacked on wafer with die index one, followed by the stacking of dies from layer four on wafer with die index three, in the next step both these temporal stacks are combined and stacked in to a single four layer chip. The considered stack sequences for the other stack sizes are depicted in the figure as well. For a stack size of five layers, we consider two different stacking approaches denoted by 5(a) and 5(b) in the figure. Note that both approaches (in-order and out-of-order) result in the same stacking order for a stack consisting of two or three layers.

III. Experimental Setup

In our previous work, an evaluation and analysis of the impact of different test flows on the 3D-SIC overall cost using in-order stacking has been presented in [11]. To perform a fair comparison with out-of-order stacking, a modified version of the test framework and cost model used in [11] will be used here in a similar way as well. The test framework and cost model are described in Sections III-A and III-B respectively. Section III-C provides the simulation parameters considered in this work.

A. Testflow Framework

The framework consists of several test flows that differ both in the applied tests (e.g., die test, interconnect test) and the test moment (e.g., wafer, partial stack, pre-packaging etc.). A conventional 2D test flow for planar wafers is depicted in Figure 4(a) [13]. Here, usually two *test moments* are applicable; i.e., a wafer test prior to packaging and the final test after packaging. The wafer test can be cost-effective when the yield is low, since it prevents unnecessary assembly and packaging costs. The goal of the final test is to guarantee the required quality of the final packaged chip. For 3D-SICs, four test moments can be distinguished in time as depicted in Figure 4(b). It contains four test moments (the dashed boxes) and are classified in: pre-bond testing (T_{pb}), intermediate stack test (T_{in}), pre-packaging test (T_{pr}), and post-packaging test (T_{pr}).

1) T_{pb}: n *pre-bond* wafer tests for each individual die on the wafer (n is the number of stacked layers).
2) T_{in}: n-2 *intermediate* tests applicable during the intermediate stacking and bonding.
3) T_{pr}: one *pre-packaging* test. This test can be applied after the complete stack is formed.
4) T_{po}: one final *post-packaging* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC.

In in-order stacking, it is assumed that after each stacking step only the *top* two dies d_t and the interconnect i_t between them could get corrupted since these dies are most susceptible to the stacking/bonding steps like heating, thinning, pressure, and TSV-related defects. The results of the test that can be applied in eache phase (i.e., T_{pb}, T_{in}, T_{pr} and T_{pr}) are shown in Table I:

- During T_{pb}, a wafer is tested for or not.

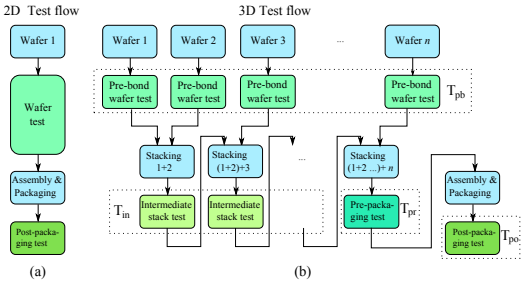


Fig. 4. 2D versus 3D D2W test flows.

TABLE I
Test flow framework

test flow	T_{pb}	T_{in}	T_{pr}
TF1	n	n	$i_a d_a$
TF2	n	i_t	$i_a d_t$
TF3	n	i_t	$i_t d_a$
TF4	n	$i_t d_t$	$i_t d_t$
TF5	y	n	$i_a d_a$
TF6	y	i_t	$i_a d_t$
TF7	y	i_t	$i_t d_a$
TF8	y	$i_t d_t$	$i_t d_t$

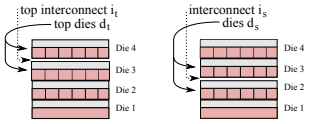


Fig. 5. Tested dies and interconnects in in-order and out-of-order stacking

- In the intermediate T_{in} phase, a die (d_t) or the interconnect (i_t) between two layers is tested for, both are skipped (n) or both of them are applied ($i_t d_t$).
- In the pre-packaging phase similarly a die or all dies (d_t or d_a respectively) and the top interconnect or all interconnects (i_t or i_a respectively) are tested for.

The framework ensures that each test flow satisfies a fault-free 3D-SIC prior to packaging to prevent unnecessary packaging and assembly costs. The final post-packaging test assumes a complete stack test. The complete derivation and assumptions of the model are described in more detail in our previous work [11].

Out-of-order stacking affects the intermediate and pre-packaging tests only. The different stacking order effects those dies that are most susceptible to faults at each stacking step. An example is depicted in Figure 5; the left side of the figure shows that the top two dies and top interconnect are susceptible to faults during intermediate stacking when using in-order stacking. Therefore, intermediate tests may include a test for top interconnect and a tests for the top two dies. The right side of the figure shows the stacking of two partial created stacks of size two using a out-of-order stacking. In this case, the dies and interconnect that assumed to be susceptible for defects are those that are involved in the stacking. Thus, the direct involved stacked dies d_s (i.e., die

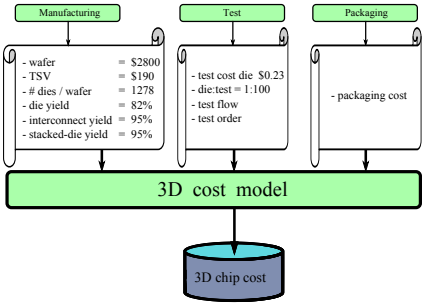


Fig. 6. Test cost model 3D D2W Stacking.

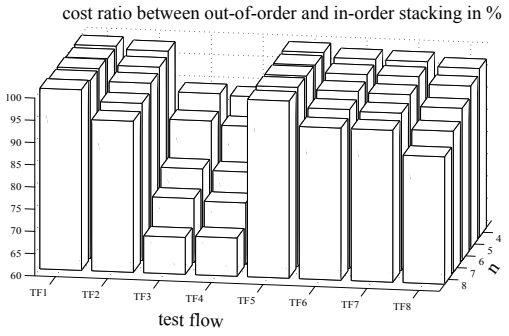


Fig. 7. Cost ratio out-of-order vs in-order stacking for variable stack sizes.

2 and 3 for this case) and the interconnect i_s (i.e., between die 2 and 3 for this case) are considered most susceptible to faults. Therefore, this requires the testing of different interconnects and dies in the intermediate test phase. In case in-order stacking is used this implies that d_s=d_t and i_s=i_t.

B. Cost Model

The cost model for the evaluation of the cost for in-order stacked ICs is explained in more detail in [11]. The cost model is extended in order to use it for out-of-order stacking. Figure 6 shows the block diagram of the cost model; it consists of three inputs:

- Manufacturing: It consist of all parameters related to 3D-SIC manufacturing; these are e.g., wafer cost, costs required for wafer processing, TSVs and 3D bonding and thinning.
- Test: This consists of all parameters related to DFT and test such as the cost related to testing dies and interconnects. Test flows have a large impact on this cost since they determine when and what to test for.
- Packaging: The cost of 3D-SIC packaging.

The values of the parameters used in this work are also depicted in Figure 6.

C. Simulation Parameters

Several parameters influence the performance of the test flows in terms of cost. These parameters include die yield, stack size, number of dies per wafer, stack yield, packaging yield, fault coverage, etc. The selected parameters for our reference process are described and their values are depicted in Figure 6. The reference process describes the default simulation parameters. The cost related parameters are assumed to be the same during all the experiments and the justification of their values is provided in [11].

Now, we describe the values of the default experiment. In the next section, we either use this reference process or vary a single parameter at a time. The die yield is based on the stacking process in [8], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. The work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50\ mm^2$, the number of Gross Dies per Wafer (GDW) are estimated to be 1278 [14]. With the negative binomial formula for yield, a die yield of $Y_D = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [15]. For the stack size we assume a default stack size n=5. The stacking yield is composed out of two parameters: the interconnect yield Y_{INT} and the stacked-die yield Y_{SD}. In our simulations, the TSV yield Y_{INT} is considered to be 95%. For the good dies that enter the stack, a small probability exists that they get corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 95% as well. Several research work assume a complete stack yield of approximately 95% [8,17].

IV. Simulation Results

This section considers two type of experiments. The first set of experiments evaluate the impact of out-of-order stacking on the overall 3D-SIC cost, and compare the results with those of in-order stacking. The second set of experiments focus in more depth on the out-of-order stacking approach.

A. Comparison of in-order and out-of-order stacking

In order to evaluate out-of-order against in-order stacking, three experiments are conducted. In the first experiment, the impact of the stack size is simulated, while keeping the other parameters fixed. Likewise, in the second experiment the die yield is varied, while the last experiment considers a variable stack yield.

Figure 7 shows cost ratio in percentage between out-of-order and in-order stacking for stack sizes between n=4 and n=8. For n=5 we used the stacking order of 5(b) as depicted in Figure 3, since this stacking order resulted in lower cost as compared to 5(a). From the figure the following can be concluded.

- Out-of-order stacking realizes the best cost reduction for test flows TF3 and TF4. For example, for n=4 out-of-order stacking results in a 10% cheaper 3D-SIC as compared to in-order stacking. The reduction becomes more significant as the stack size increases. This reduction is due to the fact that the these test flows do no perform pre-bond testing and that in the case of

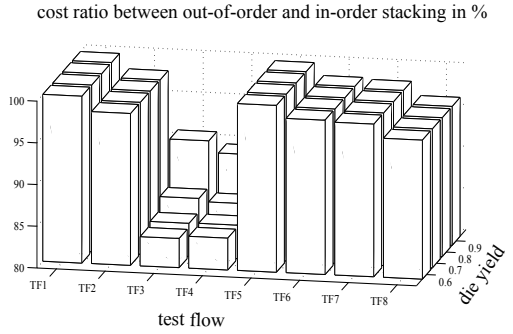


Fig. 8. Cost ratio in-order vs out-of-order stacking for variable die yields.

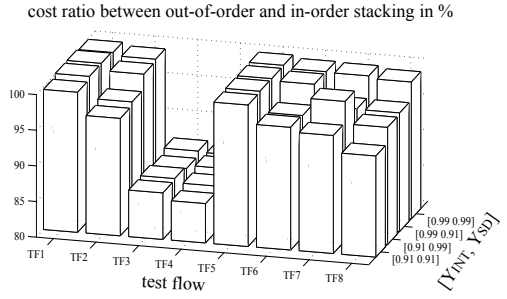


Fig. 9. Cost ratio in-order vs out-of-order stacking for variable stack yields.

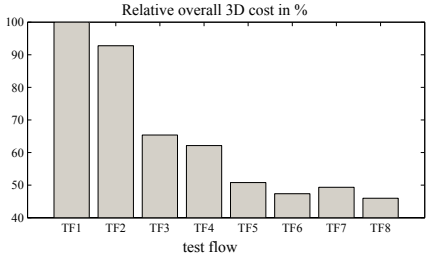


Fig. 10. Relative cost for the reference process using out-of-order stacking.

out-of-order stacking partial faulty stacks result in less wasted dies. Note that both TF3 and TF4 include die testing during the intermediate stacking. Test flow TF2 is not able to reduce the cost in a similar way, as it test for interconnects only during stacking.

- TF1 and TF5 result in the same overall cost irrespective of the used stacking order. The explanation for this is that the actual differences created by in-order and out-of-order stacking are due to the performed intermediate tests. Therefore, in cases where no intermediate testing are performed such as by TF1 and TF5, both stacking approaches result in the same cost.
- Out-of-order stacking is cost wise more effective with larger stack sizes (except for TF1 and TF5).
- It has been shown in [11] that TF8 results in lowest overall cost. The comparison of TF8 for out-of-order and in-order stacking shows that out-of-order stacking can reduce the cost. For example, for a stack size of six layers this cost reduction is 6%.

The second experiment focuses on the variation of die yield. Figure 8 shows for the reference process the impact of the die yield on the cost ratio between out-of-order and in-order stacking. From the figure the following can be concluded:

- For the same reasons as in the previous experiment, TF1 and TF5 have no influence on the cost alteration and test flows TF3 and TF4 show best improvements.
- For the test flows with pre-bond tests (i.e., TF5 until TF8), the die yield has no influence on the relative cost. This is because the difference in cost between the two stacking approaches is due to the intermediate tests. Since the same good dies enter the stack, the cost reduction is not a function of the die yield.
- Test flows TF2 shows a similar behavior since it test for interconnects only during testing. The change in die yield does not affect the cost improvement.

The results of the third experiment are shown in Figure 9. The figure shows the cost ratio of out-of-order and in-order stacking for an interconnect yield Y_{INT} and a stacked die yield Y_{SD} both taking values of 91% and 99%. The following can be concluded from the figure.

- Similarly as in the previous two experiments, TF1 and TF5 have no influence on the cost alteration and test flows TF3 and TF4 show best improvements.
- In case the stacked die yield is high (i.e, 99%), less improvement is gained for the test flows TF3 and TF7 as these test for dies only during the intermediate testing. Likewise, as the interconnect yield is high (i.e., 99%) test flows TF2 and TF6 perform less good as they test for the interconnects only.

B. Analysis of out-of-order stacking

The out-of-order stacking approach is able to reduce the cost further as compared to in-order stacking. In this section, we compare the eighth test flows with each other while considering out-of-order stacking only. Figure 10 shows for the reference process the normalized cost with respect to TF1 of all the test flows.

A similar trend is observed for the test flows as in the case for in-order stacking approach in [11]. Pre-bond enabled test flows (TF5 until TF8) are cost wise more efficient. Intermediate tests are able to reduce the further cost when performed after the pre-bond tests, i.e., the cost of a 3D-SIC is lower for TF6, TF7 and TF8 with respect to TF5. Test flow TF8, shows highest cost reduction for the selected parameters and includes both interconnect and die test during intermediate stacking.

In [11], it has been show that test flows with pre-bond tests results in lower overall 3D-SIC cost than those without. Moreover, TF8 has been shown on be the best test

flow in terms of cost reduction for the case study considered in the work. The Figures 7, 8 and 9 show that generally speaking using out-of-order stacking further reduces the cost. Therefore, using TF8 and out-of-order stacking is the optimal scenario that will reduce in optimal cost; at least for the case study considered in this work.

V. Conclusion

In this paper, we introduced the concept of out-of-order stacking for 3D-SIC and compared it with the conventional in-order stacking approach. The comparison and the analysis has been done while varying several parameters; these parameters include different stack sizes, die yields, stack yields for all the different test flows.

The simulation results show that the out-of-order stacking approach results in lower overall cost as compared to in-order stacking; this is because testing during out-of-order stacking reduces the number of wasted faulty dies (or partial stacks). This cost reduction depends on the selected test flow. For example, for the cost-wise cheapest test flow TF8 (consisting of pre-bond, intermediate, pre-packaging and post-packaging tests) and using our reference process with a stack size of six layers, out-of-order stacking is able to reduce the cost further with 6% as compared to in-order stacking. The reduction becomes more significant as the stack size increases or when the stack yield decreases. This reduction is due to the fact that when using out-of-order stacking the detection of faults within partial stacks results in less wasted dies.

References

[1] 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 on Computers*, vol. 22, no. 6, pp. 498-510, 2005.

[2] P. Garrou, Christopher Bower and Peter Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.

[3] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, vol. 94, no. 6, 2006.

[4] G.H. Loh, Y. Xie, and B. Black, "Processor Design in 3D Die-Stacking Technologies", *IEEE Micro*, vol. 27, no. 3, pp. 31-48, 2007.

[5] K. Puttaswamy and G.H. Loh, "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369-1381, 2009.

[6] Y-F. Tsai, F. Wang, Y. Xie; N. Vijaykrishnan and M.J. Irwin, "Design Space Exploration for 3-D Cache", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 444-455, 2008.

[7] T. Thorolfsson, S. Melamed, G. Charles and P.D.Franzon, "Comparative Analysis of Two 3D Integration Implementations of a SAR Processor", *IEEE International Conference on 3D System Integration*, pp. 1-4, 2009.

[8] J. Verbree, E.J. Marinissen, P.Roussel and D. Velenis, "On the Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking", IEEE European Test Symposium, pp. 269-274, 2010.

[9] M. Taouil, S. Hamdioui, J. Verbree and E.J. Marinissen, "On maximizing the compound yield for 3D Wafer-to-Wafer stacked ICs", *IEEE International Test Conference (ITC)*, pp. 1-10, 2010.

[10] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computer*, vol 26, no. 5, pp. 26-35, 2009.

[11] M. Taouil, S. Hamdioui, K. Beenakker and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking", *IEEE Asian Test Symposium*, pp. 435-441, 2010.

[12] E.J. Marinissen, J. Verbree and M. Konijnenburg "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs", *IEEE VLSI test symposium (VTS)*, pp. 269-274, 2010.

[13] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference (ITC)*, pp.1-11, 2009.

[14] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas.", IEEE Transactions on Semiconductor Manufacturing, vol. 18, no. 1, pp. 136-139, 2005.

[15] M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Wiley-VCH, Weinheim, Germany, 2000.

[16] N. Miyakawa, "A 3D Prototyping Chip Based on a Wafer-level Stacking Technology", *Design Automation Conference (ASP-DAC)*, pp. 416-420, 2009

[17] E. Beyne, "3D Integration Crossing IC technology, Packaging and Design Barriers", *Semicon West 2008, TechXPOT, Test Assembly & Packaging*, http://www.semiconwest.org/cms/groups/public/documents/web_content/ctr_024376.pdf

On modeling and optimizing cost in 3D Stacked-ICs

Mottaqiallah Taouil[1] Said Hamdioui[1]

[1]Computer Engineering Lab
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, S.Hamdioui}@tudelft.nl

Erik Jan Marinissen[2]

[2]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
{erik.jan.marinissen}@imec.be

Abstract

3D-Stacked IC (3D-SIC) technology is one of the emerging technologies with many benefits such as higher performance and heterogeneous integration. During the manufacturing of such ICs, tests can be applied at different moments such as (a) before the stacking process, (b) after the creation of each partial stacked IC, (c) after the creation of the complete stack, and (d) after packaging of the stack. Moreover, each applied test may target interconnects, one or more dies, or even both. This results into a huge number of test flows, each with its own specific test cost. Choosing an efficient and appropriate test flow providing the required outgoing product quality (for a given design and manufacturing parameters) is extremely important in order to make 3D-SIC business profitable.

This paper discusses a tool for 3D-SIC test cost modeling; It gives the requirements and classifies them in design, manufacturing, test, packaging and logistics. It further covers user-cases and shows how the tool can be used at an early design stage in order to select the most efficient test flow for given input parameters (related either to manufacturing, test, packaging or logistics); hence, optimize the design and/or include the required DFT to support the selected test flow. The tool can be also used for sensitivity analysis where the impact of parameter changes on the test cost can be analyzed.

Keywords: *3D Test Flows, 3D Test Cost, 3D Manufacturing Cost, Through-Silicon-Via.*

I. Introduction

The popularity of 3D Stacked ICs (3D-SICs) is rising among industry and research institutes [1–8]. 3D-SICs are emerging as one of the main competitors to continue the trend of Moore's Law. Currently, a number of methods have been proposed to implement the interconnection of stacked dies. One of the most promising and perhaps the most reliable way to achieve this is with *Through Silicon Vias (TSVs)* [7]. TSVs are holes going through the chip silicon substrate filled with a conducting material. They enable short interconnections in 3D-SICs. Stacking dies using vertical interconnects have many benefits [8], including:

- Low latency interconnects between adjacent dies.
- Reduced power consumption.
- High bandwidth communication as TSVs cross dies along the surface of the chip
- Heterogeneous integration. Different dies in the stack could be manufactured by different wafer fabs, but also using different technologies. DRAM and logic integration in one 3D-SIC becomes feasible.
- Improved form factor and package volume density.

After complete manufacturing, each 3D-SIC has to be tested to guarantee the required quality and satisfy the number of defects per million (DPM) level. Moreover, since *every* die has to be tested before it is shipped, any tiny test cost reduction per IC will have significant impact on the the overall cost. Moreover, the number of test options (flows) increases exponentially with the stack size [9]. Therefore, finding the *optimal test flow* for 3D-SICs is very important. Modeling test cost and analyzing the impact of different test flows prior to manufacturing is becoming crucial; not only in order to simplify and optimize the Design for Testability (DfT) circuit, but also in order to reduce the overall 3D-SIC cost.

The published work on (test) cost modeling for 3D-SICs is very limited. In [10], the author considered a manufacturing cost model for 3D monolithic memory integrated circuits; the author models the cost improvement of 3D with respect to 2D for different 3D stack sizes. In [11], the authors developed a 3D-cost model to determine the optimal stack size for 3D-SICs given circuit, where they restricted the variable parameters to only die yield and area. In [12], the authors proposed a 3D cost model for

978-1-4673-0469-6/11/$25.00 ©2011 IEEE

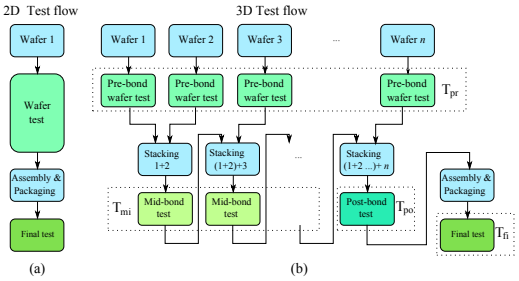


Fig. 1. 2D versus 3D D2W test flows

Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) stacking, where the different test flows were not considered. All of the published work clearly ignores the different possible test flows, and they do not consider the impact of such flows on the test cost and therefore on the overall cost. In our previous work [9], a basic cost model considering the impact of different test flows on the overall 3D-SIC cost was presented; however, the model have many restrictions such as (a) limited number of test flows, (b) the same test is applied to all dies, (c) constant fault coverage for both dies and interconnect, etc.

This paper presents an extension of our previous work [9], and proposes a tool being able to determine the test flow that results in lowest overall 3D-SIC cost. As the cost strongly depends on many parameters, the tool takes five classes of parameters as input; these are design, manufacturing, test, packaging and logistics. Moreover, the tool can calculate the total cost for each defined test flow for given inputs parameters; it can also be used for cost sensitivity analysis, i.e., how the overall cost is affected by changing a single parameter at a time. Critical parameters can be identified and optimized to reduce the overall cost further. This paper mainly focuses on the tool requirements and user-cases.

The remainder of this paper is organized as follows. Section II briefly overviews the concept of 3D test flows. Section III describes the requirements of the tool. Section IV presents the user-cases of the tool. Subsequently, Section V shows some preliminary simulation results. Finally, Section VI concludes this paper and highlights our ongoing work.

II. 3D Test Flows

For conventional testing of 2D ICs, two types of tests can be defined (as shown in Figure 1(a) [13]): a wafer test and a final test. A wafer test screens out faulty ICs prior to assembly and packaging in order to prevent unnecessary packaging costs, while a final test guarantees the quality of the packaged chip to reduce test escapes. A trade-off between the additional wafer test costs versus savings in packaging cost determines the applicability of this test. Furthermore, the test decision is based on the manufacturing yield and fault coverage. In case the yield is high enough, the test can be skipped or performed at low cost (i.e., low fault coverage).

For 3D SICs, additional tests -such as partial created stack tests- be defined. Figure 1(b) shows the natural test moments during the manufacturing of 3D-SICs. Four test moments can be distinguished in time, as depicted in Figure 1(b) and explained next.

1) T_{pr}: n pre-bond wafer tests, since there are n layers to be stacked. T_{pr} tests prevent faulty dies entering the stack. Two different types of test can be applied here. Traditional functionality of the chip can be tested for, but also preliminary TSV tests can be applied (in case of via-first [14]) as well.
2) T_{mi}: $n-2$ mid-bond tests applicable for partial created stacks. In this case, either dies, interconnects formed by the TSVs between them, a combination of the former two or none of them can be tested. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g., die thinning, and bonding [17].
3) T_{po}: one post-bond test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Here, both dies and interconnects between them can be tested for.
4) T_{fi}: one final test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied at this test moment as well.

Note that in total $2{\cdot}n$ different test moments can be identified versus 2 test moments for planar ICs. A 3D test flow can be defined as a combination of tests applied at the four test moments.

III. Tool requirements

The tool requirements identifies the parameters to-be-specified in order for the tool to process them and produce the necessary outputs. Obviously, in order to determine the most cost-effective test flow the test cost should be specified. However, this is by far not enough to produce a fair comparison of test flows. Other cost classes have to be specified as input requirements as they have a large impact on the overall cost as well. For example, a pre-bond TSV test requires additional hardware (which might not be reused after stacking), while it prevents faulty dies (due to defects in TSVs) to enter in the stack when the

defect is detected. The area increase (less dies per wafer) and additional pre-bond TSV test are justifiable if enough faulty TSVs are detected and corruption of more expensive (partial) stacks is prevented. Such trade-offs are interesting to investigate, but require an extensive model.

As the production of 3D-SICs requires design, manufacturing, test and packaging, all of these are considered as possible input parameters of the tool. Moreover, in oder to make a distinction between fab-less, fab-lite and IDM companies, an additional class of input parameters, referred to as logistic, is defined. For instance, a fab-less company may perform stacking and testing in different houses/countries, while IDM may perform all the required processing steps in a single house/location. Hence, logistics cost for fab-less companies is much higher than that of IDM.

In the rest of this section, each of the above tool requirement/input classes will be briefly discussed.

A. Design

Design for Testability (DfT) starts at the design phase to accommodate for tests at later stages (pre-bond, mid-bond, post-bond and final tests). Therefore, it is necessary to determine the impact of 3D test flows at this stage. For example, test of pre-bond TSVs using landing pads affect the chip layout and chip area negatively, while can detect some faulty TSVs prior to stacking using capacitance tests [14]. Similarly, mid-bond testing requires specific hardware to enable tests during this phase. These types of trade-off must be decided at design time as they impact the design and its associated cost.

B. Manufacturing

The manufacturing cost consists of the largest cost share for 3D-SICs, though it is strongly influenced by the applied test flow [9]. The Manufacturing class covers a wide range of parameters. The most obvious ones related to 3D are the bonding type and stacking operation. Each stacking operation is performed either in a Die-to-Die (D2D), D2W or W2W fashion, with the dies oriented in a Face-to-Face (F2F), Back-to-Face (B2F) or Back-to-Back (B2B) manner [8]. These stacking operations impact the cost and yield of the stack differently. In D2D and D2W stacking, Known Good Dies (KGD) can be stacked on each other to maximize the yield. This is not applicable in W2W stacking and therefore generally results in lower yield [15,16].

Heterogeneous integration allows dies of different technology to be stacked on each other. Therefore, dies with different wafer costs can be integrated together. For example, relative cheaper memory dies can be stacked on a more expensive logic layer. The manufacturing cost per individual die depends on the wafer cost and the number of dies on the wafer. In case additional hardware is integrated for DfT, the number of dies per wafer reduces and therefore increases the chip cost.

Manufacturing inherently induce defects in the interconnects and dies. Failures due to 3D processing steps are different in nature (improper TSV filling, defects introduced after to die thinning, etc. [17]). These defects can be modeled by additional yield parameters for the 3D processing. Thus, dies do not only have a yield for the wafer manufacturing phase, but also for the 3D stacking. The same applies for TSVs and the interconnects they form after stacking.

An new type of stacking recently introduced is the so called 2.5D FPGA of Xilinx, where four FPGAs are stacked on a passive silicon interposer layer for routing purposes [18]. Here, the 3D stacking is performed in an unnatural way. This concept can be generalized further into Multiple Tower (MP) stacking [19]. MP stacking involves stacking in which 3D-SICs consist of towers of different heights. Whether it is advantageous to manufacture such 3D-SICs is part of our work.

C. Packaging

After the 3D-SIC is manufactured and perhaps tested (a post-bond test), the 3D-IC is assembled and packaged. The cost attributed to packaging depend on the used materials and technology. We assume an independent cost for the packaging, i.e., it has no dependency with the other classes. Since all processing steps are defect-prone, a yield for the packaging can be considered as well.

D. Test

Testing 3D-SICs or parts of them can be performed at 4 phases as depicted in Figure 1(b). The applied tests could be: (a) pre-bond test, (b) mid-bond test, (c) post-bond test and (d) final test. A test consist of two parts, a test for the interconnects and dies. The vertical interconnects are new in the stack and testing them after stacking seems rational. Moreover, the testing gets more complicated as each tests could have different fault coverage and thus a different test cost attached to it.

A higher fault coverage requires usually more effort in test pattern generation at a higher test cost. However, additional hardware can be embedded in the design to simplify and reduce test time. Another possibility to reduce test time is by considering more advanced and expensive probe cards for pre-bond testing. Expensive probe cards might facilitate on board hardware (reduce chip area) and

reduce test time. Consider the following situation in which one of the 3 cases occurs.

1) In this case, no pre-bond testing is performed.
2) In this case, pre-bond testing is performed with a relative cheap probe card.
3) In this case, pre-bond testing is performed with a more expensive probe card.

In case 1, there is zero test cost and therefore zero fault coverage. Testing could be done for example in a later stage. In case 2, test time will be larger with respect to case 3 due to the inferior probe card. Both cases 2 and 3 will result in higher fault coverage with respect to case 1. To compare these cases fairly all parameters that define them should be considered.

The quality of the applied tests can be influenced by other companies as well. Depending on whether one or more companies are involved in the manufacturing of 3D-SICS, different test requirements can be set for the pre-bond wafer test quality [20]. If the wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place (e.g., for an IDM), the pre-bond test quality is subject to optimization. This means, there is not only the option to perform pre-bond testing or not, but also to perform pre-bond testing at a higher or lower test quality. Faulty undetected dies can be detected in a later stage, e.g., in higher quality final tests. Similarly, a high quality mid- or post-bond test (Known-Good-Stacks test) can be applied.

The type of test that is performed impacts the overall 3D-SIC cost. Therefore, specifying a test flow should be with full freedom, i.e., without restrictions on the test phase, testing of interconnects and/or dies, fault coverage etc. The complexity of the test flow, is based on the number of test moments. The number of test moments increase linearly with the stack size and therefore 3D-SICs could be probed several times. A drawback of this is additional probe damage due to over frequent testing. Hence, an upper limit to the number of touch downs per 3D-SIC might be required.

Another import parameter that affects cost in testing is the support for both parallel testing as well as serial testing of the dies in the stack. A memory consisting of multiple layers could be tested in parallel if each layer contains its own BIST engine. This reduces test time at the expense of increased area.

E. Logistics

An unobvious cost that impacts the 3D-SIC cost is related to the transport of wafers between companies, e.g., between the 3D fab and test house. Depending on the company type (fab-less, fab-lite or IDM), the cost picture for logistics is different. A fab-less company lacks manufacturing capabilities, and thus has to outsource all its manufacturing activities to foundries. In an IDM all production activities can be integrated and therefore are less subject to logistics cost. A fab-lite company outsources some of its activities while contains others in-house. Costs due to logistics for these types of companies depend on the outsourced activities.

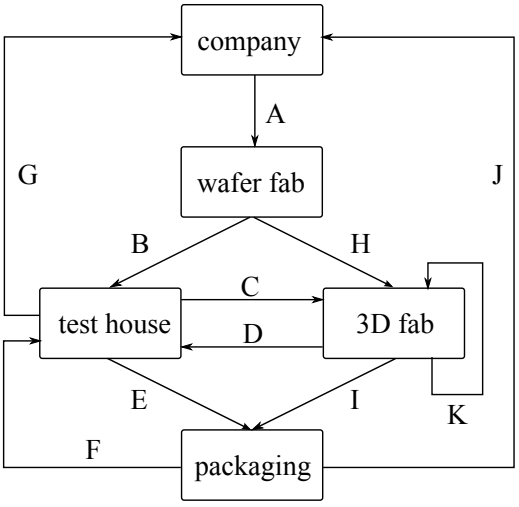


Fig. 2. Logistics cost for 3D-SIC

Figure 2 shows a diagram that considers the logistic cost considered in our tool. It presents the logistic cost for the worst case scenario in which each activity (manufacturing, test, 3D stacking and packaging) is separated from another. Each letter next to the arrow depicts the cost to move from one company to another one. It covers the following cost:

A: Cost between the *company* and *wafer fab*
B: Cost of moving tiers from *wafer fab* to *test house* (e.g., needed for pre-bond test)
C: Cost of moving tiers from *test house* to *3D fab* (e.g., to perform stacking after a pre-bond or mid-bond test)
D: Cost of moving tiers from *3D fab* to *test house* (e.g., for mid-bond or post-bond tests)
E: Cost of moving tiers from *test house* to *packaging fab* (e.g., after post-bond test)
F: Cost of moving tiers from *packaging fab* to *test house* (e.g., for final test)
G: Cost of moving tiers from *test house* to the *design company* (e.g., after final test)
H: Cost of moving tiers from *wafer fab* to *3D fab* (e.g.,

to perform stacking in case no pre-bond test is used)

I: Cost of moving tiers from *3D fab* to *packaging fab* (e.g., no post-bond test)

J: Cost of moving tiers from *packaging fab* to *company* (e.g., no final test)

K: Cost of moving tiers between 2 different *3D fabs* or possibly within the same 3D fab. (e.g. no mid-bond test)

The figure shows all possible costs related to transport of tiers. Depending on the test flow, some of the costs are not applicable. For example, in case pre-bond tests are skipped (arrow H), the cost associated with arrow B is inapplicable. Furthermore, for some companies some of these values are zero. For example, if a company performs both the testing and stacking in-house, costs associated with arrows C and D are minimal or zero.

IV. User-Cases

User-cases defines the different possible outputs of the tool. There are three main user-cases.

The primary case is to calculate the cost of each defined test flow, based on pre-defined input parameters; the latter are related to design, manufacturing, test, packaging and logistics. This allows the comparison of different test flows in order to identify the most effective flow.

The second case is the analysis of this cost by breaking it down into manufacturing, test, packaging and logistics costs. This analysis reveals the share of each cost.

The third user-case is related to sensitivity analysis; it identifies those parameters that have biggest impact on the overall cost. Thus, improving these parameters first results in largest cost reduction.

V. Case Study

In this section we briefly review some experimental results of simulating the test flows of Table I. First, we define 3D test flows, subsequently we show some results of our preliminary cost model.

The test flow framework can be extracted from the test moments of Figure 1. Depending on whether no or at least one test is performed at each possible test moment, we can distinguish 2^{2n} possible test flows. This number will further increase if we assume that tests at each moment may target different faults. For instance, if we assume that T_{mi} may test (1) one or more interconnects,(2) one or more dies, (3) a combination of (1) and (2), or (4) none, then the number of test flows will become 2^n (T_{pr}) $\times 4^{n-2}$ (T_{mi}) $\times 2$ (T_{po}) $\times 2$ (T_{fi}) $= 2^{3n-2}$. Hence, considering all 'theoretical' possible test flows will result in an unmanageable space. Realistic assumptions have to be made in order to create a clear overview (without loss of generality) [9]. For example, if we restrict mid-bond testing to be one of the following:

TABLE I. Test flow framework

Test flow	T_{pr}	T_{mi}	T_{po}
TF1	n	n	$i_a d_a$
TF2	n	i_t	$i_a d_t$
TF3	n	i_t	$i_t d_a$
TF4	n	$i_t d_t$	$i_t d_t$
TF5	y	n	$i_a d_a$
TF6	y	i_t	$i_a d_t$
TF7	y	i_t	$i_t d_a$
TF8	y	$i_t d_t$	$i_t d_t$

- Test for the *interconnect* between the top dies (i_t= top interconnect) only, .
- Test for the *top dies* (d_t= dies top) only,
- Test for both the *top interconnect* and *top dies* ($i_t d_t$), or
- none (n).

Then, T_{mi} will be reduced to $T_{mi} \in \{i_t, d_t, i_t d_t, n\}$.

In [9], reasonable assumption were made such that the test flows were reduced to eight for D2W stacking. They are given in Table I; e.g., TF1 denotes a test flow based on no pre-bond test (T_{pr}), no *mid-bond* (T_{mi}) and $T_{po} = i_a d_a$, where i_a denotes a test for *all interconnect* and d_a denotes a test for *all dies* of the 3D-SIC. Note that for each flow (a) a T_{pr} can be either applied ('y') or not ('n'), (b) $T_{mi} \in \{i_t, d_t, i_t d_t, n\}$ and (c) T_{pr} can be any of $\{i_t, d_t, i_a, d_a, \}$ or a combination of those as long as the test flow guarantees that each die and each interconnect in the stack is tested at least once [9].

In our preliminary model, only manufacturing, packaging and test inputs are considered. We will present the impact of test flows on the overall 3D-SICs for different stack yields. The die yield is based on the stacking process in [15], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. This work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50$ mm^2, the number of Gross Dies per Wafer (GDW) are estimated to be 1278 [21]. With the negative binomial formula for yield, a die yield of $Y_D = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [22]. For the stack size we assume a default stack size n=5. The stacking yield is composed of two parameters: the interconnect (TSV) yield Y_{INT} and the stacked-die yield Y_{SD}. For the good dies that enter the stack, a small probability exists that they get corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 95% as well.

Figure 3 depicts the overall 3D cost versus stacked yield (i.e., interconnect Y_{INT} and stacked-die Y_{SD}) for the test flows. In the figure, Y_{INT} and Y_{SD} are set to either 91% and 99%. The 3D cost of the flows are normalized

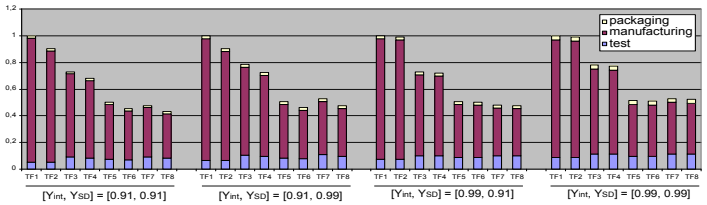


Fig. 3. Cost breakdown for variable die stack yield

to the cost of TF1 for each different stack yield. The figure shows that test flows with pre-bond tests (TF5 to TF8) significantly reduce the overall cost. In addition, TF6 and TF8 are the most cost-effective. If Y_{SD} is very high (i.e., 99%), then TF6 is the best as it tests only for interconnect. However, in case Y_{SD}=91%, TF8 performs better, since it tests for dies during the mid-bond phase. Therefore, it is able to prevent unnecessary stacking of dies in faulty partial stacks. Furthermore, the figure shows the breakdown of the 3D cost. The higher the stack yield, the higher the test and packaging shares. For example, for TF8 the test and packaging shares are 19% and 4% respectively for a stack yield [Y_{INT}, Y_{SD}] = [91%, 91%], while this increases to 21% and 6% for a stack yield of [Y_{INT}, Y_{SD}] = [99%, 99%].

VI. Conclusion and Future Work

In this paper the requirements and user-cases of a tool able to model and analyze the (test) cost of 3D-SIC is presented. The requirements were classified into: (a) design, (b) manufacturing, (c) packaging, (d) test and (e) logistics. Each class was described briefly and some of the design trade-offs were mentioned. The tool provides three main user-cases: (a) the calculation of the overall cost of 3D-SIC for each defined test flow, (b) cost breakdown, and (c) sensitivity analysis. The tool is wrapped with a Graphical User Interface (GUI) in order to simplify the parameter handling. Currently, the tool is under implementation and more experimental results will be presented in the future.

References

[1] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, vol. 94, no. 6, 2006.

[2] G. Loh et al. "Processor design in 3D die-stacking technologies", *IEEE Micro*, vol. 27, no. 3, pp. 31-48, 2007.

[3] K. Puttaswamy et al. "3D-Integrated SRAM Components for High-Performance Microprocessors", *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1369-1381, 2009.

[4] G. Loh et al. "Processor Design in 3D Die-Stacking Technologies", *IEEE Transactions on Computers*, vol. 27, no. 3, pp. 31-48, 2007.

[5] Y-F. Tsai et al. "Design Space Exploration for 3-D Cache", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 444-455, 2008.

[6] T. Thorolfsson et al. "Comparative analysis of two 3D integration implementations of a SAR processor", *IEEE International Conference on 3D System Integration*, pp. 1-4 Oct. 2009.

[7] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Desig Test on Computers*, vol. 22, no. 6, pp. 498-510, 2005.

[8] P. Garrou, Christopher Bower and Pater Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.

[9] M. Taouil, S. Hamdioui and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking", *Asian Test Symposium (ATS)*, pp. 435-441, 2010.

[10] A.J. Walker, "A Manufacturing Cost Model for 3-D Monolithic Memory Integrated Circuits", *IEEE Transactions on Semiconductor Manufacturing*, vol. 22, no. 2, pp. 268-275, 2009.

[11] P. Mercier, S.R. Singh, K. Iniewski, B. Moore and P. O'Shea, "Yield and Cost Modeling for 3D Chip Stack Technologies", *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 357-360, 2006.

[12] Y. Chen et al. "Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 471-476, 2010.

[13] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference (ITC)*, pp.1-11, 2009.

[14] P. Chen, C. Wu and D. Kwai, "On-Chip TSV testing for 3D IC before bonding using sense amplification", *Asian Test Symposium (ATS)*, pp. 450-455, 2009.

[15] J. Verbree, E.J. Marinissen, P. Roussel and D. Velenis, "On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking", *European Test Symposium (ETS)*, pp. 36-41, 2010.

[16] M. Taouil, S. Hamdioui, J. Verbree and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking ", *International Test Conference (ITC)*, pp. 1-10, 2010.

[17] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computers*, vol 25, no. 5, pp. 26-35, Oct. 2009.

[18] I. bolsens, "Entering the Era of 'Crossover' SoCs", *http://www.xilinx.com/innovation/research-labs/keynotes/Designcon-Keynote.pdf*

[19] C-C Chi et al., "DfT Architecture for 3D-SICs with Multiple Towers", *European Test Symposium (ETS)*, pp. 51-56, 2011.

[20] E. J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs", *Design, Automation and Test in Europe (DATE)*, pp. 1689-1694, 2010.

[21] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas", *IEEE Transactions on Semiconductor Manufacturing*, Vol 18, Issue 1, pp. 136-139, Feb. 2005.

[22] M. Bushnell and V. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Wiley-VCH, Weinheim, Germany, Aug. 2000.

Using 3D-COSTAR for 2.5D Test Cost Optimization

Mottaqiallah Taouil[1] Said Hamdioui[1]
[1]Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{m.taouil, s.hamdioui}@tudelft.nl

Erik Jan Marinissen[2]
[2]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
erik.jan.marinissen@imec.be

Sudipta Bhawmik[3]
[3]Qualcomm
5000 Somerset Corp. Blvd.
Bridgewater, NJ, USA
sbhawmik@qti.qualcomm.com

***Abstract*—Selecting an appropriate and efficient test flow for a 2.5D/3D Stacked IC (2.5D-SIC/3D-SIC) is crucial for overall cost optimization. This paper uses 3D-COSTAR, a tool that considers costs involved in the whole 2.5D/3D-SIC chain, including design, manufacturing, test, packaging and logistics, e.g. related to shipping wafers between a foundry and a test house; and provides the estimated overall cost for 2.5D/3D-SICs and its cost breakdown for a given input parameter set, e.g., test flows, die yield and stack yield. As a case study, the tool is used to evaluate the overall 2.5D-SIC cost for three test optimization problems: (a) the impact of the fault coverage of the pre-bond silicon interposer test, (b) the impact of pre-bond testing of active dies using either dedicated probe-pads or micro-bumps, and (c) the impact of mid-bond testing and logistics on the overall cost. The results show that for the selected parameters: (a) pre-bond testing of the interposer die is important for overall 2.5D-SIC cost reduction; the higher the fault coverage, the lower the overall cost, (b) using micro-bump probing results in much lower overall cost as compared to probe-pads, and (c) mid-bond testing can be avoided for high stacking yield.**

I. Introduction

Tremendous effort has been put in place to bring *through-silicon via (TSV)* based 2.5D and 3D-SIC technology closer to market [1–3]. Realizing such ICs is attractive due to major benefits [4] such as (a) increased electrical performance, (b) reduced power consumption due to shortened interconnects, (c) heterogeneous integration, (d) reduced form factor, etc.

One of the major challenges that has to be addressed in order to make 2.5D technology commercially successful is overall cost optimization. A 2.5D-SIC consists of two or more active dies stacked on a passive silicon interposer that forms the interconnection between the active dies and to the external world. As is the case for any IC, TSV-based 2.5D-SICs must be tested in order to guarantee the outgoing product quality and reliability. Hence, test cost is indispensable. Inherent to their manufacturing process, 2.5D-SICs provide several test moments such as before stacking, during manufacturing of partial stacked IC, after the complete manufactured stack, etc. This results into a large space of test flows; each with its own cost. Determining the optimal and most efficient test flow requires the analysis of all test flows, as different design and/or manufacturing parameters may impact the cost differently. Therefore, an appropriate cost model is required. The cost model should be able to evaluate the cost of each test flow, while considering all relevant incurred costs in the production chain of the 2.5-SIC.

Several cost models have been published in this area for 2.5D/3D-SICs. In [5], the author considered a manufacturing cost model for 3D monolithic memory integrated circuits; cost improvement of 3D with respect to 2D (for different 3D stack sizes) was modeled. In [6], the authors developed a 3D-cost model to determine the optimal stack size for a given 3D-SICs circuit, where they restricted the variable parameters to only die yield and area. In [7], the authors proposed a 3D cost model for Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) stacking. In [8], a detailed cost model of IMEC is presented; the paper primarily focuses on (a) the difference between cost integration for D2W and W2W stacking, (b) the impact of the number of TSVs and (c) the effectiveness of different 3D testing strategies in the pre-bond phase for D2W stacking. In [9], a 3D cost model is presented that focuses on modeling of metal layers and die area impact on 3D-cost integration for D2W and W2W integration. In [10], a 3D cost model is primarily developed to estimate the optimal tier count that leads to a minimal TSV count and subsequently partition the netlist into these tiers. In [11], the authors presented a cost estimation method for 2.5D ICs by extending their 3D floorplanning tool and 3D cost models; their models only include area and wire length, and do not consider testing at all. In [12], the authors proposed a cost model that emphasizes on manufacturing and test cost; the authors investigated the impact of D2W and W2W stacking on overall cost and determined the lower bound of the yield of the final package level test given the number of stacked dies and the final yield.

The state-of-the art described above clearly shows that none of the published cost models incorporated the impact of partial stack tests and different test flows. In our previous work [13], a basic cost model for D2W stacking considering the impact of limited test flows on the overall 3D-SIC cost was presented. However, this model suffers from many limitations such as (a) a lack of support for variable fault coverage (FC), (b) a restriction to a small set of test flows, (c) a focus on D2W stacking only, (d) no consideration of logistics cost, (e) no distinction between die and interconnect tests.In our work [14], we reported the requirements and user cases of a cost model tool that addressed most of the shortcomings of our work in [13]. In this paper, we build on our previous work in [14] to develop 3D-COSTAR; a tool that considers all costs involved in the whole 2.5D/3D-SIC production chain, including design, manufacturing, test, packaging and logistics (e.g. related to

978-1-4673-6484-3/13/$31.00 ©2013 IEEE

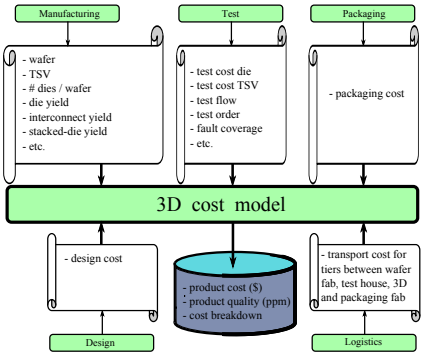


Fig. 1. 3D-COSTAR Organization.

shipping wafers between a foundry and a test house) in order to provide both the estimated overall cost for 2.5D/3D-SICs as well as its cost breakdown. More importantly, this paper analyses and reports about three case studies with respect to 2.5D-SIC test cost optimization; these are: (a) the impact of the FC of the interposer pre-bond test on the overall cost, (b) whether it is more advantageous to perform pre-bond testing for the active dies using dedicated probe pads or through micro-bumps, and (c) the impact of mid-bond testing and logistics on the overall cost.

The rest of this paper is organized as follows. Section II presents the architecture and flow of 3D-COSTAR respectively. Section III covers case studies where the test trade offs are described. Section IV presents the results of the experiments. Finally, Section V concludes the paper.

II. 3D-COSTAR

This section describes the architecture of 3D-COSTAR. First, the tool requirements are discussed followed by the use cases.

A. 3D-COSTAR Requirements and Cost Classes

In order to determine the most cost-effective test flow, the test requirements should be specified. However, taking only the test cost into consideration is not sufficient to provide a fair comparison; a test flow does not only impact test cost, but also design and manufacturing cost. For example, a pre-bond active die test with additional probe-pads increases die area and reduces the number of dies per wafer.

Figure 1 shows the general architecture of 3D-COSTAR, which can both evaluate 2.5D and 3D-SICs. The tool has five input classes which symbolize the costs involved in the whole 2.5D/3D-SIC production flow; these include design, manufacturing, test, logistics and packaging cost.

a) Design: Design for Testability (DfT) starts at the design phase to accommodate for tests at later stages (pre-bond, mid-bond, post-bond and final tests). For example, pre-bond testing of TSVs using probe pads affects the chip layout and chip area, while can detect some faulty TSVs prior to stacking [15]. Similarly, mid-bond testing requires dedicated hardware to support testing during this phase. These types of

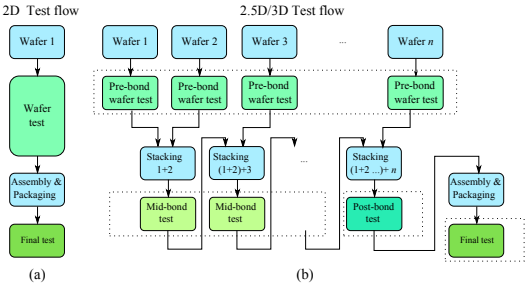


Fig. 2. 2D versus 2.5D/3D D2W test flows.

trade-off are strongly test flow dependent and must be decided at design time as they impact the design and its associated cost.

b) Manufacturing: Manufacturing requirements are related to the fabrication, processing of wafers and the stacking of tiers. As the manufacturing is not perfect, TSV yield, die yield, and stacking yield are required to accurately determine the cost. The manufacturing class covers a wide range of parameters and consists mainly of two parts: (a) manufacturing cost related to 2D IC and (b) cost related to 2.5D/3D stacking processing steps. The first part depends on the wafer cost, die yield, number of dies per wafer, cost of manufacturing steps, etc.; all of these results into a cost of a die per wafer. In case additional hardware is integrated for DfT, the number of dies per wafer reduces and therefore increases the chip cost. The second part depends on the cost of TSVs, wafer thinning, bonding (i.e., Die-to-Die (D2D), D2W and W2W), stacking process (i.e., Face-to-Face (F2F), Back-to-Face (B2F) or Back-to-Back (B2B)), interconnect yield, stacked-die yield, etc.; and it strongly depends on the applied test flow [13]. It is worth noting that the chosen bonding type and stacking process have a large impact on the cost and the yield of the 2.5D/3D-SIC; for instance, in D2D and D2W stacking, Known Good Dies (KGD) can be stacked on each other to maximize the yield. KGD stacking is is not applicable in W2W stacking and therefore generally results in lower yield [16,17]. For the 2.5D-SICs we assume dies to be stacked in a D2W F2F fashion. Moreover, as exact profiles of faults introduced during the stacking are not know/published yet, the tool is built such that it supports any defect distribution of dies during stacking.

c) Test: Figure 2(a) shows the conventional 2D test flow for planar wafers [18]; it consists of two test moments: a wafer test prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low as it prevents unnecessary assembly and packaging costs, while the final test is used to guarantee the final quality of the packaged chips. 2.5D/3D-SICs, however, provide additional test moments; e.g., additional test moments can be defined for each partial stack. Moreover, at each moment a distinction can be made between different tests such as die tests and interconnect tests. In general, four test moments can be distinguished for a 2.5D-SICs consisting of n dies as depicted in Figure 2(b):

(1) n *pre-bond* wafer tests, (2) n-2 *mid-bond* tests, (3) one *post-bond* test prior packaging and (4) one *final* test.
A *test flow* can be can be extracted from the above four defined test moments, which consist in total of $2n$ different moments. A test flow is as a collection of tests applied at these test moments. At each test moment, zero, one or more tests, possibly with different FCs, both for dies and/or interconnects, can be applied. Depending on the used test flow, the test cost might increase significantly. Therefore, skipping or reducing quality requirement at some test moments can restrain the test cost.
In addition, using advanced test equipment to reduce the test cost, parallel testing can be also used. Dies belonging to different layers can be tested in parallel if there is DFT support available for it. 3D-COSTAR does support the calculation of test cost for both simultaneous and serial testing of dies in a 2.5D or 3D-SIC.

The test cost can be company dependent as the quality of the applied tests could differ, e.g., for IDM and fab-less companies. For instance, depending on whether one or more companies are involved in the supply chain for the manufacturing of 2.5D/3D-SICs, different test requirements can be set for the pre-bond wafer test [19]. If the wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place (e.g., for an IDM), the pre-bond test quality is subject to optimization. Hence, at pre-bond test moment, we can not only perform or skip the pre-bond test, but we can also tune the quality of the applied test for cost optimization. Faulty undetected dies at this test moment can be detected in a later test moment, e.g., when applying a higher quality test in the final test moment. Similarly, a high quality mid- or post-bond test can be applied.

3D-COSTAR calculates test cost for any possible test combinations (test flow). Both the type of test and the used test flow impacts the overall 2.5D/3D-SIC cost. Therefore, specifying an optimized test flow should be with full freedom, i.e., without any restrictions on the test moment, on the used test (die, interconnect or both), neither on the FC, etc. The complexity of the test flow depends on the number of test moments, which increase linearly with the stack size. Hence, 2.5D/3D-SICs could be probed several times. However, having several touch-downs on the bottom wafer for testing purposes can damage the bonding-bumps. Therefore, setting an upper limit of maximal allowed touch-downs is practical.

d) Packaging: After the 2.5D/3D-SIC is manufactured and perhaps tested (a post-bond test), the 2.5D/3D-SIC is assembled and packaged. The cost attributed to packaging depends on the used materials and technology [20]. We assume an independent cost for the packaging, i.e., it has no dependency with the other classes. Since all processing steps are defect-prone, a yield for the packaging has to be considered as well. In this paper, we further ignore the packaging cost as it is irrelevant for the performed experiments.

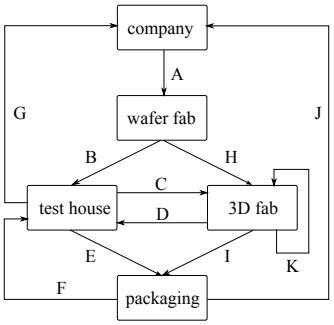


Fig. 3. Logistics cost for 2.5D/3D-SIC.

e) Logistics: The production of 2.5D/3D-SICs requires design, manufacturing, test and packaging costs. However, to make a distinction possible between fab-less, fab-lite and IDM companies, an additional set of hidden costs, referred to as logistics, is needed. For instance, a fab-less company may perform stacking and testing in different houses/countries, while IDM may perform all the required processing steps in a single house/location. Therefore, logistics costs are a direct consequence of moving dies and wafers between different locations. Figure 3 shows an overview of logistics costs considered in our tool. It presents all possible logistics costs for the worst case scenario in which each activity in the 2.5D/3D-SIC production chain can be outsourced; hence, the associated logistics costs have to be separated from each other. The figure assumes five companies/houses to be involved in the production chain: design company, wafer fab, 3D fab, test house and packaging house. A cost is associated to any moving activity of lots/wafers between any of these companies; for example, arrow B defines the cost for the logistics between wafer fab and test house. There are in total 11 possible costs.

It is worth noting that test flows have an impact on the logistics cost. Depending on the company type and test flow, some of the costs are not applicable. For example, in case pre-bond tests are skipped (arrow H), the cost associated with arrow B is inapplicable.

B. Use cases

Use cases define the functionality of the tool in terms of inputs and outputs. There are three main use cases.

1) *Overall cost calculation*. The primary goal of the tool is to calculate the overall cost of the production of 2.5D/3D-SICs for different test flows, based on pre-defined input parameters. The overall cost includes design, manufacturing, test, packaging and logistics cost.
2) *Cost breakdown*. The second use case is the analysis of the cost by breaking it down into design, manufacturing, test, packaging and logistics costs. This analysis reveals the share of each cost and provides insights about possible further cost optimization.

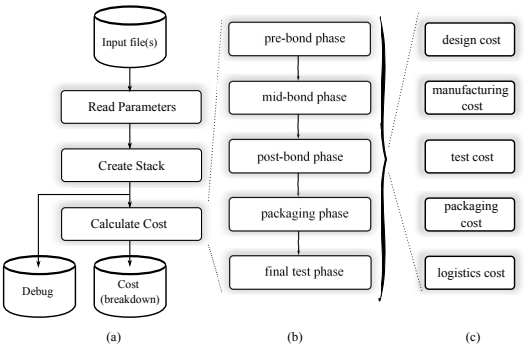


Fig. 4. Tool flow.

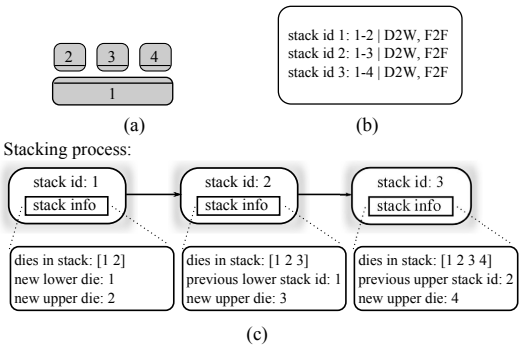


Fig. 5. Creating the stack.

3) *Sensitivity analysis*. The third use case is sensitivity analysis of input parameters; it identifies those parameters that have largest impact on the overall cost. Thus, tuning these parameters first results in largest cost reduction.

C. Tool Flow

Figure 4(a) presents a high-level overview of the tool flow. The tool starts by reading all input parameters from the input files and subsequently creating the stack. Thereafter, the cost is calculated by taking involved costs into consideration and moving through the IC production chain of the IC (see Figure 4(b)). At each step, the tool updates the impacted cost if applicable. For instance, if a mid-bond test is performed, then the test cost has to be updated. Reading the input parameters, creation of the stack and the cost calculation are the core steps of the tool. They are explained next.

Read parameters
The first stage of the tool reads the input parameters of each class. The parameters are specified by keywords and read from a file. For example, keywords that must be specified that are related to manufacturing are *die cost*, *die yield* etc.

Stack creation
Figure 5 shows an example of how the creation of a stack ta ly stored. Part (a) of the figure depicts a particular multiple tower stack IC. It consists of a bottom die/wafer labeled 1, a die labeled 2 stacked on die/wafer 1 using D2W stacking process with a F2F stacking orientation, followed by dies 3 and 4 in a similar manner. Part (b) of the figure, shows how this stack is defined. This particular stack consists of 3 stacking operations; each operation requires a specific stacking process and orientation. Figure 5(c) shows how the stack is internally stored. The stack is stored as an array of stacking operations. For example, after the first stacking operation (stack id: 1), the created stack consists of die 1 as a bottom/lower die and die 2 as an upper die. A debug file is created for verification.

Cost calculation
Given the input parameters, the different involved costs are calculated step by step by moving through the different phases shown in Figure 4. All costs are impacted by one of more of such phases. For example, pre-bond and mid-bond phases contribute to the manufacturing cost and requires DFT hardware (hence impacting the design cost as well), while these two phases together with post-bond phase and final phase contribute to test cost. The logistics cost strongly depends on the required number of movements of lots/wafers; e.g., between wafer fab, 3D stacking fab, test house, etc. The packaging cost is calculated based on the required packages for all the considered good stacked ICs (outgoing yield of the stack) after the post-bond test. The overall cost of 2.5D/3D-SIC is calculated by summing up all the cost of design, manufacturing, test, packaging and logistics.

Not all dies enter the stack. For instance, dies that are tested faulty in the pre-bond phase. To obtain the cost, the ratio of dies that enter the stack have to be calculated properly. We use Equation 1 to define the relation between test escapes TE, ingoing yield Y_{in} and outgoing yield Y_{out}; TE is the ratio of faulty dies that pass the test. The ingoing yield is the actual yield, the outgoing yield is the fraction of dies that is considered good after testing. Equation 2 [21] shows the relation between the test escapes, ingoing yield and the FC. By combining Equations 1 and 2 we obtain Equation 3, the outgoing yield as a function of the ingoing yield and FC.

$$TE = \frac{Y_{out} - Y_{in}}{Y_{out}} \quad (1)$$

$$TE(Y_{in}, FC) = 1 - Y_{in}^{1-FC} \quad (2)$$

$$Y_{out} = \frac{Y_{in}}{1 - TE} = \frac{Y_{in}}{Y_{in}^{1-FC}} = Y_{in}^{FC} \quad (3)$$

We assume that all these equations are valid for all yield operations involved in the manufacturing of the 2.5D-SIC; i.e, for the manufacturing of dies and interconnects. For instance, imagine that dies of type d_2 need to be stacked on the top of dies of type d_1 (see Figure 5); each die has its own yield and FC. If d_1 is total number of bottom dies, then the total number

of dies of type d_2 (say d_2) needed for the stacking will be:

$$d_2 = \frac{d_1 \cdot Y_{out,1}}{Y_{out,2}} \quad (4)$$

All cost and yield operations are based on the principle of updating partial or final stack yields. Consider the IC depicted in Figure 5(a). First, the outgoing yield of each die is calculated before stacking. Subsequently, the yields of the die in the stack are updated each time stacking a new die. Each time a new die enters an already existing partial stack, its quantity is determined by the combined outgoing yield of the dies in the stack. These steps are repeated until all dies are considered. The yields (pre-bond, mid-bond etc.) related to particular dies are tracked and stored individually. This allows us to detect faulty dies that escaped the pre-bond phase in a later stadium (mid-bond/final test). Once all partial and final stack yields are calculated, we can determine the number of dies, the number of tests and logistics for each individual die and partial/complete stack. To calculate the cost-price of a 2.5D-SIC, all costs involved in the production chain are attributed to good 2.5D-SICs only. For example, faulty detected dies in the pre-bond phase have also a manufacturing and test cost share in the overall cost.

III. Experimental Setup

This section describes the experiments performed in this work. Note that the yield and cost parameters considered for these experiments do not describe any processes at Qualcom, IMEC or partners, nor at TU Delft. The inputs of 3D-COSTAR are flexible and fully parameterized. By tuning these input parameters almost anything can be proven. Nevertheless, we provide inputs as realistic as possible. The experiments are performed for two types of applications; a mobile and FPGA application denoted by Case A and Case B respectively.

A. Reference Cases

This section describes the default parameter values of both applications. We assume that for both Case A and B the stack is composed out of four dies as depicted in Figure 5(a). For the mobile application, we assume the active dies to be heterogeneous (one big die and two smaller dies), while for the FPGA application all three active dies are identical. The parameters for both cases are summarized in Table I. In the table, Die 1 denotes the interposer and Dies 2,3 and 4 the active stacked dies.

First, we describe the parameters that are related to the pre-bond phase. As the interposer is passive (no FEOL processing) the wafer cost is much cheaper than the active dies which are usually implemented in the newest technology nodes. We assume standard 300mm diameter wafer with an edge clearance of 3mm, i.e., the effective radius equals 147mm. Wafers that contain interposer dies are assumed to cost 700$ only, while wafers with active dies cost 3000$.

For Case A (mobile application), we assume the passive interposer to be A=210mm^2, large enough to fit the active dies stacked on them. The big die is assumed to have an area of A=100mm^2, and the two smaller dies an area of A=50mm^2 each. For the given die areas and effective wafer radius, the number of gross dies per wafer (GDW) approximately equals to 293, 622, and 1283 [22] for the interposer, the large die, and the two smaller dies respectively. The defect density is considered to be d_0 = 0.5 defects/cm^2 for the silicon interposer (older technology and no FEOL) and 1.0 defect/cm^2 for the active dies, with both a defect clustering parameter $\alpha = 0.5$. The die yield can be estimated by the negative binomial formula as: $y = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$ [21]. This results into die yields of 56.80%, 57.74%, 70.71% for the interposer, the bigger die and two smaller dies respectively.

TABLE I
Default parameters Case A and Case B

	Case A			Case B	
Parameter	Die 1	Die 2	Die 3/4	Die 1	Die 2/3/4
Wafer costs ($)	700	3000	3000	700	3000
Effective wafer radius (mm)	147	147	147	147	147
Die Area (mm)	210	100	50	460	150
Dies per wafer	293	622	1283	125	411
Defect density (cm^{-2})	0.5	1	1	0.5	1
Die yield (%)	56.80	57.74	70.71	55.05	50.00
Pre-bond FC (%)	100	99	99	100	99
Pre-bond test cost ($)	0.20	1.00	0.50	0.40	1.50
Stacking cost ($)	0	0.05	0.05	0	0.05
Stacked die yield (%)	99.5	99	99	99.5	99
Interconnect yield (%)	-	99	99	-	99
Mid-/post-bond FC (%)	0	0	0	0	0
Mid-/post-bond test cost ($)	0	0	0	0	0
Final FC (%)	100	99	99	100	99
Final test cost ($)	0.05	1.00	0.50	0.10	1.50

For Case B (FPGA application), the area of the three active dies is assumed to be A=150mm^2, while the interposer has an area of A=460mm^2. Using the same GDW algorithm and negative binomial yield formula the number of dies per wafer yields 411 and 125 with a yield of 55.05 and 50.00% for the interposer and three active dies respectively.

Further we assume a 100% FC for the interposer at a cost of 0.20$ for Case A and 0.40 $ for Case B. For the active dies we assume a test cost of 0.50$ for the smallest dies of 50mm^2, 1.00$ for the dies of 100mm^2 and 1.50$ for the dies of 150mm^2. In all cases, the FC for active dies is assumed to be 99%.

The next group of variables in the table contain parameters related to the mid-bond and post-bond. For both Case A and Case B we assume these parameters to be the same. Each time an active die is stacked on an interposer, the stacked-die yield (stack pass yield) of the active die is assumed to be 99%, while the stacked-die yield of the interposer is assumed to be 99.50%. The yield of the interconnects is assumed to be 99% (which includes the micro-bumps) between each pair of stacked dies, i.e., between Die 2, Die 3 or Die 4 and Die 1. Note that there are 3 stacking operations and 3 sets of interconnects. We assume no mid-bond testing for dies as well as interconnects for the reference cases.

In the final phase, we assume the same test costs for the active dies as in the pre-bond phase. However, we assume that

TABLE II
PROBE-PADS VERSUS MICRO-BUMPS FOR CASE A

	Probe-pads			Micro-bumps		
Parameter	Die 1	Die 2	Die 3/4	Die 1	Die 2	Die 3/4
Die Area (mm)	210	101	51	210	100	50
Dies per wafer	293	618	1254	293	622	1283
Die yield (%)	56.80	57.54	70.36	56.80	57.74	70.71
Pre-bond FC (%)	100	99	99	100	99	99
Pre-bond test cost ($)	0.20	10.00	5.00	0.40	1.05	0.55
Interconnect yield (%)	-	99	99	-	98	98
Final FC (%)	100	99	99	100	99	99
Final test cost ($)	0.05	1.00	0.50	0.10	1.00	0.50

testing the interposer (Die 1) is less expensive, as they can be tested by an EXTEST [23]. For Case A this cost is assumed to be only 0.05$, while for Case B 0.10$. Note that the test cost for interconnects (including micro-bumps) is not mentioned in the table as they are tested through the interposer die. We assume a 100% default FC for interconnects.

B. Experiments

The values presented in the previous section form the default parameters of each experiment. We explain the experiments in more depth and examine the relevant parameters for each case study. The three experiments are as follows.

1) Impact of the FC of pre-bond test of the interposer.
2) The use of probe-pads versus micro-bump probing.
3) Impact of mid-bond testing and logistics.

The experiments are described in the next sections, and apply both to Case A and Case B. Note that these experiments are only a small subset of what 3D-COSTAR can do.

Impact of the FC of interposer pre-bond testing: In this experiment, the impact of pre-bond testing for the passive silicon interposer is examined. The experiment considers a variable FC for the interposer test, i.e, between 0% and 100%. Similarly, we assume the test cost to scale linearly in the range between 0.00$ and 0.20$ for Case A, e.g., if the FC is 50% then the value of the interposer test cost is 0.10$. The reason for the linear relation is because the interposer consist of wires only.

The remainder parameters are considered to be the same as the reference case described in Table I. For Case B, the relation between test cost and FC for the interposer is applied in a similar manner (0.40$ for 100% FC).

Probe-pads versus Micro-Bump probing: In this second experiment, we investigate the trade-off between pre-bond tests probing dedicated pads and micro-bumps [24] for the active dies. As the active dies have no I/O pads, testing these dies in the pre-bond phase should be performed by one of the two methods. Table II and III show the parameter values that changed with respect to the reference case for Case A and Case B respectively. The extra probe-pads for pre-bond testing occupy additional area and this has to be accounted for. We consider for example the wide-IO memory [25] where 1200 micro-bumps are placed and assume only 10% of these microbumps (i.e. 120) have dedicated probe-pads of size $80\mu m$ by $80\mu m$. Note that if more probe-pads are considered,

TABLE III
PROBE-PADS VERSUS MICRO-BUMPS FOR CASE B

	Probe-pads		Micro-bumps	
Parameter	Die 1	Die 2/3/4	Die 1	Die 2/3/4
Die Area (mm)	460	151	460	150
Dies per wafer	124	404	124	406
Die yield (%)	55.05	49.88	55.05	50.00
Pre-bond FC (%)	100	99	100	99
Pre-bond test cost ($)	0.40	15.00	0.40	1.55
Interconnect yield (%)	-	99	-	98
Final FC (%)	100	99	100	99
Final test cost ($)	0.10	1.50	0.10	1.50

it will increase the die area. We can estimate the area of these 120 dedicated pads to add an extra area of $120{\cdot}80\mu$m$\cdot 80\mu$m $\approx$1mm^2. This extra die area impacts the die yield and GDW as shown in the second columns of the tables. For example, for the bigger die of Case A the number of dies that decreases from 622 to 618 if dedicated pads are added, while the yield reduces from 57.74% to 57.54%. Similarly, the table shows the numbers for the other dies. Moreover, as there only 10% of the micro-bumps are used as probe-pads, the pre-bond test cost of the active dies is assumed to be 10 times more expensive as compared to that of the reference case.

For the micro-bump probing we assume that a micro-bump probe-card cost 50k$ for 1 million touch downs. This results into an additional test cost of 0.05$ for each active die in the pre-bond phase. Moreover, as micro-bump touchdowns can cause later defects in the interconnections, we assume the interconnect yield to be 1% less (i.e., 98% instead of 99%) as compared to the case were extra probe-pads are used.

Impact of Mid-Bond Testing and Logistics: In order to investigate the impact of mid-bond testing and logistic cost we consider the following three sub-cases for both Case A and Case B:

1) No mid-bond testing and no logistic cost (reference case)
2) Mid-bond testing and no logistic cost.
3) Mid-bond testing and logistic cost.

We assume the FC and test cost for the mid-bond tests to be same as their values in the final test. For the logistics, we attribute costs to applicable arrows of Figure 3. We assume the cost of moving a single wafer (independent of the number of dies stacked on it) per arrow to be in the range of 1% up to 10% of the wafer manufacturing cost.

IV. SIMULATION RESULTS

This section describes the results of the three experiments.

A. Experiment 1

Figure 6 shows the impact of variable pre-bond interposer FC on the overall 2.5D-SIC cost for Case A and B. The results clearly show that performing a high quality interposer pre-bond tests realizes a significant relative cost reduction; the higher the FC the higher the cost reduction. Moreover, the results reveals that the larger the dies the higher the relative cost reduction; for instance, in Case B (with larger dies) the relative cost reduction is about 52%.

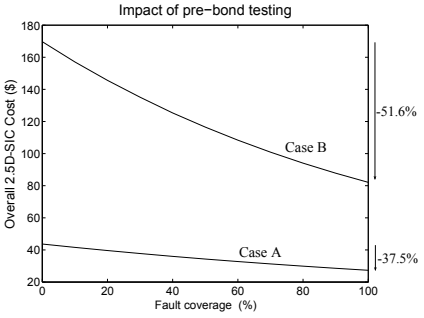


Fig. 6. Impact of pre-bond interposer FC.

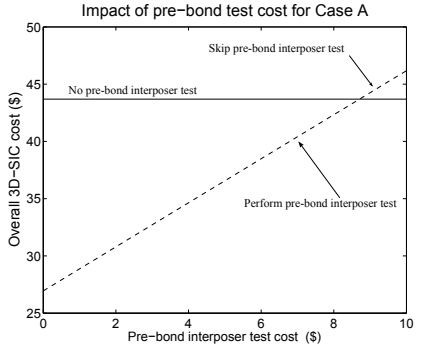


Fig. 7. Break-event cost point for Case A

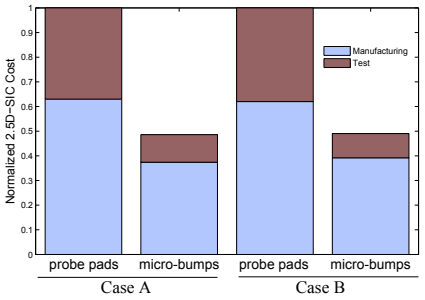


Fig. 8. Dedicated probe pads vs micro-bump probing.

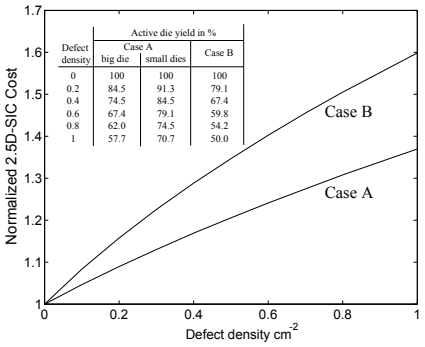


Fig. 9. Impact of defect density on Experiment 2.

However, testing of interposers is still a major challenge; cheap and efficient DFTs are still missing. Therefore, it is worth to analyze (for the given parameters) the break-even cost point where testing the interposer leads to the same overall cost as where no test is performed. This trade-off is depicted in Figure 7 for Case A. The figure contains two lines; the horizontal solid line shows the overall cost in case no pre-bond testing is performed and the dashed rising line shows the overall cost for variable pre-bond interposer cost for 100% FC. Note that for this particular case, the break-even point is around 8.50$. Hence, it is worth to use pre-bond test with maximal FC only if the associated test cost is below this threshold. Similar analysis has done for Case B; the break-even point found to be around 33.00$.

B. Experiment 2

The second experiment considers the analysis of test trade-off between dedicated probe pads and micro-bump probing. Figure 8 reports the results of such analysis; it shows the normalized 2.5D-SIC costs for both cases. Irrespective of the case, additional pads for testing result in higher overall cost, mainly due to test cost increase (as the cost break down shows), but also due to a slight yield loss (extra area of the pads). Moreover, the results show that the expensive probe-cards seems to pay off. The cost break down shows that largest share of costs are due to manufacturing of dies (around 80% in case micro-bumps and around 65% in case probe/extra pads). Note that the difference in the overall cost between using micro-bumps and probe pads is about 50%. This cannot justified with the difference in pre-bond test cost only; there are hidden costs primarily due to the faulty dies thrown away in the pre-bond phase. Therefore, it is important to analyze this behavior for different die yields. We performed a sensitivity analysis for the defect density. Figure 9 shows the results for both cases. As the defect density increases (i.e., the die yield reduces) the overall cost increases. The die yields that correspond to the defect density values are depicted in the table at the top left of figure. The impact of the die yield is more severe for Case B as the dies are larger and more expensive.

C. Experiment 3

Figures 10 and 11 shows the relative cost increase if mid-bond testing and/or logistics are considered; the results are given for various stacked-die Y_{SD} and interconnect yield Y_{INT}.

The figures contain four planes; the non-labeled planar planes shows the normalized base-line (i.e., no mid-bond test and no logistic cost), while the other labeled planes describe the results of the cases where mid-bond testing is performed; Planes 1, 2 and 3 show the impact of the logistics cost when assuming such cost to be 0%, 1% and 10% of the wafer cost respectively. From the figures we conclude the following:

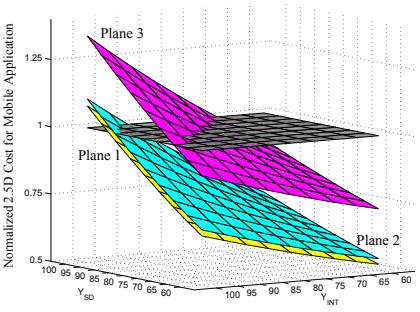


Fig. 10. Impact of mid-bond testing for Case A.

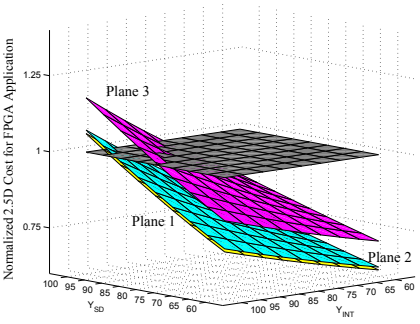


Fig. 11. Impact of mid-bond testing for Case B.

- Irrespective of logistic costs, mid-bond testing can be avoided if the stacked-die yield and the interconnect yield are high; in our case study higher than $> 90\%$. It is worth noting that the simulation has been done while considering an intensive pre-bond test both for interposer (100%) and active dies (99%).
- Logistics cost has a minor impact on the overall cost if they are low. However, they can substantially increase the overall cost if they are high (e.g. 10% of the wafer cost).

The results of all the experiments clearly show that optimizing the overall/test cost is a complex task; it strongly depends on the test flow, FC of each test, different yield components, etc. Therefore using a tool such as 3D-COSTARis extremely important to make appropriate trade-offs at an early stage in the design and optimize overall cost.

V. Conclusion

In this paper, 3D-COSTAR was introduced and used to evaluate different test flows and strategies for 2.5D-SICs; the tool considers all costs involved in the production (including design, manufacturing, testing, packaging and logistic) and produces the overall cost as well as the cost breakdown.

The case studies presented in the paper showed the significant importance of using such a tool in order to make appropriate trade-offs for overall cost optimization. For example, the simulation results showed that when appropriate test strategies (test flow and FC) are used for given design and manufacturing parameters, the overall cost can be reduced. Pre-bond testing of the interposer die is important for overall 2.5D-SIC cost reduction; using micro-bump probing results in much lower overall cost as compared to probe-pads; mid-bond testing can be avoided for high stacking yield.

References

[1] C. Zinck, "3D Integration Infrastructure Amp; Market Status," in *3D Systems Integration Conference*, nov. 2010, pp. 1–34.
[2] M.J. Wang *et al.*, "TSV Technology for 2.5D IC Solution," in *Electronic Components and Tech. Conf.*, june 2012, pp. 284–288.
[3] E.J. Marinissen, "Challenges and Emerging Solutions in Testing TSV-based $2\frac{1}{2}$D- and 3D-Stacked ICs," in *Design, Automation Test in Europe Conference Exhibition*, march 2012, pp. 1277–1282.
[4] *Handbook of 3D Integration*.
[5] A. Walker, "A Manufacturing Cost Model for 3-D Monolithic Memory Integrated Circuits," *IEEE Trans. on Semiconductor Manufacturing*, vol. 22, pp. 268–275, may 2009.
[6] P. Mercier *et al.*, "Yield and Cost Modeling for 3D Chip Stack Technologies," in *Custom Integrated Circuits Conference, 2006*, sept. 2006, pp. 357–360.
[7] Y. Chen *et al.*, "Cost-Effective Integration of Three-Dimensional (3D) ICs Emphasizing Testing Cost Analysis," in *IEEE/ACM Int. Conf. on Computer-Aided Design*, nov. 2010, pp. 471–476.
[8] D. Velenis *et al.*, "Impact of 3D Design Choices on Manufacturing Cost," in *IEEE Int. Conf. on 3D System Integration*, sept. 2009, pp. 1–5.
[9] X. Dong and Y. Xie, "System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs)," in *Asia and South Pacific Design Automation Conference*, jan. 2009, pp. 234–241.
[10] C.C. Chan, Y.T. Yu, and I.R. Jiang, "3DICE: 3D IC Cost Evaluation Based on Fast Tier Number Estimation," in *12th Int. Symp. on Quality Electronic Design*, march 2011, pp. 1–6.
[11] C. Zhang and G. Sun, "Fabrication Cost Analysis for 2D, 2.5D, and 3D IC Designs," in *IEEE International 3D Systems Integration Conference*, feb. 2012, pp. 1–4.
[12] Y.W. Chou *et al.*, "Cost Modeling and Analysis for Interposer-Based Three-Dimensional IC," in *IEEE 30th VLSI Test Symposium*, april 2012, pp. 108–113.
[13] M. Taouil *et al.*, "Test Cost Analysis for 3D Die-to-Wafer Stacking," in *19th IEEE Asian Test Symposium*, dec. 2010, pp. 435–441.
[14] M. Taouil, S. Hamdioui, and E.J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," in *IEEE 6th International Design and Test Workshop*, dec. 2011, pp. 24–29.
[15] P.Y. Chen, C.W. Wu, and D.M. Kwai, "On-Chip TSV Testing for 3D IC before Bonding Using Sense Amplification," in *Asian Test Symposium*, nov. 2009, pp. 450–455.
[16] J. Verbree *et al.*, "On The Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking," in *15th IEEE European Test Symposium*, may 2010, pp. 36–41.
[17] M. Taouil *et al.*, "On Maximizing The Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *IEEE International Test Conference*, nov. 2010, pp. 1–10.
[18] E.J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias," in *International Test Conference*, nov. 2009, pp. 1–11.
[19] E.J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs," in *Design, Automation Test in Europe Conference Exhibition*, march 2010, pp. 1689–1694.
[20] R. Tummala, *Fundamentals of Microsystems Packaging*.
[21] V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*, ser. Frontiers in Electronic Testing.
[22] D. de Vries, "Investigation of Gross Die per Wafer Formulas," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 18, no. 1, pp. 136–139, feb. 2005.
[23] (2013) IEEE p1838. [Online]. Available: http://grouper.ieee.org/groups/3Dtest/
[24] K. Smith *et al.*, "Evaluation of TSV and Micro-Bump Probing for Wide I/O Testing," in *IEEE International Test Conference*, sept. 2011, pp. 1–10.
[25] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 111–119, jan. 2010.

Impact of Mid-Bond Testing in 3D Stacked ICs

Mottaqiallah Taouil[1] Said Hamdioui[1]
[1]Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, S.Hamdioui}@tudelft.nl

Erik Jan Marinissen[2]
[2]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
erik.jan.marinissen@imec.be

Sudipta Bhawmik[3]
[3]Qualcomm
5000 Somerset Corp. Blvd.
Bridgewater, NJ, USA
sbhawmik@qti.qualcomm.com

***Abstract*— In contrast to planar ICs, during the manufacturing of three-dimensional stacked ICs (3D-SICs) several tests such as pre-bond, mid-bond, post-bond and final tests can be applied. This in turn results into a huge number of test flows/strategies. Selecting appropriate and efficient test flow (for given design and manufacturing parameters such as stack size, die yield, stack yield, etc) is crucial for overall cost optimization. To evaluate the test flows, a case study is performed in which 3D-COSTAR is used to compare the overall cost of producing a 3D-SIC using variable fault coverage during the mid-bond tests. In addition, we investigate the impact of the logistics cost for various test flows. The impact of logistics costs depend on the outsourced processing steps during the manufacturing. Simulation results show, for our parameters, that by choosing an appropriate test flow the overall 3D-SIC cost for appropriate fault coverages can reduce the overall cost up to 20% for a 5-layered 3D-SIC with die yields of 90%.**

Keywords: *3D integration, cost modeling, test cost, test flows.*

I. Introduction

Tremendous effort has been put in place to bring *Through Silicon Via (TSV)* based 3D-SIC technology closer to market [1–3]. Realizing such ICs is attractive due to major benefits [4] such as (a) increased electrical performance, (b) reduced power consumption due to shortened interconnects, (c) heterogeneous integration supporting optimized logic, memory, RF, MEMs etc., and (d) reduced form factor, etc. The mentioned benefits therefore drive the production of a new generation of 3D chips.

One of the major challenges that has to be addressed in order to make this technology commercially successful is testing. As is the case for any IC, TSV-based 3D-SICs must be tested in order to guarantee the outgoing product quality and reliability. Therefore, making test cost an indispensable part. Inherent to their manufacturing process, 3D-SICs provide several test moments such as before stacking, during manufacturing of partial stacked IC, after the complete manufactured stack, etc. This results into a huge space of test flows; each with its own cost. Determining the optimal and most efficient test flow requires analysis of all test flows, as different design and/or manufacturing parameters may impact the cost differently. Therefore, an appropriate cost model is required.

In this paper, we use 3D-COSTAR to evaluate 3D test flows [5,6]. In [5], we presented a preliminary version of our tool that had many limitations, such as lack of support for variable fault coverage, logistics cost etc. These limitations have been addressed in [6]. The tool is based on a cost model considering all costs involved in the 3D-SIC production chain including design, manufacturing, test, packaging and logistics; the logistics costs are due to transport of wafers and dies between different companies during the 3D-SIC production chain. As a case study, the tool is used to evaluate different test flows for 3D-SICs primarily focusing on variable fault coverage during pre- and mid-bond testing. Note that mid-bond testing (partial stack testing) could impact logistics cost, as tiers have to be transported to testers. The main contribution of this paper are as follows.

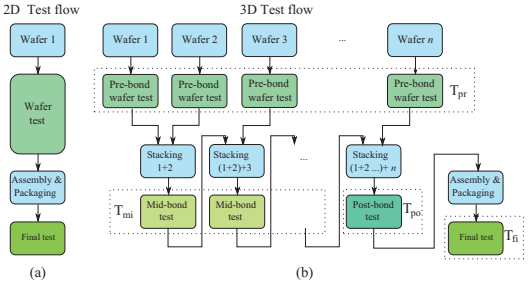


Fig. 1. 2D versus 3D D2W test flows.

- To our best knowledge, we are the first to experiment with test flow analysis for 3D-SICs with variable fault coverage during pre-bond and mid-bond testing.
- We investigate and analyze the impact of two logistics models on the overall 3D-SIC cost.

The rest of this paper is organized as follows. Section II provides the background of this paper; it discusses the difference between 2D and 3D test flows and briefly explains 3D-COSTAR. Section IV analyzes the impact of variable fault coverage on the overall 3D-SIC cost. Subsequently, Section V analyzes the impact of logistic costs. Finally, Section VI concludes the paper.

II. Background

A. 2D versus 3D Testing

Figure 1(a) shows the conventional 2D test flow for planar wafers [7]; it consists of two test moments: a wafer test prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low as it prevents unnecessary assembly and packaging costs, while the final test is used to guarantee the final quality of the packaged chips. 3D-SICs, however, provide additional test moments;

978-1-4799-1585-9/13/$31.00 ©2013 IEEE

e.g., additional test moments can be defined for each partial stack. Moreover, at each moment a distinction can be made between different tests such as die tests and interconnect tests. In general, four test moments can be distinguished for 3D-SICs as it is depicted in Figure 1(b); they are explained next.

1) T_{pr}: n *pre-bond* wafer tests, since there are n layers to be stacked. T_{pr} tests prevent faulty dies entering the stack. Besides die test, preliminary TSV interconnect tests can be applied. Several research work already exists regarding this topics; e.g., in [8] the authors use a capacitance test to detect some of the faulty TSVs and in [9] the authors propose active probing to detect faulty TSVs.
2) T_{mi}: n-2 *mid-bond* tests applicable for partial created stacks. In this case, either the dies, the interconnects, their combinations or none of them can be tested. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g., die thinning, and bonding [10].
3) T_{po}: one *post-bond* test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Both interconnects and dies can be tested.
4) T_{fi}: one *final* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied at this test moment as well.

A *test flow* can be can be extracted from the above four defined test moments, which consist in total of $2n$ different moments. A test flow is as a collection of tests applied at these test moments. At each test moment, zero, one or more tests, possibly with different fault coverages, both for dies and/or interconnects, can be applied. Depending on the used test flow, the test cost might increase significantly. Therefore, skipping or reducing quality requirement at some test moments can restrain the test cost.

B. 3D-COSTAR

This section describes the high level architecture of 3D-COSTAR. In order to determine the most cost-effective test flow, the test requirements should be specified. However, taking only the test cost into consideration is not sufficient to provide a fair comparison between the different test flow. This is because a test flow does not only impact test cost, but also manufacturing cost and even design cost.

As already mentioned, a pre-bond TSV test requires additional DFT hardware (which might not be reused after stacking), while it prevents faulty dies (due to defects in TSVs) to enter in the stack if detected. As a consequence, the die area increases (less dies per wafer).

Figure 2 shows the general architecture of 3D-COSTAR. The tool has five classes of inputs which reflect the cost involved in the whole 3D-SIC production; these include design cost, manufacturing cost, test cost, logistics cost and packaging cost. We briefly review the requirements associated with each input class.

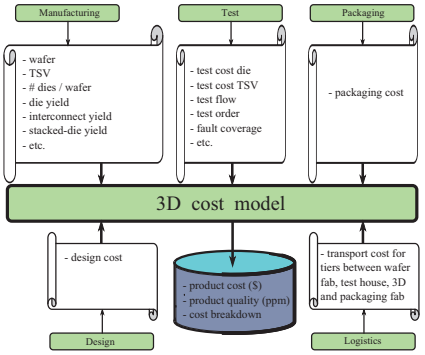


Fig. 2. 3D-COSTAR Organization.

a) Design: Design for Testability (DfT) starts at the design phase to accommodate for tests at later stages (pre-bond, mid-bond, post-bond and final tests). Therefore, it is necessary to determine the impact of 3D test flows at this stage. For example, pre-bond testing of TSVs using landing pads affects the chip layout and chip area, while can detect some faulty TSVs prior to stacking using capacitance tests [8]. Similarly, mid-bond testing requires dedicated hardware to support testing during this phase. These types of trade-off are strongly test flow dependent and must be decided at design time as they impact the design and its associated cost.

b) Manufacturing: Manufacturing requirements are related to the fabrication, processing of wafers and the stacking of tiers. The first part depends on the wafer cost, die yield, number of dies per wafer, cost of manufacturing steps, etc.; all of these results into a cost of a die per wafer. In case additional hardware is integrated for DfT, the number of dies per wafer reduces and therefore increases the chip cost. The second part depends on the cost of TSVs, wafer thinning, bonding (i.e., Die-to-Die, Die-to-Wafer and Wafer-to-Wafer), stacking process (i.e., Face-to-Face (F2F), Back-to-Face (B2F) or Back-to-Back (B2B)), interconnect yield, stacked-die yield, etc.; and it strongly depends on the applied test flow [5]. It is worth noting that the chosen bonding type and stacking process have a large impact on the cost and the yield of the 3D-SIC; for instance, in D2D and D2W stacking, Known Good Dies (KGD) can be stacked on each other to maximize the yield. This is not applicable in W2W stacking and therefore generally results in lower yield [11,12]. Moreover, as the exact profile of faults introduced during the 3D stacking is not know/published yet, the tool is built such that it supports any fault distribution during the stacking.

c) Test: The test class defines the test flows as defined in II-A. We will slightly redefine a test flow; a test flow defines what to test (dies or interconnect) and when to test them. A test flow consists of the following attributes:

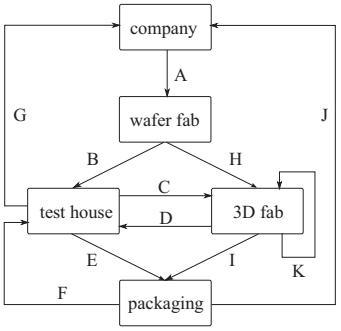


Fig. 3. Logistics cost for 3D-SIC.

- test moments: for each test phase (pre-, mid-, post-bond and final) test you can apply or skip tests for all dies.
- test contents: each time a test is performed the user can specify whether TSVs (restricted to pre-bond only), interconnects or dies are tested. In addition, the user also must define the quality of the tests in terms of fault coverage for each sub-test.
- test order: the test order tell us for each phase the order the sub-tests for dies and interconnects are performed.

In this work, interconnects are assumed to be tested prior dies (in case both are tested for); therefore, if a fault is detected in the interconnects then there is no need to test the dies as the 3D-SIC will is faulty. The reason to test interconnects first is because it is assumed to be cheaper as compared to die tests and vertical interconnects must be working properly in order to access the upper layer(s).

d) Packaging: After the 3D-SIC is manufactured and perhaps tested (a post-bond test), the 3D-SIC is assembled and packaged. The cost attributed to packaging depends on the used materials and technology [13]. We assume an independent cost for the packaging, i.e., it has no dependency with the other classes. Since all processing steps are defect-prone, a yield for the packaging can be considered as well.

e) Logistics: The production of 3D-SICs requires design, manufacturing, test and packaging costs and all of these must be considered as possible input parameters of the tool as depicted in the figure. However, to make a distinction possible between fab-less, fab-lite and IDM companies, an additional set of requirements, referred to as logistic, are needed. For instance, a fab-less company may perform stacking and testing in different houses/countries, while IDM may perform all the required processing steps in a single house/location. Therefore, logistics costs are a direct consequence of moving dies and wafers between different locations. For example, between the wafer fab, test house and 3D stacking fab.

Figure 3 shows an overview of logistics costs considered in our tool. It presents all possible logistics costs for the worst case scenario in which each activity in the 3D-SIC production chain can be outsourced; hence, the associated logistics costs have to be separated from each other. The figure assumes five companies/houses to be involved in the production chain: design company, wafer fab, 3D fab, test house and packaging house. A cost is associated to any moving activity of lots/wafers between any of these companies and is denoted by an arrow with a letter. There are in total 11 possible costs; they are explained next. It is worth noting that test flows have a large impact on the logistics cost. Depending on the test flow, some of the costs are not applicable. For example, in case pre-bond tests are skipped (arrow H), the cost associated with arrow B is inapplicable. Furthermore, depending on the type of the company producing 3D-SICs, some of these values can be not applicable or equal to zero. For example, if a company performs both testing and stacking in-house, costs associated with arrows C and D are zero.

III. Reference process

This section discusses the most relevant parameters for each input class that are used in our experiments.

Manufacturing cost: Manufacturing cost consists of cost related to wafer/die, cost related to TSVs and cost related to stacking process.

Wafer/die cost depends on several parameters, e.g., stack size, die yield, number of dies per wafer, stacking yield, interconnect yield, etc. We consider a stack size n=5 where dies are stacked in a D2W fashion, in which the dies are identical in terms of yield and cost. The yield of the dies is based on the reference process in [11], where a standard 300 mm diameter wafer is used with an edge clearance of 3 mm. This work assumes a defect density of d_0 = 0.5 defects/cm^2 and a defect clustering parameter $\alpha = 0.5$. With a die area $A = 50\ mm^2$, the number of Gross Dies per Wafer (GDW) are estimated to be 1283 [14]. With the negative binomial formula for yield, a die yield of $Y_D = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$ = 81.65% is expected [15]. To estimate the cost to manufacture and process a wafer we use the cost model of [16]; the total price of a 300 mm wafer is estimated at approximately $2779. The model in [16] considers a variety of costs, including installation, maintenance, lithography and material.

For the cost of manufacturing TSVs, we base our numbers on the work of EMC-3D consortium reported in [5]; the cost of fabricating 5 μm TSVs on a single wafer cost $190 and these cost are additive to the wafer cost. We assume the cost of manufacturing TSVs to be 60% of the 3D stacking process cost [17]. Further, we assume the TSVs to have a yield of 98% per die.

The 3D stacking process cost (including bonding, thinning etc..) is assumed to be $126 (40% of total 3D cost) [17]. In addition, the stacking yield is assumed to be composed of two parameters: the interconnect (TSV) yield Y_{INT} and the stacked-die yield Y_{SD}. In our simulations, the interconnect yield Y_{INT} is considered to be 99%. For the good dies that enter the stack, a small probability exists that they get

TABLE I

FAULT COVERAGE VERSUS TEST COST.

fault coverage (%)	ratio test cost (%)	test cost ($cent)
100	100	23
95	28	6.44
85	13	2.99
75	3	0.69
0	0	0.00

corrupted during stacking; this is modeled by the stacked-die yield Y_{SD} and is assumed to be 99%. In [11], a stack yield of approximately 96% is used.

It is worth noting that for our case-study, we assumed that during the stacking only the *top* two dies and the interconnect between them could be corrupted; they are assumed to be defect-prone to stacking/bonding steps like heating, thinning, pressure.

Test cost: To estimate the test cost per die, the model in [15] is used; the model includes depreciation, maintenance and operating cost and assumes five ATE machines operating simultaneously. The derived test cost equals 3.82 $cent/second per die. Assuming a test time of 6 seconds per die, the test cost will be $0.23 per die. We attribute this test cost to a 100% fault coverage. Table I shows the relation between the fault coverage and die test cost [15] for the remaining considered fault coverage values. In [9], the authors estimate a test time of 80 μs to estimate 10000 TSVs using active probing. Hence, we ignore the test cost for pre-bond TSV test. We assume a pre-bond TSV fault coverage of 100%.

For the interconnects between the die, a test cost ratio of 1:100 with respect to the die cost is assumed (as in [11]). For the interconnects a fault coverage of 100% is assumed as well. We assume the fault coverage in the post-bond and final-test to be 100% to prevent faulty packaged ICs and to guarantee the final product quality.

Logistics cost: As discussed in Figure 3, there are many costs related with transportation of tiers during the production of 3D-SICs. For the default process, we assume zero cost for logistics.

Packaging cost: The packaging cost for 3D SICs used in our model is assumed to be 1.25 dollar per 3D-SIC [18]. The costs are comprehensive and include machine, maintenance, labor and material cost. We assume a 100% packaging yield, therefore impacting all the test flows in the same way.

IV. IMPACT OF VARIABLE FAULT COVERAGE

In this section, we analyze the impact of variable fault coverage on the overal 3D-SIC cost. Section IV-A lists the performed experiments. Section IV-B presents and discusses the impact of variable fault coverage. Note that because of space limitations we focus only on the overall cost rather then on the cost break down.

A. Experiments performed

We compare the overall cost of a 3D-SIC by performing the following three experiments given next for the test flows.

TABLE II

FAULT COVERAGE FOR DIFFERENT TEST FLOWS

	Test Flow number											
	1	2	3	4	5	6	7	8	9	10	11	12
pre-bond	100	100	100	95	95	95	85	85	85	75	75	75
mid-bond	100	85	0	100	85	0	100	85	0	100	85	0

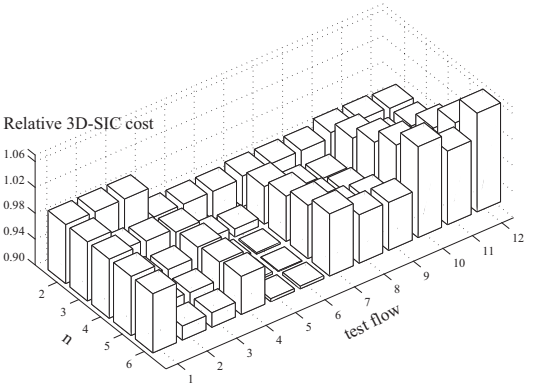


Fig. 4. Impact of variable stack size.

1) **Impact of variable stack size**: The experiment considers a stack size $2 \leq n \leq 6$.
2) **Impact of variable die yield**: The experiment considers a die yield $0.6 \leq Y_d \leq 0.9$.
3) **Impact of variable stack yield**: The experiment considers an interconnect yield $0.91 \leq Y_{INT} \leq 0.99$, and a stacked-die yield $0.91 \leq Y_d \leq 0.99$.

Each experiment is performed for 12 test flows; each test flows consists of the following tests:

1) Pre-bond tests: we assume tests with variable fault coverage for pre-bond testing; see Table II; for example, for test flow 4 we assumed FC=95%.
2) Mid-bond tests: Similarly, as in the pre-bond, we assume again variable FC in this test phase; see Table II; for example, test flows 3, 6 and 9 have no mid-bond test at all while test flows 2, 5, 8, 11 have FC=85%.
3) Post-bond and final tests: The FC for both tests is assumed to be 100%. This to prevent faulty packaged ICs and to guarantee the final product quality.

B. Simulation Results

The section describes the results of the three experiments.

Impact of variable stack size: Figure 4 depicts the relative cost of producing a 3D-SIC for the 12 test flows for stack sizes $2 \leq n \leq 6$; the cost is normalized to Test Flow 1 (TF1). Inspecting Figure 4 reveals the following conclusions.

- Depending on the stack size and the chosen test flow, the overall cost of 3D-SIC increases or decreases for different test flows.
- For a given stack size, the overall cost can be optimized by choosing appropriate test flow combined with appropriate pre-bond and mid-bond fault coverage. For example, for n=3, 4, 5 or 6, the cost is optimal when

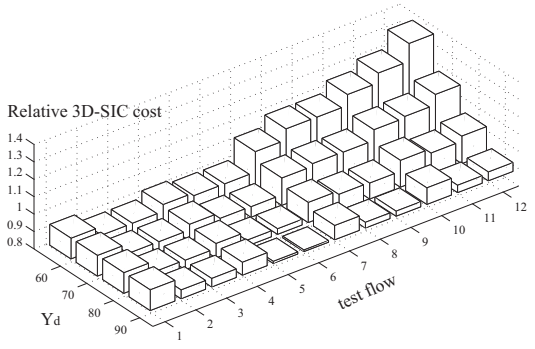


Fig. 5. Impact of variable die yield.

using TF6 with a pre-bond fault coverage of 95% and a mid-bond fault coverage of 0%. A cost reduction of almost 20% can be obtained for a stack size n=6 with respect to TF1.

- Having a pre-bond fault coverage of 100% does not always results in optimal overall cost. In our case, the optimal cost is realized for a pre-bond fault coverage of 95% (TF6).
- Having a mid-bond fault coverage of 100% or a fault coverage of 0% does not always results in optimal overall cost. In our case, the optimal cost is realized for a mid-bond fault coverage of 0% (TF6).

Impact of variable die yield: Figure 5 shows the normalized cost of a 3D-SIC for the 12 test flows for variable die yield $60\% \leq Y_d \leq 90\%$. Inspecting Figure 5 reveals the following conclusions.

- The overall cost depends significantly on the die yield and the chosen test flow. For example, in case the die yield equals 60% TF2 performs best. However, for higher die yields (70% and higher) TF6 peforms best.
- Choosing appropriate values for the pre-bond and mid-bond fault coverage that leads to optimal costs reduction is die yield dependent.

Impact of variable stack yield: Figures 6 and 7 depict the relative cost of a 3D-SIC for the 12 test flows for variable stacked die yield $91\% \leq Y_{SD} \leq 99\%$ and variable interconnect yield $91\% \leq Y_{INT} \leq 99\%$ respectively. Inspecting Figure 6 reveals the following conclusions.

- Depending on the stacked-die yield and the selected test flow, the overall cost significantly depends on the quality of the mid-bond test. For example, test flows with no mid-bond testing (TF3, TF6, TF9 and TF12) result in higher overall cost for lower stacked-die yields.
- For high stacked-die yields (>95%) experiment TF6 performs best. However, for stacked-die yields equal to 95% and lower TF5 results in lowest costs and mid-bond testing pays off.

Figure 7 reveals the following conclusions.

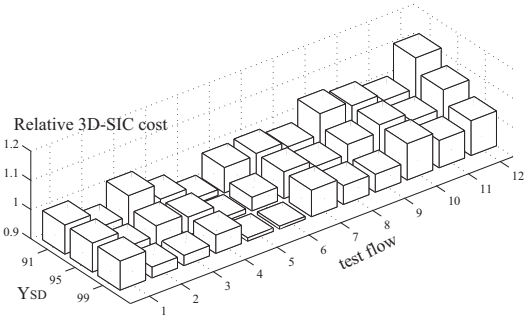


Fig. 6. Impact of variable stacked-die yield.

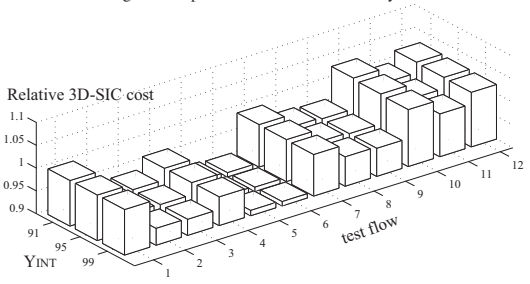


Fig. 7. Impact of variable interconnect yield.

- As the fault coverage for the interconnects is constant and 100%, the relative costs are almost independent of the test flow.
- Relatively, to TF1, the impact of the interconnect yield is almost constant. The reason for this is that we do not modify the interconnect fault coverage. Note however, that the absolute costs for TF1 change for different interconnect yield.
- In this experiment, TF6 always results in overall optimal 3D-SIC cost. Note that the considered interconnect yield is considered to be larger than 91%.

V. Impact of Logistics costs

In this section, 3D-COSTAR will be used to evaluate the impact of logistic cost using the most important tests flows presented in the previous section. The impact of logistic costs is company dependent. For example, the cost for logistics for an IDM company which has all its activities in-house (i.e., manufacturing, testing and packaging) and a fab-less company which outsources its activities are different.

We assume two different models for the logistics cost. In the first model, referred to as the extensive model, we assume non-zero values for all arrows in Figure 3. For the test flows with no mid-bond testing such as TF3, appropriate zero cost values will be for example assigned to arrow D as this arrow is not applicable for this case. For the second model, we assume a reduced logistics model in which some of the activities are joint. Figure 8 shows this model with the applicable arrow labels of Figure 3. In this model, the

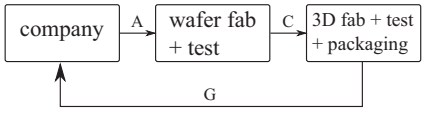


Fig. 8. Reduced logistics model.

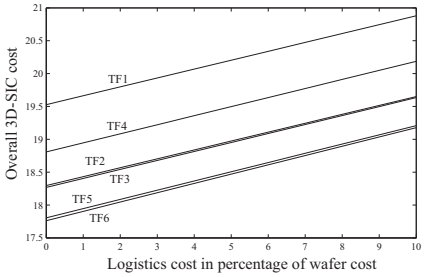


Fig. 9. Impact of extensive logistics cost model.

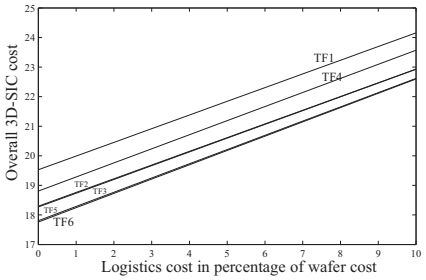


Fig. 10. Impact of reduced logistics cost model.

foundry is responsible for manufacturing the wafer and the OSAT for the remaining steps [19].

We assume the cost to move a single wafer between any to fabs between 0% and 10% of the manufacturing cost of a single wafer (i.e., for each of the involved arrows in Figures 3 and 8), regardless of the stack size.

Figures 9 and 10 show the impact on the overall 3D-SIC cost of variable logistics cost for the reduced and extended logistics cost model; this is performed for the most relevant test flows TF1 up to TF6 of the previous section. Both graphs have the same 3D-SIC cost when the logistics cost is 0%. The figures reveal that (a) with increased logistics cost, the impact of the extensive logistics model on the overall 3D-SIC cost is larger; and (b) the impact of logistics is nearly independent of the test flow, i.e., the slopes of the lines are similar.

In order to optimize the overall test cost, an appropriate test flow should be selected depending on the manufacturing and design parameters. A tool such as 3D-COSTAR can a have an added value in making appropriate trade-offs.

VI. Conclusion

In this paper a tool, 3D-COSTAR, is used to evaluate the different test flows for 3D-SIC; the tool considers all costs involved in the 3D-SIC production (including design, manufacturing, testing, packaging and logistic) and produces the overall cost. As a case study, 3D-COSTAR was used to compare the overall cost of producing a 3D-SIC for variable fault coverage. As mid-bond testing increases the amount of wafer transport, we investigate the impact of logistics as well. Our results show that the optimal test flow strongly depends on design, manufacturing and test parameters such as stack size, die yield, stack yield, fault coverage, etc. In addition, the impact of two logistics models on the most relevant test flows show that as long as the transports costs per single wafer are low, the overall impact of the logistics is relatively minor.

References

[1] C. Zinck, "3D Integration Infrastructure Amp; Market Status," in *IEEE Int. 3D Systems Integration Conference (3DIC)*, Nov. 2010, pp. 1 –34.
[2] M.-J. Wang, C.-Y. Hung, C.-L. Kao, P.-N. Lee, C.-H. Chen, C.-P. Hung, and H.-M. Tong, "TSV Technology for 2.5D IC Solution," in *IEEE 62nd ECTC*, 29 2012-june 1 2012, pp. 284 –288.
[3] E. J. Marinissen, "Challenges and Emerging Solutions in Testing TSV-based $2\frac{1}{2}$D- and 3D-Stacked ICs," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 1277–1282.
[4] P. Garrou, C. Bower, and P. Ramm, *Handbook of 3D Integration: Technology and Applications of 3D Integrated Circuits*, no. v. 1.
[5] M. Taouil, S. Hamdioui, K. Beenakker, and E. J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking," in *19th IEEE Asian Test Symposium (ATS)*, Dec. 2010, pp. 435 –441.
[6] M. Taouil, S. Hamdioui, and E. J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," in *IEEE 6th Int. Design and Test Workshop (IDT)*, Dec. 2011, pp. 24 –29.
[7] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias," in *Int. Test Conference (ITC)*, Nov. 2009, pp. 1–11.
[8] P.-Y. Chen, C.-W. Wu, and D.-M. Kwai, "On-chip tsv testing for 3d ic before bonding using sense amplification," in *Asian Test Symposium, 2009. ATS '09.*, 2009, pp. 450–455.
[9] B. Noia and K. Chakrabarty, "Pre-bond probing of tsvs in 3d stacked ics," in *IEEE International Test Conference (ITC)*, 2011, pp. 1–10.
[10] H. Lee and K. Chakrabarty, "Test challenges for 3d integrated circuits," vol. PP, no. 99, 2013, pp. 1–1.
[11] J. Verbree, E. J. Marinissen, P. Roussel, and D. Velenis, "On The Cost-Effectiveness of Matching Repositories of Pre-Tested Wafers for Wafer-to-Wafer 3D Chip Stacking," in *ETS*, May 2010, pp. 36–41.
[12] M. Taouil, S. Hamdioui, J. Verbree, and E. J. Marinissen, "On Maximizing The Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *IEEE Test Conference (ITC)*, Nov. 2010, pp. 1 –10.
[13] R. Tummala, *Fundamentals of Microsystems Packaging*. McGraw-Hill, 2001. [Online]. Available: http://books.google.nl/books?id=_7S-QwAACAAJ
[14] D. de Vries, "Investigation of Gross Die per Wafer Formulas," *IEEE Trans. on Semiconductor Manufacturing*, vol. 18, no. 1, pp. 136–139, Feb. 2005.
[15] V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*, ser. Frontiers in Electronic Testing.
[16] Sematech. (2013) Sematech wafer cost comparison calculator. [Online]. Available: http://ismi.sematech.org/modeling/agreements/wafercalc.htm
[17] D. Velenis, M. Stucchi, E. J. Marinissen, B. Swinnen, and E. Beyne, "Impact of 3D Design Choices on Manufacturing Cost," in *3DIC*, sept. 2009, pp. 1–5.
[18] DIMES. (2010) Delft institute of microsystems and nanoelectronics. [Online]. Available: http://www.dimes.tudelft.nl/
[19] R. Hwang. Test in Transition: An OSAT Perspective. [Online]. Available: http://semiconwest.org/sites/semiconwest.org/files/Roger Hwang_ ASE Group.pdf

2014 IEEE 32nd VLSI Test Symposium (VTS)

Quality versus Cost Analysis for 3D Stacked ICs

Mottaqiallah Taouil[1] Said Hamdioui[1]
[1]Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{m.taouil, s.hamdioui}@tudelft.nl

Erik Jan Marinissen[2]
[2]IMEC vzw
3D Integration Program
Kapeldreef 75, 3001 Leuven, Belgium
erik.jan.marinissen@imec.be

***Abstract*—To fulfill customer demands, IC products must satisfy the required quality generally expressed in defective parts per million (DPPM). To meet this DPPM target, appropriate test infrastructures and test approaches must be developed. This is a challenging task for 3D Stacked-ICs (3D-SIC) due to a large test flow space; each test flow may require different design-for-test features and impact the product quality and total stack cost differently. Therefore, appropriate models to predict the impact of test flows on the product quality and overall stack cost at early design stage is important for quality versus cost trade-offs. This paper presents a model that predicts the 3D product quality in terms of DPPM for different test flows and associated cost; it incorporates the quality of the wafer manufacturing, stacking and packaging process. For example, the presented case study showed that maintaining the same product quality for larger stack size might result in a significant test cost increase.**

I. Introduction

Tremendous effort has been put in place to bring *through-silicon via (TSV)* based 2.5D and 3D-SIC technology closer to market [1–3]. Realizing such ICs is attractive due to major benefits [4] such as (a) increased electrical performance, (b) reduced power consumption due to shortened interconnects, (c) heterogeneous integration, (d) reduced form factor, etc.

Providing an acceptable product quality (measured in defective parts per million (DPPM)) to satisfy the customer needs is crucial in semiconductor industry, including 2.5D/3D technology. Obviously, the required product quality is strongly application dependent. To guarantee the required DPPM level, appropriate testing should be performed. However, testing 2.5D/3D is much complexer than testing 2D chips as they provide several test moments such as before stacking, during partial stacks, or after the complete packaged manufactured stack. This results into a large space of test flows; each test flow requires different cost and DFT infrastructure and may result in a different DPPM level. Determining most cost-effective test flow being able to provide the required DPPM level is of great importance in order to optimize the overall 3D IC cost. Note that different test flows, executed after manufacturing, may require different design-for-test features, which need to be incorporated in the various dies during their early design stages.

Several cost models have been published in the area for 2.5D/3D-SICs. Most of them have focused on cost modeling for 3D manufacturing (as in [5] and [6]), stacking and integration (as in [7], [8] and [9]), TSV count and die area (as in [10] and [11]). However, limited work is published on test cost modeling and its impact on the overall chip quality and cost. In [12], the authors proposed a cost model that emphasizes on manufacturing *and* test cost; the authors investigated the impact of Die-to-Wafer (D2W) and Wafer-to-Wafer (W2W) stacking on overall cost and determined the lower bound of the yield of the final package level test given the number of stacked dies and the final yield. In [13,14] the tool 3D-COSTAR was proposed and used to analyze the impact of test flows on the overall stack cost; 3D-COSTAR is a tool being able to incorporate design cost, manufacturing cost, test cost and logistic cost to both estimate (a) the overall 2.5D and 3D-SIC cost, and (b) cost break down for any 3D test flow. In [15], the author propose heuristics to find cost-wise optimal test flows that include mid-bond testing. The above clearly shows that estimating the DPPM level for a given test cost budget and/or providing the cost effective test flow for a given DPPM level is not investigated yet.

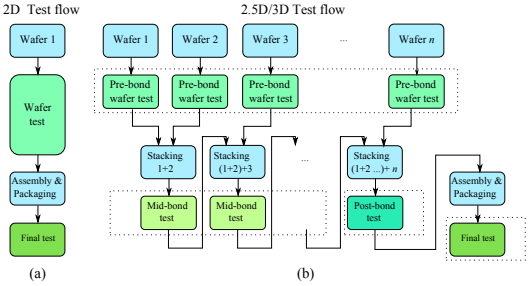


Fig. 1. 2D versus 2.5D/3D D2W test flows.

In this paper, we present a novel 3D modeling framework that estimates the product quality in DPPM; the framework incorporates the quality (expressed in yield) of the wafer manufacturing, stacking and packaging process. In addition, it supports all 3D tests flows, i.e., pre-bond, mid-bond, post-bond and final tests can be applied both for dies and interconnects each with variable test quality.

The remainder of this paper is organized as follows. Section II briefly presents some background on 3D-SIC testing. Section III presents a framework to estimate the product quality for 3D-SICs. Section IV describes the experiments and results. Finally, Section V concludes the paper.

II. 3D-Testing

Figure 1(a) shows the conventional 2D test flow for planar wafers [16]; it consists of two test moments: a wafer test

978-1-4799-2611-4/14/$31.00 ©2014 IEEE

prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low as it prevents unnecessary assembly and packaging costs, while the final test is used to guarantee the final quality of the packaged chips. 2.5D/3D-SICs, however, provide additional test moments; e.g., additional test moments can be defined for each partial stack. Moreover, at each moment a distinction can be made between different tests such as die and interconnect tests. In general, four test phases can be distinguished for a 3D-SICs consisting of n dies as depicted in Figure 1(b): (1) n *pre-bond* wafer tests, (2) n-2 *mid-bond* tests, (3) one *post-bond* test prior packaging and (4) one *final* test; resulting into 2·n test moments [17].

A *test flow* can be extracted from the above four defined test phases and is a collection of tests applied at one or more of these phases. At each test phase, zero, one or more tests, possibly with different fault coverages, both for dies and/or interconnects, can be applied. Outgoing product quality as well as test cost is test flow dependent. Therefore, choosing an optimal test flow to suit the targeted DPPM level and/or the test budget is of great importance.

It is worth noting that 100% flexibility in choosing appropriate test flows (with associateed fault coverage) is not always possible. For instance, depending on whether one or more companies are involved in the supply chain for the manufacturing of 2.5D/3D-SICs, different test requirements can be set for the pre-bond wafer test [16]. If the wafers are produced by a company (or companies) different than the company responsible for stacking, then a high pre-bond wafer test quality (e.g. a Know-Good-Die contract) often is agreed upon. If such a contract is not in place (e.g., for an IDM), the pre-bond test quality is subject to optimization. In this case, we do not have only the freedom to include or skip the pre-bond test phase in the test flow, but also the freedom to tune the fault coverage as well. Faulty undetected dies at this test phase can be detected in a later test phase, e.g., at the finale test (after packaging).

III. 3D Test Quality Framework

A 3D-SIC consists of multiple dies and interconnects between them; both are susceptible to defects during manufacturing, stacking and packaging. In essence, we distinguish between three defect sources:

1) Manufacturing defects prior to the pre-bond test.
2) Stacking defects prior to mid- and post-bond tests.
3) Assembly and packaging defects prior to the final test.

The quality of these processes are denoted by different yields as illustrated in Figure 2, where the stacking process of three dies is shown. Prior to stacking, the quality of each wafer *manufacturing* process is described by the actual pre-bond die yield denoted as $Y_{ma,d}(i)$ for Die i. The fault coverage for the pre-bond test is similarly denoted by $fc_{ma,d}(i)$ for Die i. The quality of the test determines the outgoing die yield, i.e., the fraction of good dies after testing which may include test escapes. Several work exists that models this relation

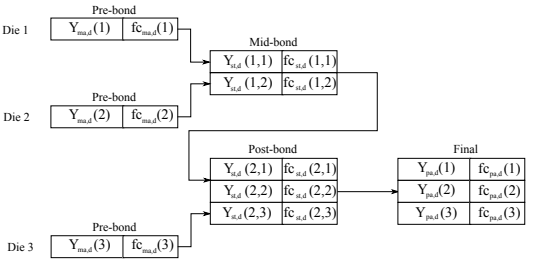


Fig. 2. Yield modeling of different processing steps.

between actual yield, test quality, and outgoing yield [18,19] for planar dies.

After pre-bond tests of Die 1 and 2 both dies are stacked. During this stack operation, new defects might emerge due to e.g. thinning, bonding, thermo-mechanical stress, etc. The quality of this entire *stacking* process is modeled by the stacked-die yield of each die in the partial stack denoted as $Y_{st,d}(i,j)$, where the first index presents the i^{th} stacking operation, j the die index, and st stands for stacking; the interconnect yield will be discussed later in this section. $Y_{st,d}(1,1)$ and $Y_{st,d}(1,2)$ present the stacked-die yields for Die 1 and 2, respectively, after the first stacking operation. After this partial stack is created, dies can be re-tested (mid-bond phase) with test(s) having possibly different fault coverages for the two dies, denoted as $fc_{st,d}(1,j)$ for Die j. Note that during this test, not only faults due to the stacking process can be detected, but also those escaped the pre-bond test. Similarly, when Die 3 is stacked (during the *second* stacking operation) on the partial stack consisting of Dies 1 and 2, yields can be attributed to all dies in the stacking denoted by $Y_{st,d}(2,j)$ for Die j; each die might be tested (post-bond test) with a fault coverage $fc_{st,d}(2,j)$.

The final process that may introduce defects into the stack is the assembly and *packaging* process. In a similar way, the quality of this process is estimated by the yield of each die in the package denoted by $Y_{pa,d}(j)$ for Die j, pa stands for packaging. After assembly and packaging, a final test(s), possibly with different fault coverages for each die (denoted by $fc_{pa,d}(j)$ for Die j) can be applied.

Similarly as for the dies, we define the yields $Y_{st,i}$, and $Y_{pa,i}$ to estimate the quality of interconnects during stacking and packaging respectively (note that they are not shown in Figure 2). For example, $Y_{st,i}(i,j)$ is the interconnect yield of the i^{th} stacking operation for Interconnect j (i.e., between Dies j and j+1). For the sake of simplicity, the pre-bond TSV yield is ignored, i.e., its yield is assumed to be 100%; although the model can easily be extended to incorporate this.

Next, we will estimate the overall test escape rate of a 3D-SIC after the whole stacking process, as depicted in Figure 2; obviously, this test escape rate is a function of defects that emerged at wafer manufacturing, stacking, and/or assembly and packaging, and passed all the tests applied at different

test moments. However, first the relationship between escapes and the test quality will be discussed.

Equation 1 defines the relation between test escapes TE, actual or ingoing yield Y_{in} and outgoing yield Y_{out}; TE describes the ratio of faulty dies that pass the test. The ingoing yield is the actual yield (which is an estimated number), the outgoing yield is the fraction of dies that is considered to be good after testing (includes both good dies and test escapes). Equation 2 [18] shows an example of the relation between test escapes, ingoing yield and the fault coverage (FC). By combining equations 1 and 2, the outgoing yield as a function of the ingoing yield and fault coverage is obtained in Equation 3.

$$TE = \frac{Y_{out} - Y_{in}}{Y_{out}} \quad (1)$$

$$TE(FC) = 1 - Y_{in}^{1-FC} \quad (2)$$

$$Y_{out}(FC) = \frac{Y_{in}}{1 - TE} = \frac{Y_{in}}{Y_{in}^{1-FC}} = Y_{in}^{FC} \quad (3)$$

In general, unless otherwise stated, we assume that Equations 1, 2 and 3 hold. Our target is to obtain the overall test escape rate of the 3D-SIC, say TE_{3D}. Again, this test escape rate is a function of the test escape due to defects emerged during manufacturing, stacking and/or assembly and packaging. This rate is captured by Equation 4.

$$TE_{3D} = 1 - (1 - TE_{ma}) \cdot (1 - TE_{st}) \cdot (1 - TE_{pa}) \quad (4)$$

In this equation, TE_{ma} presents the test escape rate due to defects emerged at wafer manufacturing that passed all tests applied at the different test moments, similarly TE_{st} and TE_{pa} presents the test escape rate due to defects emerged at different stacking operations and assembly/packaging respectively. The ratio $1-TE$ describes the good fraction of each particular process after all tests are applied. Next, the test escapes rates due to wafer manufacturing, stacking and assembly/packaging are discussed individually.

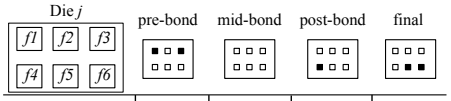
A. Test Escapes related to Wafer Manufacturing

Equation 2 assumes an infinite defect clustering parameter which has the tendency to report too pessimistic number of test escapes [19]. Therefore, we rather consider the clustering parameter for accurate and realistic estimation. In that case, the outgoing yield changes into [19]:

$$Y_{out}(FC) = \left(1 + \frac{FC \cdot A \cdot d_0}{\alpha}\right)^{-\alpha} \quad (5)$$

where Y_{out} presents the outgoing yield as a function of the fault coverage FC, A the die area, d_0 the defect density, and α the clustering parameter; note that in general any description of Y_{out} as a function of FC can be used here. By inserting Equation 5 into Equation 1, we obtain the test escape rate as a function of the fault coverage, die area, defect clustering parameter and defect density as it is shown in Equation 6.

$$TE(FC) = 1 - \frac{(\alpha + FC \cdot A \cdot d_0)^{\alpha}}{(\alpha + A \cdot d_0)^{\alpha}} \quad (6)$$



test moment	*i*=0	*i*=1	*i*=2	*i*=3
fault list	{*f1*,*f3*}	{0}	{*f4*}	{*f5*,*f6*}
$fc_{ma,d}(i,j)$	33.33%	00.00%	16.67%	33.33%
FC (unified)	33.33%	33.33%	50.00%	83.33%

Fig. 3. Incremental fault coverages.

We initially defined the fault coverage for a die test that detects manufacturing defects as $fc_{ma,d}(j)$ (as shown in Figure 2). Here index j presents the die index of the particular die. However, as dies can be tested at each test phase, $fc_{ma,d}(j)$ is expanded to $fc_{ma,d}(i,j)$ where i holds information regarding the test moment. Note that this information is not depicted in Figure 2. The pre-bond fault coverage for a Die j is defined for i=0 and denoted as $fc_{ma,d}(0,j)$, while i=1 until i=n-1 present the fault coverage for mid- and post-bond tests applied after the i^{th} stacking operation (note that there are n-1 stacking operations) and finally, i=n presents the fault coverage for Die j at the final moment (i.e., after packaging). It is worth noting that not all values of i are applicable for each Die j during mid- and post-bond tests. For example, Figure 2 shows that Die 3 enters the stack at $i = 2$ and therefore $fc_{ma,d}(0,3)$ and $fc_{ma,d}(2,3)$ are valid, but $fc_{ma,d}(1,3)$ is not. For a sequential in-order stacked 3D-SIC, i and j can only take the values $1 \leq i \leq (n-1)$ and $1 \leq j \leq (i+1)$ for the mid- and post-bond test moments.

In order to estimate the test escape rate related to wafer manufacturing for a *single* die in a stack, we redefine Equation 6, that describes the defect level after wafer test, by computing a single unified fault coverage $FC_{ma,d}(j)$ for all the tests targeting manufacturing defects in Die j. Figure 3 shows how to obtain this fault coverage for a particular die of a three-layer 3D-SIC. The top left of the figure presents the fault space of the die consisting of 6 faults denoted by *f1* until *f6*. The top right part of the figure shows for each test the faults that are detected (denoted by the black boxes). For example, the pre-bond test (i=0) covers faults *f1* and *f3* resulting in a fault coverage of ($fc_{ma,d}(0,j) = \frac{|\text{fault\_list}|}{\text{number of faults}}$=33.33%). The combined fault coverage is obtained by including the faults that were previously tested. For example, after the final test faults *f1*, *f3*, *f4*, *f5* and *f6* are covered resulting in $FC_{ma,d}(j)$=$\frac{5}{6}$=83.33%. This unified fault coverage for each Die j is denoted by $FC_{ma,d}(j)$. Equation 7 describes the test escapes due to manufacturing defects for Die j.

$$TE_{ma}(j) = 1 - \frac{(\alpha + FC_{ma,d}(j)) \cdot A \cdot d_0)^{\alpha}}{(\alpha + A \cdot d_0)^{\alpha}} \quad (7)$$

As a stack consists of n dies, the test escape rate due to defects emerged during wafer manufacturing of all these dies can be estimated by Equation 8; it combines the test escape ratios of each die in the stack.

$$TE_{ma} = 1 - \prod_{j=1}^{n} (1 - TE_{ma}(j)) \quad (8)$$

B. Test Escapes related to Stacking

Defects introduced during stacking are inherent to 3D processing steps such as bonding, thinning and die alignment; they can impact either dies or interconnects. Therefore, two yield parameters are used. The first one is related to the dies and is called *stacked-die* yield and the second one to the interconnects referred to as *interconnect* yield; both are discussed next.

Stacked-die yield: The stacked-die yield $Y_{st,d}$ (see Figure 2) specifies the fraction of dies that survive the stacking operations. As Figure 1 shows, there are $n-2$ mid-bond stacking operations. Therefore, $Y_{st,d}$ is as a two-dimensional array of size $(n-1) \times n$. The $n-1$ rows specify the index of the stacking operations, and the n columns specify the stacked-die yield for each die. For a sequential in-order stacked 3D-SIC, $Y_{st,d}(i,j)$ is valid for $1 \leq j \leq (i+1)$.

Die defects may emerge due to stacking. These can be detected either during the mid-bond, post-bond or final test phase. We denote the FC of each test applied to detect stacking caused die defects as $fc_{st,d}(i,j)$; this presents the fault coverages of Die j during mid-bond test for $i \leq n-2$, during post-bond test for $i=n-1$, and during final test for $i=n$.

We extend Equations 2 and 3 to be able to calculate the outgoing yield and test escapes after applying a sequence of two or more stacking operations to a particular die. It is possible to test this die after the first and second stacking operation with different FCs f_{c1} and f_{c2}, respectively, to target stacking defects. Note that the stacked-die yield might differ for each of the two stacking operations for the considered die. After the first stacking operation, only the test with FC=f_{c1} is applicable and Equations 2 and 3 still hold. After the second one, the test escapes depend on the stacking yields and fault coverages of both tests. To determine the test escapes due to the first stacking operation (with yield y_1) the unified fault coverage of the tests with fc_1 and fc_2 (denoted by $U(fc_1, fc_2)$) needs to be used, while for the second stacking operation (with yield y_2) only fc_2 is applicable. Therefore, the test escape rate can be formulated by TE $=1-\{y_1^{(1-U\_FC(fc_1,fc_2))} \cdot y_2^{(1-fc_2)}\}$

Each test targets a set of faults due to stacking, which are assumed to be similar for all stacking operations. The unified fault coverage is calculated in a similar way as presented in Figure 3 for manufacturing defects. In general, after p stacking operations the test escape rate equals:

$$TE = 1-\{y_1^{1-U(fc_1,fc_2,...,fc_p)} \cdot y_2^{1-U(fc_2,...,fc_p)} ... \cdot y_p^{1-U(fc_p)}\} = 1-\prod_{i=1}^{p} y_i^{1-U(fc_i,...,fc_p)} \quad (9)$$

A 3D-SIC has in general p=n-1 stacking operations and n dies in the stack, resulting into n test moments (one after each stacking operation and one final test). By using Equation 9, the test escapes for Die j can be formulated by:

$$TE_{st,d}(j) = 1-\prod_{i=max(1,j-1)}^{n-1} Y_{st,d}(i,j)^{(1-FC_{st,d}(i,j))} \quad (10)$$

In this equation, the test escape rate for Die j is calculated by considering all stacked-die yields $Y_{st,d}(i,j)$ of Die j confined by the product operator. For each of these stacking operations a unified fault coverage $FC_{st,d}(i,j)$ is calculated as defined by Equation 11. This unified fault coverage includes all stack tests for Die j after the i^{th} stacking operation which include (a) mid-bond tests (if applicable), (b) the post-bond test and (c) the final test.

$$FC_{st,d}(i,j) = U(fc_{st,d}(i,j), ..., fc_{st,d}(n,j)) \quad (11)$$

Next, we define $TE_{st,d}$ in Equation 12 as the combined escape rate due to all stacking defects in all dies by using the expression in Equation 10 for the test escapes of n individual dies.

$$TE_{st,d} = 1-\prod_{j=1}^{n}(1-TE_{st,d}(j)) \quad (12)$$

a) Interconnect: The second type of components in the stack are the interconnects. Interconnects form the electrical connections between the dies. For example, Interconnect j=1 specifies the interconnection between Die 1 and 2. Similarly as for dies, the matrix $Y_{st,i}(i,j)$ specifies the yield of Interconnect j after the i^{th} stacking operation. We denote the FC of each test applied to detect interconnect defects as $fc_{st,i}(i,j)$; this presents the fault coverages of Interconnect j during mid-bond test for $i \leq n-2$, during post-bond test for $i=n-1$, and during final test for $i=n$. The array $fc_{st,i}(i,j)$ contains only valid values for $1 \leq i \leq n$ and $1 \leq j \leq min(i, n-1)$.

Using a similar approach that is used for the calculation of $TE_{st,d}$, one can easily derive that the test escape rate, for Interconnect j, due to stacking defects is:

$$TE_{st,i}(j) = 1-\prod_{i=j}^{n-1} Y_{st,i}(i,j)^{(1-FC_{st,i}(i,j))} \quad (13)$$

$$FC_{st,i}(i,j) = U(fc_{st,i}(i,j), ..., fc_{st,i}(n,j)) \quad (14)$$

$$TE_{st,i} = 1-\prod_{k=1}^{n-1}(1-TE_{st,i}(j)) \quad (15)$$

Equation 15 defines the combined test escape rate due to all stacking defects in all interconnects by combining the individual test escape rates in Equation 13. If we combine Equations 12 and 15 for the partial test escapes of dies and interconnects due to stacking operations, we obtain the test escapes due to stacking as follows:

$$TE_{st} = 1-(1-TE_{st,d}) \cdot (1-TE_{st,i}) \quad (16)$$

TABLE I
DEFAULT PARAMETERS

Parameter	Value	Parameter	Value
Wafer costs ($)	2279	Effective wafer radius (mm)	147
Die Area (mm)	50	Dies per wafer	1283
Defect density (cm^{-2})	0.5	Die yield (%)	81.65
Pre-bond die test cost ($)	0.50	Pre-bond die fc (%)	99.6
Stacking cost ($)	0.10	–	–
Stacked die yield (%)	99	Stacked int. yield (%)	99
Mid-bond die test cost (%)	0	Mid-bond die fc (%)	0
Post-bond die test cost ($)	0.10	Post-bond die fc (%)	100
Mid-bond int test cost (%)	0.00	Mid-bond int fc (%)	0
Post-bond int test cost ($)	0.05	Post-bond int fc (%)	100
Packaging cost ($)	1.25	Packaging yield (%)	99
Final die test cost ($)	0.10	Final die fc (%)	100
Final int test cost ($)	0.05	Final int fc (%)	100

C. Test Escapes related to Assembly and Packaging

Similarly, let $Y_{pa,d}$ and $Y_{pa,i}$ denote the die and interconnect yield, respectively, of the assembly and packaging step. The fault coverages for testing packaging related defects are defined by $fc_{pa,d}$ for the dies and $fc_{pa,i}$ for the interconnects. Equations 17, 18 and 19 summarize the test escape rate.

$$TE_{pa,d} = 1 - \prod_{j=1}^{n} Y_{pa,d}(j)^{(1-fc_{pa,d}(j))} \quad (17)$$

$$TE_{pa,i} = 1 - \prod_{j=1}^{n-1} Y_{pa,i}(j)^{(1-fc_{pa,i}(j))} \quad (18)$$

$$TE_{pa} = 1 - (1 - TE_{pa,d}) \cdot (1 - TE_{pa,i}) \quad (19)$$

In these equations, $Y_{pa,d}(j)$ and $fc_{pa,i}(j)$ present the yield and fault coverage of Die j respectively, $TE_{pa,d}$ and $TE_{pa,i}$ the test escape rates of dies and interconnects, respectively, while TE_{pa} the total test escape rate due to assembly and packaging.

The total 3D test escape rate can be obtained by substituting the test escape rates of the manufacturing, stacking and packaging processes (Equations 8, 16 and 19 respectively) into Equation 4.

IV. SIMULATION RESULTS

This section describes the case studies of this paper. In particular, Section IV-A describes the experiments, Section IV-B the default parameters, and finally, Section IV-C the results.

A. Experiments

Note that the yield and the cost parameters considered for the experiments in this paper do not describe any processes at IMEC or partners, nor at TU Delft. The inputs of the framework are flexible and fully parameterized. By tuning these input parameters almost anything can be proven. Nevertheless, we provide input values as realistic as possible. In this paper, we investigate the following two experiments.

1) Impact of FC and its associated test cost on the product quality and total 3D-SIC cost for various stack sizes.
2) Impact of the die defect density on the test escapes and total 3D-SIC cost for various stack sizes.

TABLE II
FAULT COVERAGE VERSUS TEST COST

fault coverage (%)	98.2	98.5	99.3	99.6	100
test cost ($cent)	4	5	8	50	150

B. Reference Process

Table I summarizes the default parameters used in our experiments; the default stack size is n=4. The first block in the tables describes the parameters related to the pre-bond phase. We assume a standard 300mm diameter wafer with an edge clearance of 3mm, i.e., the effective radius equals 147mm. We estimate a cost of such a wafer to be 2279$ per wafer [20]. The die areas for each die in the stack are assumed to be A=50mm^2. For the given die areas and effective wafer radius, the number of gross dies per wafer (GDW) approximately equals to 1283 [21]. To estimate the cost of manufacturing TSVs, the work of EMC-3D consortium [22] is used; the cost to fabricate 5 um TSVs in a single wafer is assumed to be 190$ and these cost are additive to the wafer cost. All dies except the top die are assumed to have TSVs.
The defect density is considered to be d_0 = 0.1 defects/cm^2 (mature process) with a defect clustering parameter $\alpha = 0.5$. The pre-bond die yield can be estimated by the negative binomial formula as: $Y_{ma,d}(j) = (1 + \frac{A \cdot d_0}{\alpha})^{-\alpha}$=95% [18] for each Die j. The pre-bond fault coverage is assumed to be $fc_{ma,d}(0, j)$=99.6% at a cost of 0.50$.

The next group of variables in the table contains parameters related to the mid-bond and post-bond. We assume the cost of manufacturing TSVs to be 60% of the 3D stacking process cost [8]. Using this relation, we estimate the cost of a single stacking operation to be 0.10$ per die. Each time a die is stacked on another die or on a partial stack, the stacked-die yield of the top two dies are assumed to be 99%. The remainder dies in the stack is assumed to have 100% yield. The yield of the interconnects is assumed to be 99% (which includes the micro-bumps) between each pair of stacked dies. Note that because n=4 there are three stacking operations and three sets of interconnects between the dies. In addition, we assume no mid-bond testing for dies and relative low-cost interconnect tests at a cost of 0.05$ realizing 100% fault coverage. In the final phase, we assume a packaging cost of 1.25 [23] with an overall packaging yield of 99%. To test for packaging defects, the same costs and fault coverage and cost are assumed as in the post-bond.

C. Simulation Results

The reference process resulted in a number of test escapes equal to 780 DPPM at the cost of 13.68$ per 3D-SIC.

The results of the first experiment, the impact of pre-bond FC with variable fault coverage and test cost, are shown in Figure 4. The solid lines present the product quality in DPPM and the dashed line the total 3D-SIC cost, for different stack sizes. In order to attribute test cost to the pre-bond tests, we compile the test cost for tests with different FCs using the data presented in [24]; the results are shown in Table II. From Figure 4 we conclude that:

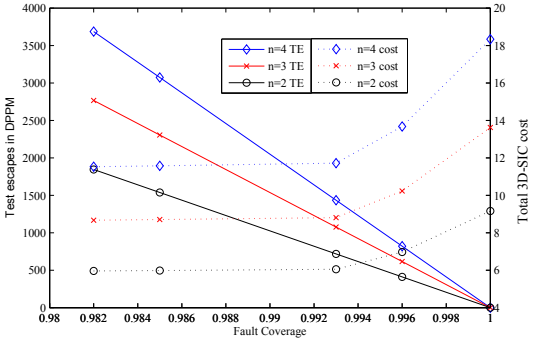


Fig. 4. Impact of FC on test escapes and 3D-SIC cost.

- The total 3D-SIC cost increases significantly for high product quality, mainly caused by the increased test cost.
- In order to obtain the same product quality, larger stacks require higher test quality. For example, to obtain a DPPM of 1000, a fault coverage of 99% is required for two stacked dies. However, in case the stack size increase to four layers, the fault coverage must be 99.95% to reach the same DPPM. As the test cost increases significantly when approaching a 100% FC, higher 3D-SIC costs can be predicted if the DPPM requirement must be maintained.

Figure 5 shows the result of the second experiment, i.e., the impact of the defect density (for all dies taken the same). Note that the fault coverage is taken 99.6% for all cases. From the figure we conclude the following:

- A higher defect density leads to increased cost. The increment is worst for larger stack sizes. For example, for n=2 the cost increment is 30% when the defect density increases from 0 (6.71$) to 1 cm^{-2} (8.73$), while the cost increment equals 33% (from 13.12$ to 17.48$) for n=4.
- The defect density has more impact on the number of test escapes for higher n. For example, the DPPM increases from 0 (d_0=0) to 2000 (d_0=1) for n=2, while this increase almost doubles to 3996 for n=4.

V. Conclusion

In this paper, we presented a novel modeling framework to estimate the 3D product quality and its associated total stack cost. The framework integrates all test moments (pre-bond, mid-bond, post-bond and final test) and considers defects from the wafer manufacturing process, stacking and packaging. The case studies presented have shown the significant importance of this quality framework in order to make appropriate quality and cost trade-offs. For example, the case study showed that maintaining the same product quality for larger stack size might result in a significant test cost increase.

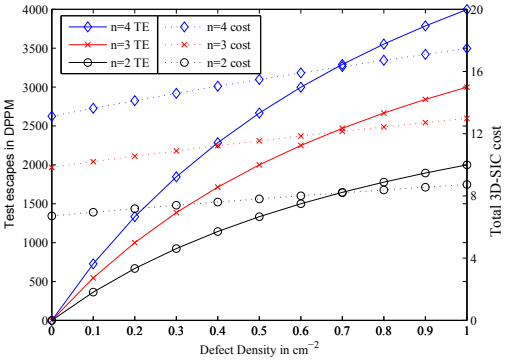


Fig. 5. Impact of defect density on the test escapes and 3D-SIC cost.

References

[1] C. Zinck, "3D Integration Infrastructure Amp; Market Status," in *3DIC*, nov. 2010, pp. 1 –34.

[2] M.J. Wang *et al.*, "TSV Technology for 2.5D IC Solution," in *ECTC*, june 2012, pp. 284 –288.

[3] C. Papameletis *et al.*, "Automated DfT Insertion and Test Generation for 3D-SICs with Embedded Cores and Multiple Towers," in *ETS*, May 2013, pp. 1–6.

[4] P. Garrou, C. Bower, and P. Ramm, *Handbook of 3D Integration: Technology and Applications of 3D Integrated Circuits*. John Wiley & Sons, 2008.

[5] A. Walker, "A Manufacturing Cost Model for 3-D Monolithic Memory Integrated Circuits," *TSM*, vol. 22, no. 2, pp. 268 –275, may 2009.

[6] P. Mercier *et al.*, "Yield and Cost Modeling for 3D Chip Stack Technologies," in *CICC*, sept. 2006, pp. 357–360.

[7] Y. Chen *et al.*, "Cost-Effective Integration of Three-Dimensional (3D) ICs Emphasizing Testing Cost Analysis," in *ICCAD*, nov. 2010, pp. 471 –476.

[8] D. Velenis *et al.*, "Impact of 3D Design Choices on Manufacturing Cost," in *3DIC*, sept. 2009, pp. 1 –5.

[9] X. Dong and Y. Xie, "System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs)," in *ASP-DAC*, jan. 2009, pp. 234 –241.

[10] C.C. Chan, Y.T. Yu, and I.R. Jiang, "3DICE: 3D IC Cost Evaluation Based on Fast Tier Number Estimation," in *ISQED*, 2011, pp. 1 –6.

[11] C. Zhang and G. Sun, "Fabrication Cost Analysis for 2D, 2.5D, and 3D IC Designs," in *3DIC*, feb 2012, pp. 1 –4.

[12] Y.W. Chou *et al.*, "Cost Modeling and Analysis for Interposer-Based Three-Dimensional IC," in *VTS*, april 2012, pp. 108 –113.

[13] M. Taouil *et al.*, "Using 3D-COSTAR for 2.5D Test Cost Optimization," in *3DIC*, 2013, pp. 1–8.

[14] M. Taouil *et al.*, "Impact of Mid-Bond Testing in 3D Stacked ICs," in *DFT*, oct. 2013, pp. 1–6.

[15] M. Agrawal and K. Chakrabarty, "Test-cost optimization and test-flow selection for 3d-stacked ics," in *VLSI Test Symposium (VTS), 2013 IEEE 31st*, April 2013, pp. 1–6.

[16] E.J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs," in *DATE*, march 2010, pp. 1689 –1694.

[17] M. Taouil *et al.*, "Test Cost Analysis for 3D Die-to-Wafer Stacking," in *Test Symposium (ATS), 2010 19th IEEE Asian*, dec. 2010, pp. 435 –441.

[18] V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer, 2000.

[19] J. De Sousa and V. Agrawal, "Reducing the complexity of defect level modeling using the clustering effect," in *DATE*, 2000, pp. 640–644.

[20] Sematech. (2013) Sematech wafer cost comparison calculator. [Online]. Available: http://ismi.sematech.org/modeling/agreements/wafercalc.htm

[21] D. de Vries, "Investigation of Gross Die per Wafer Formulas," *TSM*, vol. 18, no. 1, pp. 136 – 139, feb. 2005.

[22] (2008) Semiconductor 3-d equipment and materials consortium, emc3d.

[23] DIMES. (2010) Delft institute of microsystems and nanoelectronics. [Online]. Available: http://www.dimes.tudelft.nl/

[24] S.K. Lu, T.Y. Lee, and C.W. Wu, "Defect level prediction using multi-model fault coverage," in *ATS*, 1999, pp. 301–306.

Direct Probing on Large-Array Fine-Pitch Micro-Bumps of a Wide-I/O Logic-Memory Interface

Erik Jan Marinissen[1*] Bart De Wachter[1*] Ken Smith[2] Jörg Kiesewetter[3*] Mottaqiallah Taouil[4] Said Hamdioui[4]

[1] IMEC vzw
Kapeldreef 75
B-3001 Leuven
Belgium
erik.jan.marinissen@imec.be
bart.dewachter@imec.be

[2] Cascade Microtech, Inc.
9100 SW Gemini Drive
Beaverton, OR 97008
United States of America
ken.smith@cmicro.com

[3] Cascade Microtech GmbH
Süss Straße 1
Thiendorf 01561
Germany
joerg.kiesewetter@cmicro.com

[4] Technische Universiteit Delft
Dept. of Computer Engineering
Mekelweg 4, 2628CD Delft
The Netherlands
m.taouil@tudelft.nl
s.hamdioui@tudelft.nl

Abstract

In order to obtain acceptable compound stack yields for 2.5D- and 3D-SICs, there is a need to test the constituting dies before stacking. The non-bottom dies of these stacks have their functional access exclusively through large arrays of fine-pitch micro-bumps, which are too dense for conventional probe technology. A common approach to obtain pre-bond test access is to equip these dies with dedicated pre-bond probe pads, which comes with drawbacks such as increased silicon area, test application time, and reduced interconnect performance. In order to avoid the many drawbacks of dedicated pre-bond probe pads, we advocate the usage of advanced probe technology that allows to directly probe on these micro-bumps. This paper reports on the technical and economical feasibility of this approach.

1 Introduction

There is a lot of excitement around and expectations from 2.5D- and 3D-stacked integrated circuits [1]. In 2.5D-SICs, multiple active dies are placed side-by-side on top of and interconnected by a passive interposer die. In 3D-SICs, multiple active dies are stacked vertically. Both 2.5D- and 3D-SICs are enabled by the capability to manufacture through-silicon vias (TSVs) that provide an electrical connection between the front- and back-side of a silicon substrate [2–4]. In 2.5D-SICs TSVs connect the stacked active dies through the silicon interposer to the package substrate. In 3D-SICs TSVs provide vertical interconnections between the various stacked dies. Both types of SICs serve their particular market segments and are here to stay; 2.5D-SICs provide better chip cooling options and hence typically target high-performance computing and networking applications, whereas 3D-SICs with their small footprint are better suited for mobile applications.

In order to obtain acceptable compound stack yields, there is a need to perform *pre-bond testing* of the various dies before stacking [5, 6]. For non-bottom dies in the stack, the typical functional interface is through an array of fine-pitch micro-bumps. These micro-bumps are too small and too dense for conventional probe technology. Consequently, the current industrial approach to enable test access for pre-bond testing is to provide non-bottom dies with dedicated pre-bond probe pads [5, 7–9]. Although these dedicated probe pads achieve the job, they come at the expense of extra design effort, extra silicon area, possibly extra processing steps, extra test application time, extra load on the micro-bump I/Os during post-bond functional stack operation, and still leave the micro-bumps themselves untested.

In this work, we set out to directly probe on large-array fine-pitch micro-bumps. We are capable to do this at wafer level with a probe card in a single-site set-up. This enables a test flow in which the die's internal circuitry (logic, DRAM) is tested through dedicated pre-bond probe pads, possibly in a (massive) multi-site arrangement, and in which the micro-bumps and underlying TSVs are separately tested in a single-site set-up. It also enables an alternative test flow, in which the entire pre-bond test is performed single-site by probing directly on the micro-bumps; this will circumvent the need for dedicated pre-bond probe pads with all its associated drawbacks and costs.

Direct probing on fine-pitch micro-bumps requires advanced probe technology: fine-pitch low-force probe cards and accurate probe stations. Prior work in this domain has been reported by others [10–15] and by us [16–18], but, to the best of our knowledge, this is the first paper that reports on pre-bond contact resistance, probe marks on both top and landing micro-bumps, and impact on stack interconnect yield. In this paper, we are using the JEDEC Wide-I/O Mobile DRAM interface (JESD-229) [19–21] as a typical target for today's 2.5D- and 3D-SIC micro-bump arrays. We have designed and manufactured test wafers with this micro-bump interface and report on our experiences probing and subsequent stacking of that interface. We have used the 3D-COSTAR test flow cost modeling tool [22–25] to analyze the cost-effectiveness of our approach, in comparison to performing pre-bond testing through dedicated pre-bond probe pads.

The remainder of this paper is organized as follows. Section 2

* Part of the work of Erik Jan Marinissen and Jörg Kiesewetter has been performed in the project ESiP (winner of the ENIAC Innovation Award 2013), which was funded by the ENIAC Joint Undertaking (http://www.eniac.eu), and, for Jörg Kiesewetter, by the public authorities of Saxony. Part of the work of Erik Jan Marinissen, Bart De Wachter, and Jörg Kiesewetter is performed in the project SEA4KET (http://www.sea4ket.eu), sub-project 3DIMS; this project receives funding from the European Union's Seventh Programme for research, technological development, and demonstration under grant agreement No. IST-611332.

978-1-4799-4722-5/14/$31.00 ©2014 IEEE

discusses the importance of pre-bond testing. Section 3 describes the micro-bump probe targets. Section 4 details the selected probe technology, while Section 5 describes the test vehicle. Experiment results are given in Section 6. Our cost modeling case study is described in Section 7. Section 8 concludes this paper.

2 Pre-Bond Testing

The post-bond compound stack yield y_{stack} of a stack consisting of n dies cannot be greater than the product of the individual die yields y_d (for $1 \leq d \leq n$) and the interconnect yields y_i (for $1 \leq i \leq n-1$), where y_i is the yield of the interconnects between adjacent Dies i and $i+1$:

$$y_{\text{stack}} \leq \prod_{d=1}^{n} y_d \cdot \prod_{i=1}^{n-1} y_i \qquad (2.1)$$

Figure 1 plots the post-bond compound stack yield y_{stack} for varying die yields y_d for various values of n and y_i. The graph demonstrates that the compound stack yield decreases drastically if y_d decreases. Consequently, it is important to test dies before stacking (the so-called pre-bond test) and only stack dies passing that pre-bond test in a die-to-die or die-to-wafer scheme.

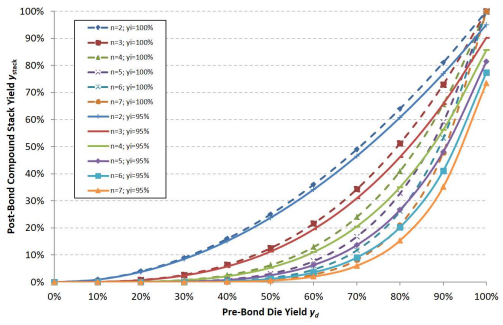


Figure 1: Post-bond compound stack yield y_{stack} as function of pre-bond die yield y_d for various stack heights n and various interconnect yields y_i.

Compared to skipping it, pre-bond testing obviously requires additional costs and the better the pre-bond test, the higher those costs will be. However, this investment typically pays off, as the alternative is that bad dies get detected only after stacking, at which point they are filtered out of the production flow together with the good dies to which they are now attached. Figure 2 shows an example of a total stack cost price calculation made with 3D-COSTAR [22–25]. We assumed a three-die stack, in which Die 1 was fully tested before stacking, but for which we varied the pre-bond test coverage and associated pre-bond test costs for Dies 2 and 3. The graph shows that more pre-bond testing (at assumed linearly increasing pre-bond test cost) actually decreases the overall stack cost price.

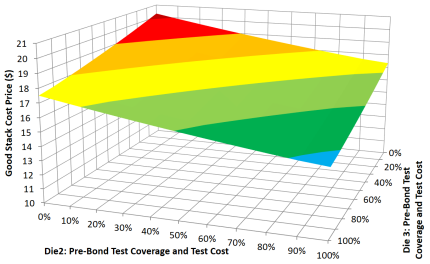


Figure 2: Good-stack cost price for a three-die stack as function of pre-bond test coverage and associated test cost for Dies 2 and 3.

Test access for pre-bond testing is through probing. Probing the bottom die of a stack is relatively easy, as the natural interface to the package substrate is implemented with large C4 bumps or copper pillars; a typical diameter is 50μm at 120μm pitch, which is no problem for today's probe technology. However, this is not true for the non-bottom dies; see Figure 3. All their functional connections (for power, ground, control, clocks, digital, analog, etc.) go through large arrays of fine-pitch micro-bumps. Typical micro-bumps have a diameter of $\sim$20μm at 40μm pitch and come in arrays of several hundreds to thousands of micro-bumps. Cantilever probe cards can achieve these small pitches, but cannot handle such large arrays. Vertical probe cards can be made in arbitrary array configurations, but are limited to pitches around 60μm.

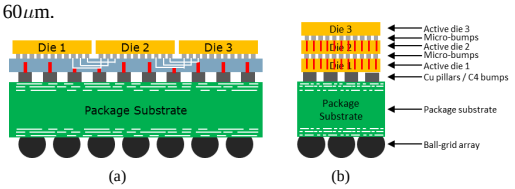


Figure 3: Cross-sections of typical (a) 2.5D- and (b) 3D-SICs containing three active dies.

Today's solution in the industry is to equip non-bottom dies with dedicated pre-bond probe pads, with sufficiently large size and pitch to accommodate today's probe technology [5, 7–9]. This solution requires extra design effort and possibly extra processing steps. Moreover, it causes a trade-off between extra silicon area and extra test time. The probe pads are larger than the micro-bumps; that is their whole purpose. Hence, typically one cannot afford as many probe pads as there are micro-bumps, as they would simply consume too much silicon area. As a result, the same pre-bond stimulus/response data needs to be pumped in and out of the die-under-test through a narrower interface and consequently the die's pre-bond test time smears out over more clock cycles, increasing the pre-bond test application cost. Furthermore, after performing a pre-bond test through dedicated probe pads, one can still not be certain of the correct operation of the functional interface through the micro-bumps. Finally, the dedicated pre-bond probe pads cause an extra capacitive load on the micro-bump I/Os during post-bond functional stack operation.

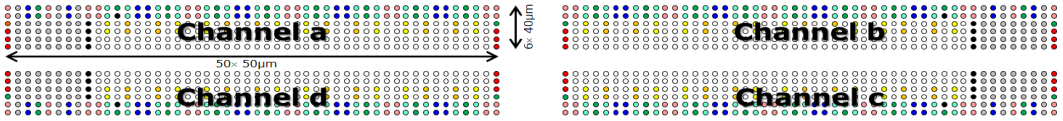
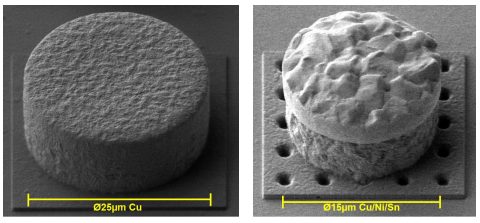


Figure 4: Standardized micro-bump lay-out according to the JEDEC Wide-I/O Mobile DRAM specification [19].

3 Micro-Bumps

Micro-bumps come in different metallurgies, forms, and shapes. IMEC's 40μm-pitch micro-bumps reference process utilizes copper (Cu) landing bumps of 9μm height and 25μm diameter and copper-nickel-tin (Cu/Ni/Sn) top bumps of 9μm height and 15μm diameter [26]. Two such micro-bumps are depicted in Figure 5. The micro-bumps have a cylindrical shape. As can be seen, the Cu micro-bumps have a rather smooth surface. As no reflow was applied (yet) on the Cu/Ni/Sn micro-bumps, their surface is significantly more rough. During stacking, the two micro-bumps form an intermetallic bond under thermo-compression.



(a) Cu micro-bump, $\varnothing$25μm (b) Cu/Ni/Sn micro-bump, $\varnothing$15μm

Figure 5: Typical micro-bumps at IMEC: (a) copper landing bump of 25μm diameter and (b) copper-nickel-tin top bump of 15μm diameter.

Micro-bumps typically come in large arrays. For this work, we took as target the representative micro-bump array of the JEDEC Wide-I/O Mobile DRAM standard [19–21]. This first standard for stackable Wide-I/O DRAMs, published as JESD-229 in December 2011, defines the functional and mechanical aspects of the Wide-I/O logic-memory interface. The interface consists of four DRAM channels (named a, b, c, and d), each consisting of an array of 6 rows $\times$ 50 columns = 300 micro-bumps with a horizontal pitch of 50μm and a vertical pitch of 40μm. The pad locations are symmetric between the four channels and also the spacing between the four channels is defined. The total interface occupies 0.52mm $\times$ 5.25mm. Figure 4 shows the lay-out of the 1,200 JEDEC Wide-I/O micro-bumps.

Direct probing on large arrays of fine-pitch micro-bumps has to meet the following criteria.

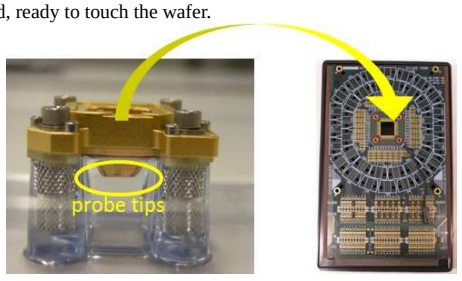
- *Good electrical contact with low contact resistance*, to allow for pre-bond testing of the die-under-test. We used as specification a contact resistance $<5\Omega$.
- *Probe marks with a limited profile*, to not impair downstream bonding or negatively impact the yield of that bonding process. We used as specification a probe mark profile $<$500nm.
- *Affordable test cost*, i.e., the cost of the required probe technology should not be excessive. We address this issue in Section 7.

4 Probe Technology

4.1 Probe Cards

Conventional probe cards are insufficient to probe on large-array fine-pitch micro-bumps, such as specified by JEDEC's Wide-I/O Mobile DRAM interface. Traditional cantilever probe cards do not come in the required array size, and vertical probe cards do not come in the required fine pitch. Hence, we needed to turn to advanced MEMS-type probe cards.

We have used the second generation of Cascade Microtech's Pyramid Probe® technology, named Rocking Beam Interposer (RBI), which is currently in its development phase. As depicted in Figure 6, this technology has a modular set-up comprising two components: the probe *core* and the probe *card*. The probe core contains the IC-design specific probe tips, whereas the probe card fits to the probe station-specific probe card holder. The probe core's rectangular frame has a screw in each of its four corners, with which it is screwed right on top of the hole in the middle of the probe card, such that the core's probe tips stick out under the probe card, ready to touch the wafer.



(a) Probe core (b) Probe card

Figure 6: Modular Pyramid Probe set-up consisting of (a) probe core and (b) probe card.

The Pyramid Probe RBI is a vertical, non-see-through probe technology. Two thin-film membranes are spanned across a 'plunger' with adjustable spring. The first membrane contains the routing layer from probe card I/Os to probe tips and vice-versa, whereas

the second (= outer) membrane contains the RBI probe tips; see Figure 7 [18]. The MEMS-type probe tips have a square probe surface of $6\times6\mu m^2$ and are placed on a rocking beam that connects to the upper routing-layer membrane through a copper post. Figure 8(a) shows a top view of an array of probe tips, while Figure 8(b) shows a cross-section of a single probe tip.

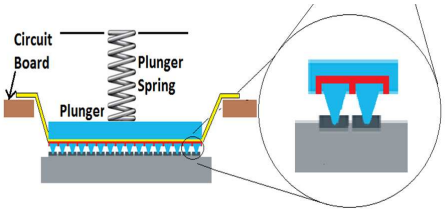
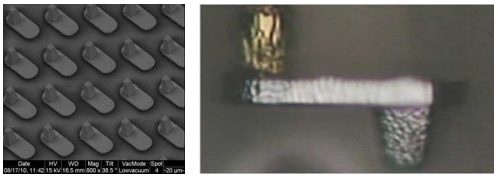


Figure 7: Conceptual cross-section view of the probe core [18].



(a) Top view (b) Cross section view

Figure 8: Pyramid Probe® RBI tips.

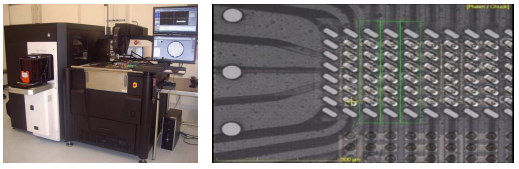
The Pyramid Probe® RBI technology has three main benefits: (1) it can be manufactured in large arrays at fine pitches, down to $20\mu m$; (2) the probe tips exercise a low probe force of up to 1 gram force per tip at a user-defined chuck over-travel, thereby inflicting minimal probe mark damage; and (3) the separate tip-layer coupon allows easy repair of inadvertently damaged probe tips.

4.2 Probe Station

The selection of a probe station had to fulfill three main criteria.

- The probe station needs to be able to work in a clean-room environment, as the stacking operations which follow after pre-bond testing are also performed in a clean-room environment and hence wafers/dies cannot be contaminated.
- The probe station has to be able to work with non-see-through vertical probe cards, which implies overlay *probe-to-pad alignment* (PTPA) in the presence of upward-looking (to the probe tips) and downward-looking (to the micro-bumps on the wafer) cameras (see Figure 9(b)).
- The probe station has to have a x, y, and θ touch-down accuracy and stepping accuracy sufficient to work with the small diameters and pitches of our micro-bump arrays.

We selected the Cascade Microtech CM300 probe station for our task. This is a brand-new prober platform with features for measurement accuracy and unattended testing inherited from the Cascade Microtech Elite300 and the Süss MicroTec MicroAlign probers respectively. Installed in IMEC's 300mm clean-room is the world's first demonstrator prototype of this probe station along with an auto-loader and material handling unit (see Figure 9(a)).



(a) (b)

Figure 9: CM300 probe station in IMEC's 300mm clean-room (a) and the prober's tip training software in action on the Wide-I/O probe core (b).

5 Test Vehicle: Vesuvius-2.5D

The IMEC-designed test vehicle for our Wide-I/O direct probing experiments is named Vesuvius-2.5D. It consists of two active Vesuvius test chips, stacked face-down side-by-side on a passive Interposer die. Figure 10 shows a picture of a single Vesuvius-2.5D die stack.

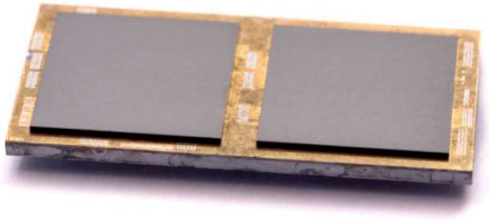


Figure 10: Vesuvius-2.5D die stack photo.

Figure 11 shows the various dimensions of the Vesuvius-2.5D die stack, both in top and cross-section view. The two Vesuvius dies atop measure $8.1\times8.1mm^2$ in a custom technology consisting of 65nm CMOS and five metal layers manufactured by GLOBALFOUNDRIES, and Cu/Ni/Sn micro-bumps (as described in Section 3) manufactured by IMEC. The bottom Interposer die measures $10\times20mm^2$ in an experimental silicon interposer technology containing four metal layers, $10\times100\mu m$ 'via-middle' TSVs and Cu micro-bumps (as described in Section 3), developed and manufactured by IMEC [27].

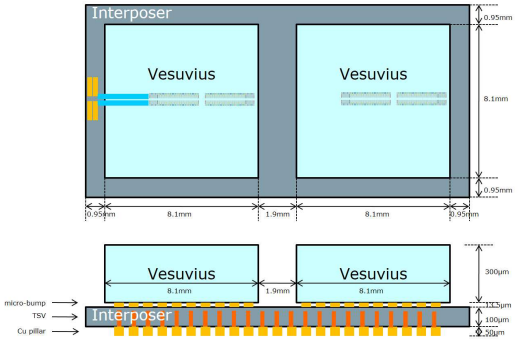


Figure 11: Vesuvius-2.5D test vehicle top view and cross-section.

Each Vesuvius die contains many test structures [28], including one full JEDEC-compliant four-channel Wide-I/O micro-bump interface [19]. Each Wide-I/O channel of 300 micro-bumps is divided in ten equal groups of 30 micro-bumps each, which after stacking form an up-down daisy-chain between Vesuvius and Interposer dies; see Figure 12. Hence, there are in total 40 daisy-chains for the four Wide-I/O channels. As depicted in Figure 11, the daisy-chains in the left-hand Vesuvius die are routed through the Interposer die to regularly-sized ($80\times60\mu m^2$) probe pads on the Interposer front-side to the left of the left-most Vesuvius die.

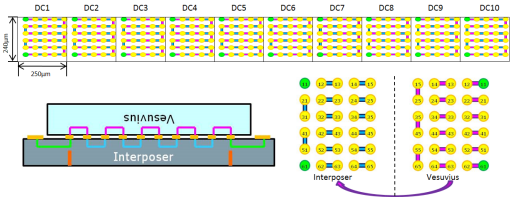
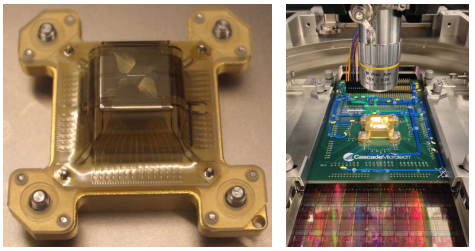


Figure 12: Vesuvius-Interposer daisy-chain consisting of 30 adjacent micro-bumps and metal interconnects.

The Pyramid RBI probe core (shown in Figure 13(a)) designed for our test vehicle is a so-called MSI core for exactly one Wide-I/O channel. The reason to probe on a single Wide-I/O channel (and not all four) is rooted in experimental considerations; this set-up allows us to evaluate the impact of probe marks on stack interconnect yield for all four cases: probed on top and bottom, probed only on either top or bottom, or not probed at all. The probe core routes all 300 probe tips out to the corresponding core-to-card I/Os; therefore the same probe core can be used on both Interposer and Vesuvius dies. We have two 4.5-inch rectangular engineering-type probe cards for our CM300 probe station: one which completes the ten daisy-chains when probing on an Interposer die (depicted in Figure 13(b)) and another one which completes the ten daisy-chains when probing on a Vesuvius die. Concatenating 30 micro-bumps in a daisy-chain limits the resolution of the probe-to-bump contact resistance that can be measured, but this design decision was due to a limitation in the number of available tester channels.



(a) (b)

Figure 13: Probe core for single-channel Wide-I/O interface (a) probe tips face-up and (b) attached to a probe card, in action in our probe station.

We have defined three test phases for Vesuvius-2.5D probing.

- In *Test Phase 1* we use the Pyramid RBI probe core with the dedicated Interposer probe card to probe on Wide-I/O channels a and b of the pre-bond Interposer dies in two subsequent touch-downs. We check the landing of the probe tips on the 25μm-diameter Cu micro-bumps, both by means of visual and scanning electron microscope (SEM) inspection of the probe marks, as well as by electrical continuity of the probe card-to-wafer daisy-chains.

- In *Test Phase 2* we use the Pyramid RBI probe core with the dedicated Vesuvius probe card to probe on Wide-I/O channels b and c of the pre-bond Vesuvius dies in two subsequent touch-downs. We verify the landing of the probe tips on the 15μm-diameter Cu/Ni/Sn micro-bumps, both by means of visual and SEM inspection of the probe marks, as well as by electrical continuity of the probe card-to-wafer daisy-chains.

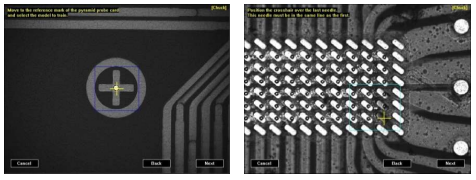
- In *Test Phase 3*, after stacking, we assess the impact of the micro-bump probing on the interconnect yield. We use a conventional cantilever probe card to probe on the regularly-sized post-bond probe pads on the front-side of the Interposer and measure the electrical continuity of the various micro-bump daisy-chains. In this, we can compare the four channels:

  - Channel a: only Interposer probed
  - Channel b: Vesuvius and Interposer both probed
  - Channel c: only Vesuvius probed
  - Channel d: no micro-bumps probed.

6 Experiment Results

6.1 Initial Hurdles

Initially, the PTPA software of the CM300 probe station was not optimally suited for automatically recognizing the Pyramid Probe RBI probe tips. The software was made to handle conventional cantilever and vertical probe needles, but turned out not to work reliably and repeatedly for the small and very different RBI probe tips. We developed a dedicated probe tip recognition routine for the RBI tips, which consists of three steps. The pattern recognition uses (1) the large cross-hair fiducials included on the four corners of the probe core's membrane (see Figure 14(a)), (2) dummy bumps on the probe membrane, and (3) two probe tips, typically located on opposite extremes of the probe tip array (see Figure 14(b)). With the deployment of the new software routine, the automatic probe tip recognition works without problems.



(a) Step 1 (b) Steps 2+3

Figure 14: Dedicated probe tip recognition routine for RBI probe tips.

During probing operation, probe tips pick up dirt, which increases the contact resistance and ultimately might obstruct electrical contact completely. Hence, probe tips need to be cleaned at regular intervals. This is also true for RBI probe tips. The recommended cleaning medium for RBI tips is a tungsten-carbide (WC) substrate, on which the to-be-cleaned tips need to touch down with regular over-travel at 15 fresh locations. For holding a cleaning substrate, the CM300 probe station is equipped with various vacuum-providing auxiliary chucks, positioned just outside the main wafer chuck (see Figure 15). The initial software version of the CM300 prober did not support usage of these auxiliary chucks with non-see-through probe cards like the Pyramid Probe RBI. This obstacle for RBI probe tip cleaning was quickly resolved in a new software version, and now the probe station can perform automatic probe tip cleaning after a user-defined number of touch-downs.

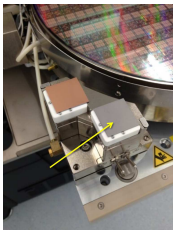
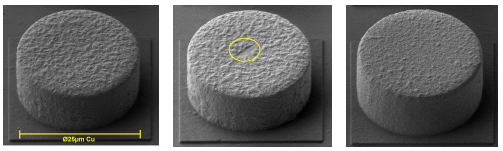


Figure 15: Tungsten-carbide (WC) tip cleaning substrate on CM300's auxiliary chuck.

6.2 Probe Marks

For given probe tip material and shape, the resulting probe marks depend on the chuck over-travel and the micro-bump metallurgies. All our experiments were performed at 150μm over-travel, which corresponds to 1gf/tip for the global plunger spring in the RBI probe cores we used.

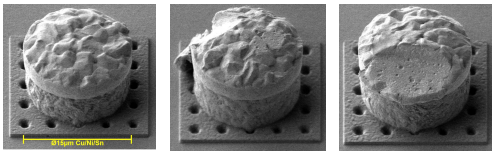
Test Phase 1 – Figure 16 shows SEM pictures of 25μm-diameter Cu landing micro-bumps before and after probing. Figure 16(a) shows such a Cu micro-bump *before* probing; the micro-bump is cylindrical in shape with a very smooth surface. Figure 16(b) shows a similar micro-bump *after* probing. On all such Cu micro-bumps, the probe marks are very uniform: a diagonal line of approximately $6\times1\mu m^2$, caused by the heel of the diagonally placed probe tip which itself measures $6\times6\mu m^2$. The probe mark is very shallow, on the order of the surface roughness of the Cu micro-bump. We do not expect any negative impact of the probe mark on the interconnect yield. Figure 16(c) shows a probed Cu micro-bump equipped with a 10nm-thick nickel-boron (NiB) cap. This NiB cap is meant to prevent the Cu micro-bump from oxidizing and thus improve the stack interconnect yield. The NiB cap is quite hard and consequently hardly any probe mark can be seen, although proper electrical contact was made [29].



(a) Before probing (b) After (c) After (NiB cap)

Figure 16: Probe marks on 25μm-diameter Cu landing micro-bumps.

Test Phase 2 – Figure 17 shows SEM pictures of non-reflowed 15μm-diameter Cu/Ni/Sn top micro-bumps before and after probing. Figure 17(a) shows such a Cu/Ni/Sn micro-bump *before* probing and reflow; the micro-bump is cylindrical in shape with a rather rough Sn surface. Figures 17(b) and 17(c) show two similar

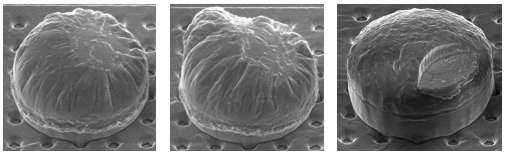


(a) Before probing (b) After – 1 (c) After – 2

Figure 17: Probe marks on non-reflowed 15μm-diameter Cu/Ni/Sn top micro-bumps.

micro-bumps *after* probing. On the softer Sn material, the probe mark is significantly larger than on the much harder Cu micro-bumps.

Figure 18 shows SEM pictures of reflowed 15μm-diameter Cu/Ni/Sn top micro-bumps before and after probing. Figure 18(a) shows such a Cu/Ni/Sn micro-bump which was only reflowed, and *not* probed; the originally rough horizontal Sn surface (as seen in Figure 17(a)) has been transformed by the reflow process in a dome-shaped cap. Figure 18(b) shows a Cu/Ni/Sn micro-bump which was first probed and subsequently reflowed. The hope was that the reflow process would eliminate the probe mark, but as can be seen from the figure, this is not entirely the case. Finally, Figure 18(c) shows a Cu/Ni/Sn micro-bump which was first reflowed and subsequently probed; as expected, the probe mark is clearly visible in the otherwise nicely smooth dome-shaped Sn cap. The remaining probe marks in Figures 18(b) and 18(c) could potentially form a location for particle or filler entrapment and hence negatively affect the bond's reliability. The smallest risk for this to happen is in the scenario where micro-bump probing precedes the reflow operation [30].



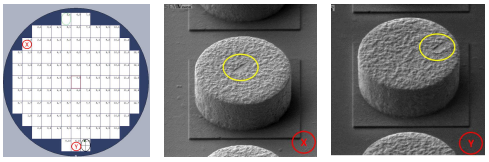
(a) Only reflowed (b) Probed; reflowed (c) Reflowed; probed

Figure 18: Probe marks on reflowed 15μm-diameter Cu/Ni/Sn top micro-bumps.

6.3 PTPA Accuracy

We want to minimize the probe mark damage to the micro-bump, in order not to negatively impact the downstream bonding yield; from that viewpoint, obtaining good electrical contact while leaving no visible probe mark is ideal. On the other hand, visible probe marks form reassuring evidence that a micro-bump was actually touched by a probe tip and allow us to determine the PTPA accuracy of the probe station.

To analyze the initial stepping accuracy of the CM300 demonstrator probe station, we stepped over all 111 dies in a 300mm Interposer wafer, starting top-left and zig-zagging down row-by-row until bottom-right, performing two touch-downs per Interposer die (on channels a and b). Analysis of the probe mark locations showed that all touch-downs were *on* the 25μm-diameter Cu micro-bumps, indicating that the stepping accuracy of the probe station was sufficient for Test Phase 1. There was little variation detected in the y-axis. However, the maximum variation in the x-axis was between Die X (-1μm, see Figure 19(b)) and Die Y (+6.2μm, see Figure 19(c)) (die locations indicated on the wafer map (Figure 19(a)). This was considered too inaccurate, as the probe mark was getting close to the micro-bump edge.



(a) Interposer wafer map (b) Probe mark X (c) Probe mark Y

Figure 19: Interposer wafer map (a) with Dies X (b) and Y (c) that showed the left-most resp. right-most probe mark location variation.

Cascade Microtech and IMEC have jointly taken several steps to improve the PTPA accuracy.

- *Probe card adapter*.
  The MSI-sized RBI probe cores have a vertical z height (*'draft'*) of 11.1mm measured from the top side of the probe card. This is higher than many other probe cards, due to which the probe tips ended up below the field-of-view of the probe station's side-view camera, which is meant to assist in the touch-down procedure. We had to lift the probe card in order to bring the probe tips back in sight of the side-view camera. Initially, this lifting was achieved with some dedicated shims. To further improve PTPA stability, this temporary workaround has been replaced with a new probe card adapter with larger z lift.
- *Thermal stability*.
  The ambient temperature in our clean-room is 22°C. However, during operation, the probe chamber of the probe station heats up to 30°C. This temperature increase can lead to a maximum radial wafer expansion of

$$\begin{aligned} r \cdot \Delta t \cdot CTE_{\mathrm{Si}} &= 150mm \cdot 8^{\circ}\mathrm{C} \cdot 2.6 \times 10^{-6}/^{\circ}\mathrm{C} \\ &= 3.12\mu m \end{aligned} \quad (6.1)$$

  where r is the wafer radius (in mm), Δt the temperature increase (in °C), and CTE_{Si} the coefficient of thermal expansion of silicon (in ppm/°C). Direct micro-bump probing requires thermal stability to avoid wafer expansion, and therefore we keep the chiller that controls the thermo-chuck on at 22°C.
- *Automatic ReAlign*.
  The CM300 software Velox has an option for automatic ReAlign after n touch-downs, with n a user-defined parameter. This software feature is mainly meant for temperature testing, where different thermal expansions of wafer and probe system might necessitate its usage. However, for fine-pitch micro-bump probing at stable ambient temperatures, the feature also provided great benefits. We have used it with good results after every 20 touch-downs. The ReAlign routine takes some time to execute (about 30 seconds in our case) and hence its usage slightly increases the overall test time. Therefore, the trade-off between touch-down accuracy and test time can be optimized.

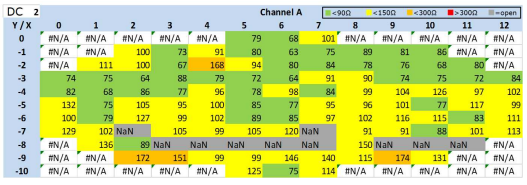
Right now, the PTPA accuracy requirements are satisfied, as the electrical measurement results in the following section confirm.

6.4 Contact Resistance

Proper electrical contact of the RBI probe tips on the micro-bumps was analyzed by performing two-point resistance measurements of the daisy-chains through probe card and wafer, for all ten daisy-chains per touch-down. Due to a probe card fault, initially daisy-chains DC6 and DC9 were found to be consistently non-continuous. This was quickly diagnosed as a problem in the probe card wiring and fixed. From this moment onward, all daisy-chains were continuous, apart from confirmed bad dies. This demonstrated that the probe tips all make proper contact to the micro-bumps.

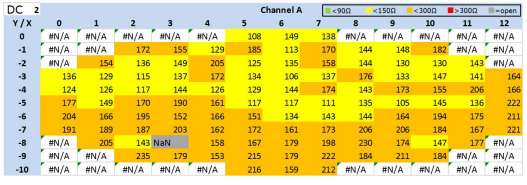
Figures 20, 21, and 22 show three representative wafer maps of two-point resistance measurements through a 30-long micro-bump-to-probe-tip daisy-chain of a DRAM channel on a micro-bumped wafer. The colors in the wafer map bin the daisy-chain resistance value R into three bins: (1) green: $R < 90\Omega$, (2) yellow: $90\Omega \leq R < 150\Omega$, (3) orange: $150\Omega \leq R < 300\Omega$, (4) red: $300\Omega \leq R$, (5) gray: $R = \infty$ ("Not-A-Number" = daisy-chain non-continuous).

Test Phase 1 – Figure 20 shows the wafer-map for DC2 of Channel a on an Interposer wafer with 25μm-diameter Cu micro-bumps. There are 111 dies on this 300mm wafer. Most daisy-chains are continuous. The non-continuous daisy-chains were confirmed (through other tests on the same dies) to be caused by wafer manufacturing issues on these particular dies. For this particular wafer, the median daisy-chain resistance was 118Ω, resulting in 3.9Ω per micro-bump.



Y / X	0	1	2	3	4	5	6	7	8	9	10	11	12
0	#N/A	#N/A	#N/A	#N/A	#N/A	79	68	101	#N/A	#N/A	#N/A	#N/A	#N/A
-1	#N/A	#N/A	100	73	91	80	63	75	89	81	86	#N/A	#N/A
-2	#N/A	111	100	67	168	94	80	84	78	76	68	80	#N/A
-3	74	75	64	88	79	72	64	91	90	74	75	72	84
-4	82	68	86	77	96	78	98	84	99	104	126	97	102
-5	132	75	105	95	100	85	77	95	96	101	77	117	99
-6	100	79	127	99	102	89	85	97	102	116	115	83	111
-7	129	102	NaN	105	99	105	120	NaN	91	91	88	101	113
-8	#N/A	136	89	NaN	NaN	NaN	NaN	NaN	150	NaN	NaN	NaN	#N/A
-9	#N/A	#N/A	172	151	99	99	146	140	115	174	131	#N/A	#N/A
-10	#N/A	#N/A	#N/A	#N/A	#N/A	125	75	114	#N/A	#N/A	#N/A	#N/A	#N/A

Figure 20: Wafer map with two-point resistance measurement values for DC 2 of Channel a on an Interposer wafer with Cu micro-bumps.



Y / X	0	1	2	3	4	5	6	7	8	9	10	11	12
0	#N/A	#N/A	#N/A	#N/A	#N/A	108	149	138	#N/A	#N/A	#N/A	#N/A	#N/A
-1	#N/A	#N/A	172	155	129	185	113	170	144	148	182	#N/A	#N/A
-2	#N/A	154	136	149	205	125	135	158	144	130	130	143	#N/A
-3	136	129	115	137	172	134	106	137	176	133	147	141	164
-4	124	126	117	144	126	129	144	174	143	173	155	206	166
-5	177	149	170	190	161	117	117	111	135	105	145	136	222
-6	204	166	195	152	166	151	134	143	144	164	194	175	211
-7	191	189	187	203	162	172	161	173	206	206	184	167	221
-8	#N/A	205	143	NaN	158	167	179	198	230	174	147	177	#N/A
-9	#N/A	#N/A	235	179	153	215	179	222	184	211	184	#N/A	#N/A
-10	#N/A	#N/A	#N/A	#N/A	#N/A	216	159	212	#N/A	#N/A	#N/A	#N/A	#N/A

Figure 21: Wafer map with two-point resistance measurement values for DC 2 of Channel a on an Interposer wafer with Cu micro-bumps with a NiB cap.

Figure 21 shows the wafer-map for DC2 of Channel a on an Interposer wafer with 25μm-diameter Cu micro-bumps with NiB cap. All daisy-chains (apart from one) are continuous. The NiB cap clearly increases the daisy-chain resistance. For this particular wafer, the median daisy-chain resistance was 170Ω, resulting in 5.7Ω per micro-bump.

Test Phase 2 – Figure 22 shows the wafer-map for DC3 of Channel b on a Vesuvius wafer with 15μm-diameter Cu/Ni/Sn micro-bumps. There are 255 dies on this 300mm wafer; due to a technical error, the testing was aborted half-way the last-but-one row at the bottom of the wafer. Most daisy-chains are continuous. The non-continuous daisy-chains were confirmed (through other tests on the same dies) to be caused by wafer manufacturing issues on these particular dies. For this particular wafer, the median daisy-chain resistance was 98Ω, resulting in 3.3Ω per micro-bump.

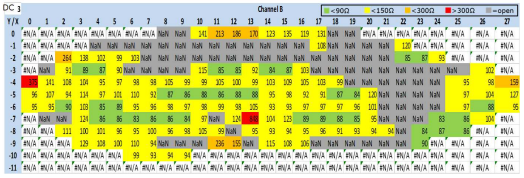


Figure 22: Wafer map with two-point resistance measurement values for DC 3 of Channel b on a Vesuvius wafer with non-reflowed Cu/Ni/Sn micro-bumps.

6.5 Probe Impact on Stack Interconnect Yield

In Test Phase 3, we verified the impact of the probe marks on stack interconnect yield. Table 1 lists the interconnect yields for 320 daisy-chains, 80 of each type. The table shows *no* significant impact. The differences between channels a–d are all explained by the variation in sheet resistance of the Interposer wires between Wide-I/O micro-bumps and post-bond probe pads, due to variations in lay-out locations of the micro-bumps.

Wide-I/O Channel	***a***	***b***	***c***	***d***
Vesuvius probed	no	yes	yes	no
Interposer probed	yes	yes	no	no
Interconnect yield	100%	100%	100%	100%
Daisychain resistance R	32.0Ω	42.4Ω	45.0Ω	33.1Ω
Std. deviation R	9.8Ω	5.8Ω	9.2Ω	8.2Ω

Table 1: Interconnect yield in Test Phase 3.

7 Cost Modeling Case Study

TU Delft and IMEC have developed a software tool named 3D-COSTAR to analyze product quality and test cost trade-offs in the many possible 3D test flows [22–25]. The tool uses as inputs lump-sum cost numbers for (1) design, (2) manufacturing, (3) test, (4) packaging, and (5) logistics. It models many different stacking approaches: simple linear 3D stacks, 2.5D stacks, complex multi-tower stacks, D2D/D2W/W2W stacking, etc. It assumes that no

manufacturing process is perfect and takes into account yields (in %) of die processing, interconnect layers, stacking, and packaging, as well as test coverage (in %) and test escape rates (in ppm). Furthermore, it attributes all costs made along the way to the end-of-line passing products.

We have used 3D-COSTAR to analyze the cost-effectiveness of our direct micro-bump probing approach, as alternative to performing pre-bond testing through dedicated pre-bond probe pads. In this cost modeling case study, we compare two scenarios.

1. *Probing through dedicated pre-bond probe pads.*
   These probe pads are by definition larger than the fine-pitch micro-bumps in order to allow probing on them with conventional probe technology. Consequently, they present a trade-off between the number of probe pads and corresponding silicon area on one hand, and the test input/output bandwidth provided and corresponding test time on the other hand (assuming a constant test data volume that needs to be pumped in and out of the device-under-test).

2. *Direct probing on micro-bumps.*
   This will require advanced (and hence expensive) probe cards, and hypothetically the micro-bump probe marks might decrease the interconnect yield after stacking.

Table 2 lists some of 3D-COSTAR's key cost model parameters. Note that there are many more parameters, which are not shown. The stack set-up and die sizes are inspired by the Vesuvius-2.5D test vehicle as described in Section 5: two active dies of 8.1×8.1=65.61mm^2 stacked on side-by-side on top of a passive interposer die of 10×20=200mm^2. We assume single-site testing on 300mm wafers with 3mm edge clearance. Unlike what was the case on our actual test wafers, we assume that the entire wafers are populated with only Vesuvius and Interposer dies respectively. The Interposer technology is assumed to be relatively cheap and mature; its defect density is fixed at 0.1 defects/cm^2. On the other hand, the active dies are assumed to be in an advanced technology node and hence have relatively expensive wafers; their defect density is varied from 0.0 to 1.0 defects/cm^2.

Parameter	**Interposer**	**Scenario 1** **Die 1+2**	**Scenario 2** **Die 1+2**
Pre-bond test contacts	n.a.	120	1200
300mm wafer cost	$ 700	$ 3000	$ 3000
Die area	200mm^2	66.61mm^2	65.61mm^2
Gross die / wafer	302	953	968
Defect density	0.1/cm^2	0.0–1.0/cm^2	0.0–1.0/cm^2
Die yield	84.52%	100–65.48%	100–65.76%
Pre-bond fault coverage	n.a.	99%	99%
Pre-bond test time	n.a.	100s	10s
Pre-bond probe card cost / die	n.a.	$ 0.00	$ 0.50
Pre-bond test cost	n.a.	$ 5.00	$ 1.00
Stack interconnect yield	100%	99%	98%
Final fault coverage	100%	99%	99%
Final test time	1s	10s	10s
Final test cost	$ 0.05	$ 0.50	$ 0.50

Table 2: Some key cost model parameters for the two test scenarios.

The case study concentrates on the pre-bond test of the active dies, and hence we modeled a test flow in which there is no pre-bond test for the Interposer die. We assume each of these active dies has a JEDEC Wide-I/O compliant micro-bump interface of 1,200 micro-bumps [19]. In Scenario 1, we are providing extra dedicated pre-bond probe pads. As we do not want to implement as many as 1,200 extra probe pads, we are assuming that we provide 120 extra probe pads only; we optimistically assume that this leads to only a $10\times$ increase in pre-bond test application time and hence test cost. In Scenario 2, we probe directly on the 1,200 micro-bumps. For this we need an expensive advanced probe card. We assume pessimistically its lifetime to be 100k touch-downs and its cost to be $50k. Assuming a single touch-down per die, this advanced probe card alone adds $0.50 costs to each die tested, on top of the assumed $0.05/s test cost. In addition, we pessimistically assume that its probe marks on the micro-bumps deteriorate the interconnect yield after stacking from 99% down to 98%. We have underlined the main differences between the two scenarios under comparison.

Figure 23 shows the test cost and the total stack cost per good stack for varying defect density of the two active dies for both scenarios. In all cases, direct probing on micro-bumps is cheaper than testing through dedicated pre-bond probe pads. The main differentiator for Scenario 1 is the $10\times$ increase in test time and hence test application costs, which makes pre-bond test a significant cost contribution in the overall stack cost price. Die yield works as a multiplier, as for example at 50% yield, two dies need to be manufactured and tested to find one good one, whose cost price should carry the cost of both dies. The increased area for dedicated pads, the expensive cost of the advanced probe card, and the (pessimistic) yield loss are all minor contributors in the overall cost calculation.

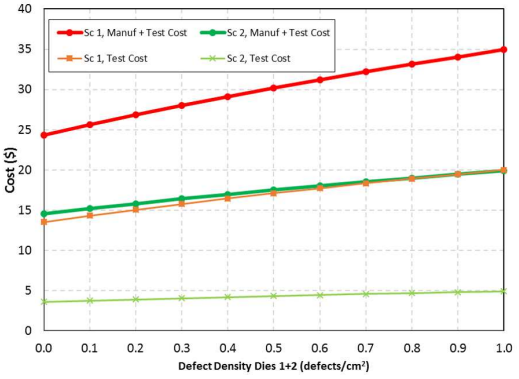


Figure 23: 3D-COSTAR test cost and total cost results.

Figure 24 shows the number of test escapes for both scenarios. In all cases, direct probing on micro-bumps results in a slightly lower number of test escapes except for the case where the defect density

value equals zero; here, the active dies have a 100% yield for both scenarios. Note that the interconnect yield, which is different for both scenarios, has no impact on the test escapes as the interconnect fault coverage is assumed to be 100% during final test.

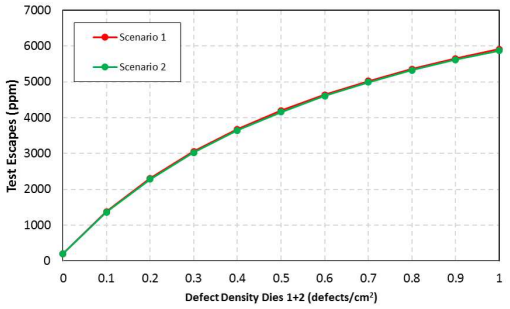


Figure 24: 3D-COSTAR test escape results.

8 Conclusion

In this paper we discussed direct probing of large-array fine-pitch micro-bumps in the context of 2.5D- and 3D-SICs. We have successfully conducted wafer-level direct probe experiments on single-channels of the JEDEC Wide-I/O Mobile DRAM interface, consisting of 6×50 arrays of 25μm-diameter Cu micro-bumps and 15μm-diameter Cu/Ni/Sn micro-bumps at 40/50μm pitches. Our experiments have shown the technical feasibility of the direct probing approach, with probe tips making proper electrical contact to the micro-bumps (i.e., contact resistance $<5\Omega$), causing only limited probe marks (i.e., probe mark profile $<$500nm, for Cu/Ni/Sn micro-bumps obtained with a post-probing reflow operation), and no measureable impact on stack interconnect yield. Our cost modeling indicates economical feasibility for single-site testing. The next step is to prepare this technology for volume production.

Acknowledgments

The authors thank many colleagues. At IMEC: Filip Beirnaert, Gerald Beyer, Eric Beyne, Kristof Croes, Miroslav Cupak, Inge De Preter, Jaber Derakhshandeh, Mikael Detalle, Joeri De Vos, Luc Dupas, Luke England, Antonio La Manna, Mireille Matterne, Julien Ryckaert, Michele Stucchi, George Vakanas, Marc Van Dievel, Geert Van der Plas, Joris Van Laer, and Dimitrios Velenis. At Cascade Microtech: Debbora Ahlgren, Juliane Busch, Claus Dietrich, Jens Fiedler, Gavin Fisher, Steve Harris, Ulf Hackius, Geert-Jan Hendricks, Torsten Kern, Clint VanderGiessen, and Eric Wilms.

References

[1] Erik Jan Marinissen. Challenges and Emerging Solutions in Testing TSV-Based 2.5D- and 3D-Stacked ICs. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1277–1282, March 2012. doi:10.1109/DATE.2012.6176689.

[2] Robert S. Patti. Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs. *Proceedings of the IEEE*, 94(6):1214–1224, June 2006. doi:10.1109/JPROC.2006.873612.

[3] Eric Beyne and Bart Swinnen. 3D System Integration Technologies. In *Proceedings of IEEE International Conference on Integrated Circuit Design and Technology (ICICDT)*, pages 1–3, June 2007. doi:10.1109/ICICDT.2007.4299568.

[4] Philip Garrou, Christopher Bower, and Peter Ramm, editors. *Handbook of 3D Integration – Technology and Applications of 3D Integrated Circuits*. Wiley-VCH, Weinheim, Germany, August 2008. ISBN 978-3-527-33265-6.

[5] Erik Jan Marinissen and Yervant Zorian. Testing 3D Chips Containing Through-Silicon Vias. In *Proceedings IEEE International Test Conference (ITC)*, November 2009. doi:10.1109/TEST.2009.5355573.

[6] Erik Jan Marinissen. Testing TSV-Based Three-Dimensional Stacked ICs. In *Proceedings Design, Automation, and Test in Europe (DATE)*, pages 1689–1694, March 2010. doi:10.1109/DATE.2010.5457087.

[7] Jung-Sik Kim et al. A 1.2V 12.8GB/s 2Gb Mobile Wide-I/O DRAM with 4×128 I/Os Using TSV-Based Stacking. In *Proceedings International Solid State Circuits Conference (ISSCC)*, pages 496–498, February 2011. doi:10.1109/ISSCC.2011.5746413.

[8] Christian Freund. Wide-IO DRAM – ST-Ericsson's First Mobile Processor Using TSV 3D-IC Technology. In *CDNLive! EMEA*, May 2011.

[9] Jung-Sik Kim et al. A 1.2 V 12.8 GB/s 2 Gb Mobile Wide-I/O DRAM With 4×128 I/Os Using TSV Based Stacking. *IEEE Journal of Solid-State Circuits*, 47(1):107–116, January 2012. doi:10.1109/JSSC.2011.2164731.

[10] Ben Eldridge and Marc Loranger. Challenges and Solutions for Testing of TSV and Micro-Bump. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, September 2011. Paper 7.2, `http://3dtest.tttc-events.org`.

[11] Matt Losey et al. A Low-Force MEMS Probe Solution for Fine-Pitch 3D-SIC Wafer Test. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, September 2011. Paper 7.3, `http://3dtest.tttc-events.org`.

[12] Onnik Yaglioglu and Ben Eldridge. Direct Connection and Testing of TSV and Microbump Devices Using NanoPierce Contactor for 3D-IC Integration. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 96–101, May 2012. doi:10.1109/VTS.2012.6231086.

[13] Joseph Foerstel and Amy Leong. 40μm Pitch Probing Evaluation. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, November 2012. Paper 5.1, `http://3dtest.tttc-events.org`.

[14] Gunther Böhm et al. Very Small Pitch Micro Bump Array Probing. In *Proceedings IEEE South-West Test Workshop (SWTW)*, June 2013. `http://www.swtest.org/swtw_library/2013proc/swtw2013.html`.

[15] Onnik Yaglioglu and Ben Eldridge. Contact Testing of Copper Micro-Pillars with Very Low Damage for 3D IC Assembly. In *Proceedings IEEE International Conference on 3D System Integration (3DIC)*, pages 1–4, October 2013. doi:10.1109/3DIC.2013.6702361.

[16] Ken Smith et al. KGD Probing of TSVs at 40μm Array Pitch. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, November 2010. Paper 4.1, `http://3dtest.tttc-events.org`.

[17] Erik Jan Marinissen et al. Wafer Probing on Fine-Pitch Micro-Bumps for 2.5D- and 3D-SICs. In *Proceedings IEEE South-West Test Workshop (SWTW)*, June 2011. `http://www.swtest.org/swtw_library/2011proc/swtw2011.html`.

[18] Ken Smith et al. Evaluation of TSV and Micro-Bump Probing for Wide I/O Testing. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, September 2011. doi:10.1109/TEST.2011.6139180.

[19] JEDEC. *Wide I/O Single Data Rate (JEDEC Standard JESD229)*. JEDEC Solid State Technology Association, December 2011. `http://www.jedec.org`.

[20] Sergej Deutsch et al. DfT Architecture and ATPG for Interconnect Tests of JEDEC Wide-I/O Memory-on-Logic Die Stacks. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, November 2012. doi:10.1109/TEST.2012.6401569.

[21] Sandeep K. Goel et al. Test and Debug Strategy for TSMC CoWoS Stacking Process Based Heterogeneous 3D IC: A Silicon Case Study. In *Proceedings IEEE International Test Conference (ITC)*, pages 1–10, September 2013. doi:10.1109/TEST.2013.6651893.

[22] Mottaqiallah Taouil et al. 3D-COSTAR: A Cost Model for 3D-SICs. In *Digest of IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-TEST)*, November 2012. Paper 10.2, `http://3dtest.tttc-events.org`.

[23] Mottaqiallah Taouil et al. 3D-COSTAR: A Cost Model for 3D-SICs. In *3-D Architectures for Semiconductor Integration and Packaging (3D-ASIP)*, December 2012.

[24] Mottaqiallah Taouil et al. Impact of Mid-Bond Testing in 3D Stacked ICs. In *Proceedings IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pages 178–183, October 2013. doi:10.1109/DFT.2013.6653603.

[25] Mottaqiallah Taouil et al. Using 3D-COSTAR for 2.5D Test Cost Optimization. In *Proceedings IEEE International Conference on 3D System Integration (3DIC)*, pages 1–8, October 2013. doi:10.1109/3DIC.2013.6702351.

[26] Joeri De Vos et al. Key Elements for Sub-50μm Pitch Micro Bump Processes. In *Proceedings IEEE Electronic Components and Technology Conference (ECTC)*, pages 1122–1126, May 2013. doi:10.1109/ECTC.2013.6575714.

[27] Mikael Detalle et al. Interposer Technology for High Bandwidth Interconnect Applications. In *Proceedings IEEE Electronic Components and Technology Conference (ECTC)*, pages 323–328, May 2013. doi:10.1109/ECTC.2013.6575590.

[28] Erik Jan Marinissen et al. Vesuvius-3D: A 3D-DfT Demonstrator. In *Proceedings IEEE International Test Conference (ITC)*, October 2014. Paper 20.2.

[29] Luke England et al. NiB Capping of Cu Landing Pads for Thermocompression Bonding. In *Workshop on Materials for Advanced Metallization*, March 2014.

[30] Jaber Derakhshandeh et al. Reflow Process Optimization for Micro-Bumps Applications in 3D Technology. In *Proceedings IEEE Electronics System-Integration Technology Conference (ESTC)*, September 2014.

Appendix C

Publications - Interconnect Testing and Diagnosis

This chapter presents the publications on cost modeling. The following papers are included:

C1: **M. Taouil**, M. Lefter, and S. Hamdioui, "Exploring Test Opportunities for Memory and Interconnects in 3D ICs," *International Design and Test Symposium (IDT)*, Marrakesh, Morocco, Dec. 2013, pp. 1–6.

C2: **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Interconnect Test for 3D Stacked Memory-on-Logic,"*Design, Automation & Test in Europe (DATE)*, Dresden, Germany, March 2014, pp. 1–6.

C3: **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Post-Bond Interconnect Test and Diagnosis for 3D Memory Stacked on Logic," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1–12, 2014*.

Exploring Test Opportunities for Memory and Interconnects in 3D ICs

Mottaqiallah Taouil Mihai Lefter Said Hamdioui
Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{M.Taouil, M.Lefter, S.Hamdioui}@tudelft.nl

***Abstract*— 3D-Stacked IC (3D-SIC) based on Through-Silicon-Vias (TSV) is an emerging technology that provides many benefits such as low power, high bandwidth 3D memories and heterogeneous integration. One of the attractive applications making used of such benefits is the stacking of memory dies on logic. System integrators for such application have to provide appropriate test strategy. However, they have to deal with block box IPs as IP providers usually refuse to share the IP content. Moreover, they dislike including JTAG in memory dies. Therefore, developing a low cost and high quality test approaches, while taking these constraints into consideration, is of great importance. This paper presents a framework of interconnect test approaches for memories stacked on logic, and look further than the only proposed JTAG solutions. The benefits and drawbacks of each possible solution is extensively discusses for stacked memories both with and without MBISTs, placed on the memory dies or on a separate logic die.**

Keywords: *iBIST, 3D Stacked IC, 3D Memory, Boundary Scan, Through-Silicon-Via*

I. Introduction

The popularity of 3D Stacked ICs (3D-SICs) is rising among industry and research institutes [1–4]. 3D-SICs are emerging as one of the main competitors to continue the trend of Moore's Law. Currently, a number of methods have been proposed to implement the interconnection of stacked dies. One of the most promising and perhaps the most reliable way to achieve this is with *Through Silicon Vias (TSVs)* [3]. TSVs are holes going through the chip silicon substrate filled with a conducting material. They enable short interconnections in 3D-SICs. Stacking dies using vertical interconnects have many benefits [4], including:

- Low latency interconnects between adjacent dies.
- Reduced power consumption.
- High bandwidth communication as TSVs cross dies along the surface of the chip
- Improved form factor and package volume density.
- Heterogeneous integration. Different dies in the stack could be manufactured by different wafer fabs, but also using different technologies. DRAM and logic integration in a single 3D-SIC becomes feasible.

Each manufactured 3D-SIC has to be tested to guarantee the required quality and defect-per-million (DPM) level. Several prior work addressed these issue and present test approaches for 3D-SICs [5–8]. For example, Lewis and Lee [9] considered pre-bond die testing in order to obtain a satisfactory compound yield. The authors proposed a scan island approach based on the IEEE 1149.1 [10] and IEEE 1500 [11]. Marinissen et al. [12] addressed many limitations of previous work by proposing a structured and scalable test access architecture using *TestTurns* and *TestElevators* to route test data through the stack, for pre-, mid- and post-bond tests. The architecture is further extended to support Multiple Tower (MT) stacking [13] and $2\frac{1}{2}$-D stacking [14,15]. Many of these features are ongoing activities in the IEEE P1838 [16,17] working group. JEDEC announced a new standard for Wide I/O mobile DRAM (JESD229 [18]). This standard supports interconnect testing through JTAG.

The state-of-the art in testing 3D stacked ICs assumes mainly the presence of scan chains and JTAG on each die, which are also used to perform interconnect test. However, stacked dies may not always contain JTAG interfaces. For instance, it is well know that memory providers are not in favor of integrating JTAG in their designs; they prefer rather to use a memory BIST (MBIST). Therefore, assuming that each stacked dies include JTAG is too optimistic. In this paper we will explore different ways of testing interconnects of stacked memory on logic both in the presence and in the absence of a JTAG interface. We also discuss the implications of having MBIST location (either on the memory die or on a logic day) on the different interconnect test approaches.

The remainder of this paper is organized as follows. Section II presents the requirements related to testing the interconnects of stacked memories. Section III presents an overview of existing 2D test standard that could be extended to 3D; it also briefly presents (on-going) 3D test standards. Section IV classifies possible 3D stacked memories. Section V explores the different interconnect BIST (iBIST) schemes for these memory classes, each scheme with its pros and cons. Finally, Section VI concludes this paper.

II. Interconnect Test Requirements

Test standards need to satisfy certain requirements. For an interconnect BIST (iBIST), requirements can be classified and belong to the memory interface, test quality, compatibility with previous standards and to test modularity. Each of them is briefly described next.

A. Support for Different Memory Interfaces

A memory interface consists of a set of uni- or bidirectional wires possibly off-chip that describe the interaction with the memory. In 3D-SICs off-chip connections are mapped in the vertical dimension on TSVs. Typical interface signals include control, address and data signals as depicted in Figure 1. A memory consists of at least 1 access port, but in general could contain multiple read and write ports. The memory dies can operate either synchronously or

978-1-4799-3525-3/13/$31.00 ©2013 IEEE

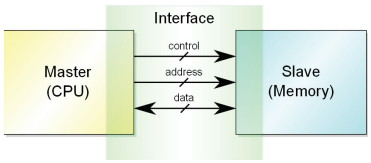


Fig. 1. Memory interface.

asynchronously and are implemented using any technology such as SRAM, DRAM, Flash etc.

As we are dealing with stacked memory on logic, the interface as shown in Figure 1 has to be realized using TSVs. Therefore, failures in the interface or interconnects can be assumed to be independent of the memory technology under consideration. Hence, the iBIST solution has to deal with the interface irrespective of the memory type (e.g. SRAM, DRAM, Flash)

B. Test Quality

Design for testability and diagnosis is an important step in the design phase. Each wire/TSV that connects the master (e.g., CPU) with the slave (stacked memory) should be tested. Although TSVs are relative huge wires as compared to on-chip wires, many defects can occur; examples are as unfilled TSVs, partial filled TSVs, opens, roughness/spikes in TSV sidewall layers, manufacturing flaws in sidewall isolation oxide layer, leakage, etc. Any test solution should target as much as possible of such defects. Test patterns for some of such defects are well known [19]. However, as TSV are new components in the stack, new fault models might become relevant since a fully understanding of all TSV failure mechanisms is still needed; TSV keep-out-zone [20] and coupling [21] are examples of that.

C. Compatibility

Any suitable iBIST will add additional DfT hardware on the dies. However, the solution has to be compatible with the existing standards such as JTAG. Ideally the solution should form an extension of an existing/ongoing standards (such as the IEEE P1838 [16]) or easy to be integrated in them.

D. Modularity

The iBIST is responsible for interconnect testing only; e.g., it can be reused for any other kind of memories stacked on logic. Therefore the solution has to be modular. The concept of modularity provides many advantages such as (a) it helps in saving the development time and cost, (b) testing interconnect separately from the other dies, (c) allows memory providers to protect their IPs and withheld the implementation details even if the solution is integrated within the memory die, etc.

III. 3D Test Architectures

This section consists of two parts; first, it describes some of the familiar 2D test standards and subsequently, the (on-going) 3D standards. Here, we primarily focus on the interconnect test part as it is the main purpose of this paper.

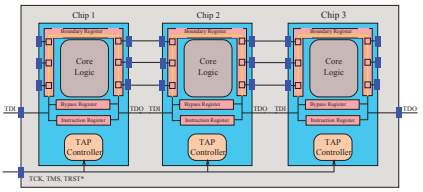


Fig. 2. IEEE std. 1149.1 wrapper.

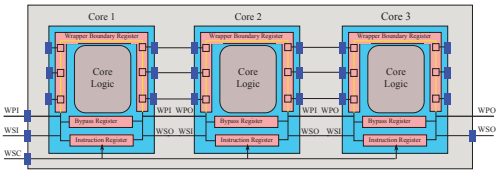


Fig. 3. IEEE std. 1500 wrapper.

A. 2D standards

The existing 2D test standards can be classified into three categories:

- Boundary Scan (BS) based: in this category input and output pins are wrapped by boundary cells. Testing input and output pins goes through this hardware.
- Test Logic (TL) based: in this configuration pins are tested by specific dedicated logic that generates output sequences based on the inputs.
- Instrumentation based: in this configuration, test units are activated by means of test instruments.

Boundary Scan based

Several standards are based on the boundary cell concept. In this section, we consider the two most important ones, i.e., JTAG [10] and IEEE Std. 1500 [11].

JTAG (also known as IEEE Std. 1149.1) is primarily developed for interconnect tests (EXTEST) on a Printed Circuit Board (PCB), but it can also be used to test independent dies on the board (e.g. diagnosis mode). JTAG comes with a low cost wrapper around each pin of each chip and is controlled by the TAP controller as depicted in Figure 2. The figure shows an example in which three chips are placed on a PCB. The Test Data Input (TDI) and Test Data Output (TDO) of each chip are cascaded and form a sequential chain. The operation mode of each chip is controlled through the TAP controller.

As System on Chips (SOCs) get more sophisticated and more IP-cores are integrated, test time becomes more critical. This necessitates a standard (IEEE Std. 1500 [11]) that supports cost-efficient testing of core-based SoCs. A similar wrapper as for the IEEE std. 1149.1 is placed around the core, with mainly the following differences: (a) the newer standard supports a wider parallel test data interface denoted by WPI, and (b) the WIR register is controlled directly at the cost of some extra I/O pins.

Several other (ongoing) standards that are based on a wrapper cell similar as in JTAG can be found in literature such as IEEE P1149.7. We refer to all of these schemes as Boundary Scan based testing.

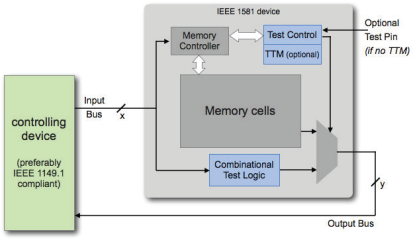


Fig. 4. IEEE std. 1581 [22].

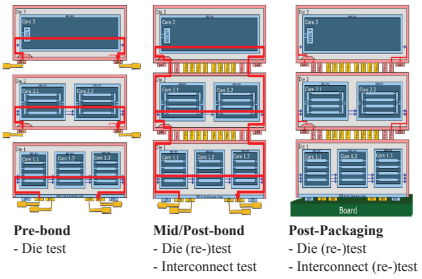


Fig. 5. IEEE P1838 [16].

Test Logic Based

A complete different way to test interconnects is by inserting dedicated logic for it. Figure 4 shows this concepts for a memory slave for the IEEE Std. 1538 [22]. In normal mode, the interconnects between the memory and host (e.g., a CPU) are in transparent mode and memory operations are not affected by the additional test hardware. However, the memory is bypassed in test mode and the inputs of the memory are directly forwarded to its outputs through combinational test logic. The combinational test logic usually consists of a couple of XOR gates. In the IEEE Std. 1538 the test mode is either activated by a dedicated pin or by the Transparent Test Mode (TTM) [22], where the TTM activates the test by special input sequences. For example, a specific clock frequency on the clock input pin of the memory, or a fixed input pattern that normally is considered to be an invalid can activate the test mode. The advantage of this scheme over BS based testing is a much more efficient test methodology for complex memories such as Flash EEPROM.

Instrumentation based

In instrumentation based testing, test resources on the chip are accessed using instruments, where each instrument could be any DfT unit such as a logic BIST, an MBIST, an analog BIST, etc. The instruments target only fractions of the chip. An ongoing standard is the IEEE P1687 [23]. By incorporating an instrument for TSVs, interconnects between dies can be tested. We do not consider this option in the remainder of this paper as it is currently not standardized.

B. 3D Standards in development

As 3D-SICs are quickly gaining more ground the need for a standardized test becomes more important. Several DFT solutions have been proposed [5–9], but with many limitations such as being not able to perform a test on a partial stacked die. Nevertheless, two promising standards are IEEE P1838 [16,17] and JEDEC 229 [18].

The IEEE P1838 is an on-going standard for 3D-SICs and focuses on dies as key components in the stack. The stack-level architecture routes both data and control signals up and down through the stack (TestTurns and TestElevators) to reach each particular die in the stack. The architecture supports both intra-die test (INTEST) and inter-die test (EXTEST) during all test phases as depicted in Figure 5. In the pre-bond phase, dedicated pads are used to test dies. In the mid-bond and post-bond phases, both dies in partial and a complete stack respectively can be tested (INTEST). EXTESTs can be performed for the interconnects during mid-bond and post-bond and are based on the JTAG [10] and IEEE1500 [11] standards. The final test (post-packaging) consists of the same tests options.

Recently, JEDEC announced a new standard for wide I/O mobile DRAM [18]. This standard is more than a test specification and targets the whole memory; the target is to improve memory bandwidth and to reduce latency, power, and form factor. The wide I/O specification defines the interface between logic and memory (LMI). With respect to testing, the following two functionalities are provided by the standard. The first added DfT hardware is JTAG used to test for contacts (TSVs and microbumps) and I/O testing. The second test feature is a post-assembly DRAM test to ensure the quality of memory dies. This makes it possible to test the DRAM independently from the logic chip. The DRAM layers are tested either through direct access pins or by electrical connection through General Purpose Input/Output (GPIO) drivers/receivers.

IV. 3D Stacked Memory Classification

In this section, we focus on the die test of the 3D memory. Section IV-A first presents the possible test moments. Thereafter, Section IV-B classifies 3D stacked memory.

A. Testing 3D ICs

This section presents first the differences between 2D and 3D test flows and shows that for 3D ICs many test moments are possible. These test moments are thereafter compiled into a framework of test flows.

A conventional 2D test flow for planar wafers is depicted in Figure 6(a) [24]. Here, usually two *test moments* are applicable; i.e., a wafer test prior to packaging and a final test after packaging. The wafer test can be cost-effective when the yield is low, since it prevents unnecessary assembly and packaging costs. The goal of the final test is to guarantee the final quality of the packaged chip. During the manufacturing of a 3D-SIC, additional test points can be defined for each partial created stack. At each test point a distinction can be made between die tests and interconnect tests. Die tests

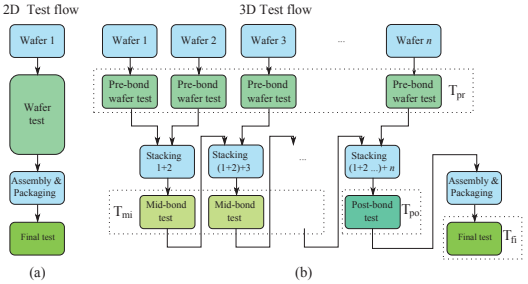


Fig. 6. 2D versus 3D D2W test flows.

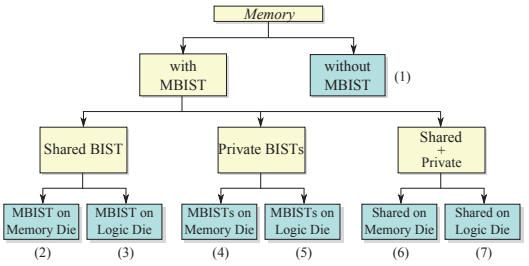


Fig. 7. 3D Stacked memory classification.

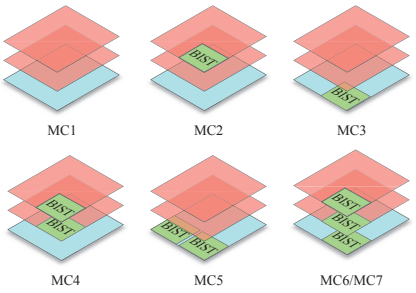


Fig. 8. Memory configurations.

ensure the functionality of individual dies, while interconnect tests ensure functional TSVs between dies. For 3D-SICs, four test moments can be distinguished in time as depicted in Figure 6(b), and explained next.

1) T_{pr}: n *pre-bond* wafer tests, since there are n layers to be stacked. T_{pr} tests prevent faulty dies entering the stack. Besides die test, preliminary TSV interconnect tests can be applied. For example, in [25] the authors use a capacitance test to detect some of the faulty TSVs. In [26], the authors propose active probing to detect faulty preliminary TSVs.
2) T_{mi}: n-2 *mid-bond* tests applicable for partial created stacks. In this case, either the dies, the interconnects, their combination or none of them can be tested. Good tested dies in the pre-bond test phase could get corrupted during the stacking process as a consequence of e.g., die thinning, and bonding [27].
3) T_{po}: one *post-bond* test. This test can be applied after the complete stack is formed. Analogous to wafer testing in the 2D test flow, T_{pr} can be applied to save unnecessary assembly and packaging costs. Both interconnects and dies can be tested.
4) T_{fi}: one *final* test can be applied after assembly and packaging to ensure the required quality of the complete 3D-SIC. Other specific packaging related tests could be applied at this test moment as well.

Note that in total there are $2 \cdot n$ different test moments. Depending on whether one or more companies are involved in the manufacturing of 3D-SICS, different requirements can be set for the pre-bond wafer test quality [28]. If the (pre-bond) wafers are produced by one or more companies and the final 3D-SIC product is processed and manufactured by another company, a high pre-bond wafer test quality (e.g. a KGD) often is agreed upon. If a KGD contract is in place, high-quality pre-bond testing is required. If such a contract is not in place, the pre-bond test quality is subject to optimization. This means that there is not only the option to perform pre-bond testing or not, but also to perform pre-bond testing at a higher or lower test quality. Faulty undetected dies can be detected in a later stadium, e.g., in higher quality post-packaging tests. Similarly, a high quality pre-packaging test (Known-Good-Stacks test) can be applied.

A pre-bond memory die could be tested with a MBIST engine. That same engine could be reused for mid-, post-bond and final tests. For the iBIST, pre-bond testing is not considered as interconnects are only formed after stacking.

B. 3D Stacked Memory Classification

Memories are typically tested by MBISTs which perform high quality at-speed tests. In 3D memories, one or several of MBISTs might exist in the stack simultaneously and there number is a trade-off between area, test time, yield, etc. For example, MBIST and/or Built-in-Self-Repair (BISR) circuits in the pre-bond phase allow dies to be tested and repaired prior to stacking, while a shared MBIST/BISR unit in the mid-bond or post-bond phase can also be used for vertical repair, i.e., inter-die redundancy. Each memory configuration affects the test and repair strategy. We define 7 cases of 3D stacked memories as depicted in Figure 7 based on the availability of MBIST engines, whether they are shared or private, and their location (on the memory or logic die) in the stack. They are further explained next and an example is given for each memory configuration (MC) in Figure 8. The examples consist of two top memory dies and a single bottom logic die. Note that each configuration could have 0, 1 or more BIST engines based on the configuration.

1) MC1 contains no MBISTs engines. Therefore, pre-bond testing (if performed) is done by probing the dies likely with lower quality and/or higher test cost as compared to using an MBIST). In the final phase, testing of memory can be performed through the logic layer, e.g., a CPU test [29]. This configuration is not interesting as it might be difficult to guarantee test quality and to perform diagnosis.
2) In MC2 a shared MBIST engine is placed on one of the memory dies. The MBIST can be used for

pre-bond testing and for post-bond testing when the whole memory is created. A clear visible drawback of this scheme is non-identical memory layers (with DfT and without DfT). The memory layer without an MBIST faces a similar pre-bond test problem as in MC1. An additional drawback are extra vertical TSV connections that are required to access and program the MBIST. Benefits of this system include area efficiency (a single BIST only), and close to at speed testing (latency to CPU is not taken into account). Moreover, an optimal test algorithm can be programmed as the memory manufacturer is responsible for the MBIST content. Theoretically speaking, if the repair rate is high enough the pre-bond tests can be skipped as the memory can be repaired in a later phase.

3) In MC3 the shared MBIST is placed on the logic die. Note that this logic die could be an interposer (used for the peripheral circuits) or be residing an actual design such as a CPU. This configuration has the benefit that at speed testing can be performed. However, as the memory dies could be manufactured in a different company then the CPU die, the system integrator is responsible for the memory test algorithms (which could be non-optimal due to confidentiality). Inter-die repair is a still possible, but the mutual sharing of resources on the memory dies becomes harder to implement. More interesting for this configuration is to use global spare resources on the logic die to replace faulty cells in the memory. Note that defects also occur due to stacking. Similarly as in MC1, pre-bond testing can only be done by probing. This configuration is efficient in terms of area (a single MBIST only) and cost less to access as compared to MC2.
4) MC4 is in essence an extension of MC2 in which each layer has its own private MBIST. If we compare this configuration with MC2, we see an extra cost in terms of area, but at each die can be tested at pre-bond using its private MBIST. Other benefits of this scheme are independent testing of layers in parallel (faster in test time) and inter-die repair can be realized similarly as in MC2.
5) MC5 can be seen as an extension of MC3, where each layer has its own MBIST on the logic die. Benefits of MC5 include test time reduction if both MBISTs run in parallel each optimized for its own memory layer and expense of extra area. The remainder benefits of this configuration are similar as MC3 such as global memory repair.
6) The theoretical difference between memory configurations MC6 and MC7 is the location of the shared MBIST in the stack (in the logic die for MC7 and in one of the memory dies in configuration MC6). We only consider the case where this shared MBIST is placed on the logic layer (i.e., MC 7). MC7 is basically now an extension of MC3 and MC4 and has therefore both benefits of these configurations (i.e., at speed testing in the pre-, mid- and post-bond and final tests all with repair capabilities). Drawback, however, is the additional area overhead. This memory configuration might exist for the case where the dies in the stack are composed of different technologies (e.g., FLASH, DRAM etc. all with their private MBISTs) and where a shared MBIST on the logic die is used for a global memory test.

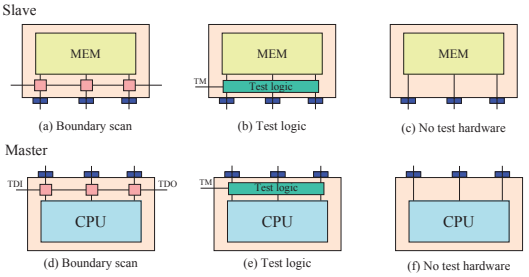


Fig. 9. iBIST schemes.

V. 3D iBIST Schemes

The second part that requires testing in 3D-SICs are the vertical interconnects. Figure 9 shows the hardware required for interconnect testing using the BS and *test logic*. The figure shows the required test infrastructure for both the master (logic) and slave (memory) dies such that interconnects become testable. On the slave side the test hardware for the interconnects consists either of (a) a boundary scan or (b) test logic or (c) no dedicated DFT. Similarly, cases (d), (e) and (f) show the same options for the master which in this case is considered to be a CPU. In total, nine combination can be formed. We discuss in short the applicability of each of these schemes which are depicted in Figure 9 from the master's perspective.

a) Master side - Boundary Scan: Figure 9(d) shows the case where the interconnects on the master die are tested with BS. In theory, the master could be connected with all the 3 test options for the slave. In case both the master and slave are connected using BS, it requires proper mode selection of the dies such that the return paths of the BS chain are matched [12]. In case, the interconnects on the memory side are connected to test logic, the BS has first to be put in the proper test mode. After that the test logic should be activated and subsequently the responses of the test logic based on the shifted patterns must be captured. The last case, where the memory has no test seems impractical for test purposes. Nevertheless, if this option is selected test patterns have to be applied in such away that the interconnects are tested through the memory. This might have a severe impact on test time for the interconnects.

b) Master side - Test logic: Figure 9(e) shows the case where the interconnects on the master die are connected using test logic. Note that the test logic on the Master side is different from the test logic on the slaves. On the master, the test logic is responsible for test pattern creation, while the test logic on the slave side only responds by sending patterns back based on its inputs. When the master contains dedicated test logic for the interconnects, it would be most

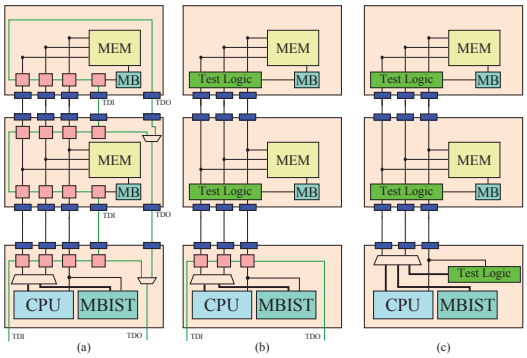


Fig. 10. Interconnect DFT for MC 7.

likely to have dedicated test logic on the slave side as well. These patterns can be stored on the master die (as only few patterns are required for interconnect tests [19]), or can be shifted in using normal scan chain if available.

c) Master side - No Test: In case there is no additional hardware support for interconnect testing (Figure 9(f)) on the master die it would be hard to communicate with any DfT on the Slave. Therefore, no direct test support for interconnects seems the only applicable case; nevertheless attempts could be made to force values on the TSVs indirectly by using internal scan chains if available. This approach without any DFT for the interconnects seems risky. In the final test, however, interconnects can still be tested indirectly for example through a CPU test, but requires more research.

From the schemes discussed above, we select the tree most promising iBIST configurations and combine their DfT with the MBISTS of configuration MC7 (most extensive configuration of Figure 8); the three iBIST schemes are depicted in Figure 10. Figure 10(a) shows this for the first case where both the master and two slaves contain a boundary scan. Note that the return path of the BS is multiplexed in order to differentiate between pre-bond and post-bond tests. Part (b) of the figure shows the second case where the slave contains test logic. As there are multiple slaves with test logic, each slave has to have its own activation circuit. This can be obtained for example by having different activation frequencies. Finally, part (c) depicts the hardware for the third case in which both the master and slave contain test logic for the interconnect test. Future research should determine which approach performs best in terms of test time and area overhead for given memory configurations.

VI. Conclusion and Future Work

In this paper, we described challenges and test opportunities for 3D stacked memories consisting of die tests and interconnect tests. First, we presented the possible test moments for a 3D-SIC. Thereafter, we explored for die tests the impact of the MBIST location and discussed how they affect quality and memory repair. For interconnect tests, several test approaches (or iBISTs) were explored. These explorations are required to develop low cost standardized test methodologies. In the future we will design and implement the iBIST schemes to obtain accurate trade-offs in term of hardware overhead and latency.

Acknowledgements

This research is supported by Catrene through the 3DIM^3 project (grant CT105).

References

[1] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", *Proceedings of the IEEE*, 2006.
[2] G. Loh et al. "Processor design in 3D die-stacking technologies", *IEEE Micro*, pp. 31-48, 2007.
[3] W. R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Design Test on Computers*, pp. 498-510, 2005.
[4] P. Garrou, Christopher Bower and Pater Ramm, "Handbook of 3D Integration", Wiley-VCH, 2008.
[5] X. Wu, P. Falkenstern, and Y. Xie. "Scan Chain Design for Three-dimensional Integrated Circuits (3D ICs)", *Intl. Conf. on Computer Design*, pp. 208-214, 2007.
[6] Xiaoxia Wu et al. "Test-Access Mechanism Optimization for Core-Based Three-Dimensional SOCs", *Intl. Conf. on Computer Design*, pp. 212-218, 2008.
[7] L. Jiang, L. Huang, and Q. Xu, "Test Architecture Design and Optimization for Three-Dimensional SoCs", *Design, Automation, and Test in Europe*, pp. 220-225, 2009.
[8] L. Jiang et al. "Layout-Driven Test-Architecture Design and Optimization for 3D SoCs under Pre-Bond Test-Pin-Count Constraint", *Intl. Conf. on Computer-Aided Design*, pp. 191-196, 2009.
[9] Dean L. Lewis and Hsien-Hsin S. Lee. "A Scan-Island Based Design Enabling Prebond Testability in Die-Stacked Microprocessors", *International Test Conference*, pp. 1-8, 2007.
[10] "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE Std. 1149.1-2001, 2001.
[11] IEEE Std. 1500, "IEEE Standard Testability Method for Embedded Core-Based Integrated Circuits", 2005.
[12] E.J. Marinissen et al. "A structured and scalable test access architecture for TSV-based 3D stacked ICs", *VLSI Test Symp.*, pp. 269-274, 2010.
[13] C-C Chi et al., "Post-bond testing of 2.5D-SICs and 3D-SICs containing a passive silicon interposer base", *ITC*, pp. 1-10, 2011.
[14] C-C Chi et al., "DfT Architecture for 3D-SICs with Multiple Towers", *European Test Symposium*, pp. 51-56, 2011.
[15] K. Saban, "Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency", *http://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf*, 2010.
[16] "IEEE P1838", *http://grouper.ieee.org/groups/3Dtest/*, 2012
[17] "P1838 - Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits", *http://standards.ieee.org/develop/project/1838.html*, 2012
[18] "JEDEC, Wide I/O Single Data Rate (Wide I/O SDR) JESD229", *http://www.jedec.org/standards-documents/results/jesd229*, 2011.
[19] P. Goel, AND M.T. McMahon, "Electronic Chip-In-Place Test", *I19th Conference on Design Automation*, pp. 482-488, 1982.
[20] S.-K. Ryu, "Effect of Thermal Stresses on Carrier Mobility and Keep-Out Zone Around Through-Silicon Vias for 3-D Integration", *IEEE Transactions on Device and Materials Reliability*, pp. 255-262, 2012.
[21] C. Liu, "Full-Chip TSV-to-TSV Coupling Analysis and Optimization in 3D IC", *Design Automation Conference*, pp. 783-788, 2011.
[22] H. Ehrenberg et al., "IEEE Std 1581 A standardized test access methodology for memory devices", *International Test Conference*, pp. 1-9, 2011.
[23] IEEE P1687, "http://grouper.ieee.org/groups/1687/", 2005.
[24] E. J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference*, pp.1-11, 2009.
[25] P. Chen, et al. "On-Chip TSV testing for 3D IC before bonding using sense amplification", *Asian Test Symposium*, pp. 450-455, 2009.
[26] B. Noia, K. Chakrabarty, "Pre-Bond Probing of TSVs in 3D Stacked ICs", *IEEE International Test Conference*, pp. 1-10, 2011.
[27] H-H. S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computers*, pp. 26-35, Oct. 2009.
[28] E. J. Marinissen, "Testing TSV-Based Three-Dimensional Stacked ICs", *Design, Automation and Test in Europe*, pp. 1689-1694, 2010.
[29] A. Goor et al., "Memory Testing with a RISC Microcontroller", *Design, Automation & Test in Europe Conference, 2010*, pp. 1-9, 2011.

Interconnect Test for 3D Stacked Memory-on-Logic

Mottaqiallah Taouil, Mahmoud Masadeh, Said Hamdioui

Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands
{m.taouil, s.hamdioui}@tudelft.nl

Erik Jan Marinissen

IMEC vzw
Kapeldreef 75, 3001 Leuven, Belgium
erik.jan.marinissen@imec.be

***Abstract*—Three-dimensional stacked IC (3D-SIC) technology based on Through-Silicon Vias (TSVs) provides numerous advantages as compared to traditional 2D-ICs. A potential application is memory stacked on logic, providing enhanced throughput, and reduced latency and power consumption. However, testing the TSV interconnects between the two dies is challenging, as both the memory and the logic die might come from different manufacturers. Currently, no standard exists and the proposed solutions fail to address dynamic and time-critical faults (at speed testing). In addition, memory vendors have not been in favor to put additional DfT structures such as JTAG for interconnect testing on their memory devices. This paper proposes a new Memory Based Interconnect Test (MBIT) approach for 3D stacked memories. Our test patterns are applied by read and write instructions to the memory and are validated by a case study where a 3D memory is assumed to be stacked on a MIPS64 processor. The main benefits of the MBIT approach are: (1) zero area overhead, (2) the ability to detect both static and dynamic faults and perform at speed testing, (3) flexibility in applying any test pattern, as this can be executed by the CPU on the logic die and (4) extreme short test execution time.**

Keywords: *interconnect testing, 3D-SIC, memory-on-logic*

I. Introduction

The popularity of 3D Stacked ICs (3D-SICs) is rising among industry and research groups [1]. 3D-SICs based on Through Silicon Vias (TSVs) are emerging as one of the main competitors to continue the trend of Moore's Law [2]. Stacking dies with vertical interconnects possess many benefits [1], such as (a) low latency between adjacent dies, (b) reduced power consumption, (c) high bandwidth communication, (d) improved form factor and package volume density, and (e) heterogeneous integration.

One of the main applications that utilizes the mentioned benefits is the stacking of memory (DRAM) on logic (CPU). After stacking, a post-bond interconnect test is required to test interconnects (TSVs + μ-bumps) between the memory and logic dies. This is not straightforward as (1) stacked dies may come from different providers (IP confidentiality), (2) memory providers are reluctant to integrate DfT such as JTAG for interconnect testing, and (3) even with DfT support, obtaining high coverage for dynamic faults is still challenging.

Currently, no standard exists to test interconnects in memories stacked on logic. However, some test approaches are being under development. IEEE P1838 [3] is currently an ongoing standard that develops DfT for general stacked ICs; it is based on the presence of Boundary Scan (BS) cells in all dies. Wide I/O [4] also supports interconnect testing using BS. However, (DRAM) memory vendors are not always in favor of integrating JTAG on their devices [5]. Other approaches such as the IEEE P1581 [5], originally for 2D ICs, can be extended in the third dimension. In test mode, the memory is bypassed and interconnects are tested by creating a direct logic function between the inputs and outputs of the memory. IEEE P1581 prefers a JTAG compliant logic chip, i.e., the test logic on the memory chip can function with a logic chip that supports JTAG. This standard can be mapped to 3D-SICs by having the bottom die (logic) JTAG compliant and where the test logic has to reside on the top die (memory). This approach, referred to as Test Logic (TL) based interconnect testing, also requires additional DfT test logic on the memory die. In addition to the undesired DfT on the memory die, both the BS and TL based test methods are unable to provide at speed testing required to target dynamic faults. Testing for dynamic faults is crucial, as 3D interconnects are expected to suffer from speed and timing related faults [6–11].

In [12] and [13] authors present hardwired BISTs with at-speed testing capability for crosstalk faults. Both methods are not flexible in altering test patterns and require additional DfT area. In [13] its reported that the area overhead of the method in [12] approximates 9.8% while their own equals 7%. They evaluate this in 90 nm technology using 15 μm TSV diameters.

This paper proposes a post-bond Memory Based Interconnect Test (MBIT) methodology being able to test interconnects between memory and logic dies by performing read and write operations from the logic die (CPU) to the memory dies. A similar approach is taken in [14], but it is inapplicable for TSV arrays. This paper also provides a classification of interconnect defects, and compiles them into fault models. In addition, it discusses the test pattern generation for these faults and uses the proposed MBIST to implement them. MBIT does not require any DfT area as it reuses existing components in the stack. It supports at-speed testing and detects static and dynamic faults. Moreover, it is very flexible in altering test patterns simply by modifying software instructions and has a extreme short test execution time.

The remainder of the paper is organized as follows. Section II presents defects, fault models and detection conditions for 3D interconnects. Section III describes thereafter the test pattern generation for the targeted fault models. Section IV provides the simulation results and compares our methodology with the state-of-the-art. Finally, Section V concludes this paper.

978-3-9815370-2-4/DATE14/©2014 EDAA

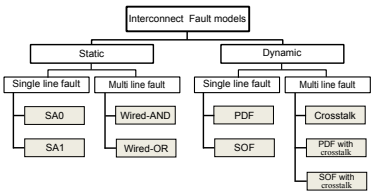


Fig. 1. Fault Model Classification for interconnects

II. Defect, Fault Models and Detection

A. Defects in Interconnects

Interconnects in 3D-stacking are a potential source of defects inherent to the manufacturing steps such as TSV fabrication/filling, bonding etc. Defects transpire both in TSVs and micro-bumps and examples of defects are given next.

Defects related to Through-Silicon-Via (TSV):

- D1 Pinhole defects occur along TSV walls and cause a short (low resistance path) between TSVs and the substrate; This may cause degradation of the signal quality in terms of strength and speed [6,7,9,15].
- D2 An incomplete fill of TSVs (voids) may originate from insufficient wetting during plating. Voids cause partial opens and lead to higher TSV resistance [6,7,9,15].
- D3 Coefficient of thermal expansion (CTE) mismatch between TSV metal (most likely copper) and substrate may lead to TSV cracks and sidewall delamination. Both lead to increased path resistance [9,15–18].
- D4 Pinch-off of TSVs during plating could lead to increased TSV resistance or partial opens [7].
- D5 Missing contacts between TSVs and the transistors or metal layer cause opens [7,8].
- D6 TSV misalignment with μ-bumps increases the resistance and causes (partial) opens [7,9,15].
- D7 Crosstalk between different TSVs [9,10].

Defects related to μ-bumps:

- D8 Damage in underlying BEOL [19].
- D9 Weak bonding due to buckled thinned Si chip [19].
- D10 Variation in TSV heights may cause tin to be squeezed out from μ-bump causing shorts between μ-bumps [19,20].
- D11 Electromigration may cause voids and cracks in the joints, resulting in higher resistive μ-bumps, or opens [21].
- D12 μ-bump cracks due to CTE mismatch between copper, silicon, and silicon-oxide [7].

B. Faults and Fault Models

Interconnect fault models can be classified into static and dynamic faults. Fig. 1 shows a classification of the faults. A defect can cause a single line or a multi line fault. Each fault is depicted in Figure 2 and explained next. Static faults include:

- Stuck-at-Fault (SAF). There are two types of SAF faults: stuck-at-0 (SA0) and stuck-at-1 (SA1) as depicted in Fig. 2(b). A SAF fault can be caused by defect D1.
- Bridge fault. *Simple* bridge faults include wired-AND (Fig. 2(c)) and wired-OR (Fig. 2(d)) faults. *Complex* bridge faults also exists, such as A dominate-AND B in which wire A is fault-free and where B takes the value A $\cap$ B. A bridge fault can be caused by defect D10.

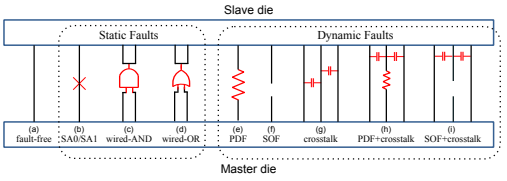


Fig. 2. Static and dynamic faults

Dynamic faults include:

- Path Delay Fault (PDF): A partial open line defect increases the line delay (Fig. 2(e)). It can affect both rising or falling delay time. PDF faults can be caused by defects D2, D3, D6, D8, D11, D12.
- Stuck Open Fault (SOF): This is caused by a complete open line defect (Fig. 2(f)). SOFs can be caused by defects D5, D6, D8, D9, D11, D12.
- Crosstalk Fault: Faults on victim lines are caused by crosstalk from aggressive neighbors (Fig. 2(g)). Several crosstalk faults exists as described by the Maximum Aggressor (MA) fault model [22] such as (1) glitch-up, (2) glitch-down, (3) falling delay, and (4) rising delay. Each fault has a specific behavior, while it represents the same phenomena. Defect D7 may cause crosstalk faults.
- PDF with Crosstalk: Faults due to partial resistive opens (victims) are affected by crosstalk from neighbors (Fig. 2(h)). PDF with Crosstalk faults can be caused by combinations of crosstalk and PDF faults.
- SOF with Crosstalk: Faults due to complete open lines (victims) are affected by crosstalk from neighbors (Fig. 2(i)). SOF with Crosstalk faults can be caused by combinations of crosstalk and SOF faults.

The dynamic faults embody most of the defects and therefore it is essential to test for them.

C. Detection Conditions

The detection conditions of each fault are described next. In general, the detection process adheres to the following steps:

1) Fault sensitization (activation): create a different behavior between the faulty and fault-free circuit.
2) Fault propagation: make the fault visible at the outputs.
3) Line justification: backtrack the values to the input of the circuit, such that the inputs sensitize the fault.

Fault propagation and line justification for address and data lines are dissimilar. Data lines can be controlled and observed directly through writing and reading. Therefore, fault propagation and line justification are straightforward. However, address lines are uni-directional and fault propagation must be performed indirectly by utilizing data lines (e.g., by writing and reading different values to different addresses). Control lines, such as write or read signals, are tested implicitly. For fault sensitization, special sequences and/or transitions are required for each fault.

TABLE I
SAF TEST PATTERNS FOR DATA LINES

OP	Op.	Address	Data
OP1	*W*	$Addr_x$	F F F F
OP2	*R*	$Addr_x$	F F F F

OP	Op.	Address	Data
OP1	*W*	$Addr_y$	0 0 0 0
OP2	*R*	$Addr_y$	0 0 0 0

SAF (SA0/SA1): A stuck-at-fault forces a wire to a specific value; either 0 (SA0) or 1 (SA1). Therefore, to sensitize a SAF fault an opposite value must be applied to the wire.

Bridge fault (Wired-AND/Wired-OR): To sensitize a bridge fault, two opposite values must be specified on each pair of lines. Simple bridge faults such as the ones depicted in Fig. 2(c) and (d) require at least one of the two patterns 0-1 or 1-0 as inputs. More complex bridges such as A dominate-AND B require both 0-1 and 1-0 inputs on each pair of wires for fault sensitization.

PDF: We assume that the path delay fault consists of a low to moderate resistance value, violating the normal operation with at most one additional clock cycle. Faults that lead to larger delays, i.e., more than one extra cycle, can be considered as SOFs. To sensitize PDF faults, both 0→1 and 1→0 transitions should be applied on each line.

SOF: For SOFs we assume that during short time intervals the non-driven part of the floats remain stable. Therefore, to sensitize SOF either a 0→1 or 1→0 transition is needed.

Crosstalk: We consider only the most relevant crosstalk faults. To sensitize a rising crosstalk fault, a victim must undergo a 0→1 transition, while the aggressors simultaneously make a 1→0 transition. The reverse applies for falling delay faults.

PDF with Crosstalk: The fault sensitization for PDF faults with crosstalk is the same as falling and rising delay faults, as this maximizes the applied stress from the aggressors.

SOF with Crosstalk: The fault sensitization for this fault requires both a 0→1 transition on the victim while keeping the aggressors stable at 0 and a 1→0 transition on the victim while keeping the aggressors stable at 1. Keeping the aggressors stable reduces the coupling with the floating part of the SOF, hence it minimizes the contribution of the aggressors to the transition on the floating part of the victim.

III. TEST PATTERN GENERATION

Due to space limitation, we discuss only a subset of fault models. We restrict ourselves to static faults and one dynamic fault (SOF with Crosstalk). Nevertheless, the experiment results in Section IV will be presented for all faults. Next, a single fault is assumed to occur at a time. In addition, during explanation we assume L_d=16 bit data lines (presented in *hexadecimal* value) and L_a=16 bit address lines (presented in *binary* value).

A. Static Faults

In this section we will present the test patterns of static faults. A SAF fault may happen in data lines or address lines. A bridge fault may happen: (1) between data lines, (2) between address lines, and (3) between data and address lines.

SAF at data lines: Table I shows the memory operations required to detect SA0 (left table) and SA1 faults (right table) on data lines. The tables consist of four columns; the first column shows the index of the operation; the second column the type of operation, i.e., read (R) or write (W); the third and fourth columns show the address and data values, respectively. Both tables contain a write followed by a read operation. SAF faults will be detected during read OP2.

TABLE II
SAF TEST PATTERNS FOR ADDRESS LINES

OP	Op.	Address	Data
OP1	*W*	0000 0000 0000 0000	Init_Data
OP2	*W*	0000 0000 0000 000**1**	$Data_1$
OP3	*W*	0000 0000 0000 00**1**0	$Data_2$
...	...	...	...
OP16	*W*	0**1**00 0000 0000 0000	$Data_{15}$
OP17	*W*	**1**000 0000 0000 0000	$Data_{16}$
OP18	*R*	0000 0000 0000 0000	Init_Data

OP	Op.	Address	Data
OP1	*W*	1111 1111 1111 1111	Init_Data
OP2	*W*	1111 1111 1111 111**0**	$Data_1$
OP3	*W*	1111 1111 1111 11**0**1	$Data_2$
...	...	...	...
OP16	*W*	1**0**11 1111 1111 1111	$Data_{15}$
OP17	*W*	**0**111 1111 1111 1111	$Data_{16}$
OP18	*R*	1111 1111 1111 1111	Init_Data

SAF at address lines: Table II shows the test patterns to detect both SA0 (left table) and SA1 faults (right table) in address lines. Detecting SA0/SA1 faults at address lines is more complex as they affect the memory address. For example, writing a value to the address all 1's and subsequently reading from this address will not detect any SA0 fault in the address lines; this is because both the write and read operation are affected in the same way and the fault is not sensitized. To test memory address lines, each address line should be tested separately. For example, by using a walking-1 sequence for SA0 as depicted in the left table. Address 0 of the memory is first initialized to *Init_Data* during OP1. During write operation OP2 to OP17 (with different data than *Init_Data*), any SA0 in the address lines will overwrite *Init_Data* of address 0. Therefore, read operation OP18 is able to detect any SA0 fault. The same applies for SA1 faults, but with complement addresses.

Bridges between data lines: The detection of wired-AND or wired-OR bridges between data lines requires that each pair of data lines must fulfill at least the combination 0-1 or 1-0. Modified Counting Sequence (MCS) satisfies this requirement at a cost of $log_2(L_d + 2)$ test patterns [23]. The total number of memory operations required to execute such test patterns equals $2 \cdot log_2(L_d + 2)$ memory operations; for each pattern there is a write and read operation (to any address). The effectiveness of these patterns is proven in literature [23]. Complex bridge faults, such as A-dominant AND B, require both 0-1 and 1-0 inputs on each pair of wires. The True/Complement Algorithm [24] can be used for this; it consists of $2 \cdot log_2(L_d + 2)$ test patterns resulting into $4 \cdot log_2(L_d + 2)$ memory operations.

Bridges between address lines: Wired-And and wired-OR faults between address lines must be considered separately.
Wired-AND bridge fault: Wired-AND bridge faults can be detected by a walking-1 pattern, similar to the detection of SA0 faults in address lines (left side of Table II); due to wired-AND fault operations OP2 till OP17 will overwrite *Init_Data* of OP1. This is detected by OP18.
Wired-OR bridge fault: Wired-OR bridge faults can be detected by a walking-0 pattern, similar to the detection of SA1 faults in address lines (right side of Table II). The walking-1 sequence for wired-AND faults and walking-0 for wired-OR detect both simple and complex bridge faults.

TABLE III
BRIDGE FAULTS TEST PATTERNS THAT FLIP DATA LINES

OP	Op.	Address	Data
OP1	*W*	0000 0000 0000 0000	F F F F
OP2	*R*	0000 0000 0000 0000	F F F F

OP	Op.	Address	Data
OP1	*W*	1111 1111 1111 1111	0 0 0 0
OP2	*R*	1111 1111 1111 1111	0 0 0 0

TABLE IV
BRIDGE FAULTS TEST PATTERNS THAT FLIP ADDRESS LINES

OP	Op.	Address	Data
OP1	*W*	0000 0000 0000 0000	F F F F
OP2	*W*	0000 0000 0000 000**1**	0 0 0 0
OP3	*W*	0000 0000 0000 00**1**0	0 0 0 0
...	...	...	...
OP16	*W*	0**1**00 0000 0000 0000	0 0 0 0
OP17	*W*	**1**000 0000 0000 0000	0 0 0 0
OP18	*R*	0000 0000 0000 0000	F F F F

OP	Op.	Address	Data
OP1	*W*	1111 1111 1111 1111	0 0 0 0
OP2	*W*	1111 1111 1111 111**0**	F F F F
OP3	*W*	1111 1111 1111 11**0**1	F F F F
...	...	...	...
OP16	*W*	1**0**11 1111 1111 1111	F F F F
OP17	*W*	**0**111 1111 1111 1111	F F F F
OP18	*R*	1111 1111 1111 1111	0 0 0 0

Bridges between data and address lines: Bridge faults behave as wired-AND or wired-OR, and may cause data or address lines to flip.

Bridge faults that flip data lines: The left side of Table III provides the memory operations that detect wired-AND bridge faults that lead to faulty data lines. Any data line that suffers from a wired-AND with an address line will cause the data line to flip to zero (on the data side), which is easily detectable. These test patterns are similar to those of SA0 in data lines when $Addr\_x$ of Table I is set to value 0. The right side of Table III provides the test patterns that detect wired-OR bridge faults that lead to faulty data lines. Here, the operations take the complement values of those of wired-AND. Any data line suffering from a wired-OR with an address line will cause the data line to flip to one, which is easily detectable. These test patterns are similar to those of SA1 in data lines when $Addr\_y$ of Table I is set to a value of all 1's.

Bridges that flip address lines: The left part of Table IV provides the test patterns needed for the detection of wired-AND bridges that cause address lines to flip. A walking-1 pattern on the address lines ensures the detection of these types of faults. In OP1, the address consisting of all 0's is initialized with all 1's data (FFFF in hex). Note that for the initialization pattern (OP1) the address is not impacted in the presence of wired-AND faults. Any address line that suffers from a wired-AND with a data line will cause the address line to flip to zero during the walking-1 sequence (OP2 up to OP17). This will overwrite the original initialization. Therefore, the last read (OP18) results in a data value of FFFF for non-faulty interconnects and 0000 in case a fault is present. These test patterns are similar to those in the left side of Table II used to detect SA0 faults in address lines when *Init_Data* = FFFF and $Data_x$ = 0000. The right part of Table IV provides the test patterns needed to detect wired-OR bridges that cause address lines to flip. Here, all address and data values are the complements of the wired-AND patterns. The memory operations are the same as to test for SA1 faults in address lines (right part of Table II) under the condition that *Init_Data* = 0000 and $Data_x$ = FFFF.

B. Dynamic Faults

Dynamic faults consist of single and multi line faults. For single line faults the same general approach as static faults can be used in which data lines are tested in parallel and address lines individually. However, for multi-line faults the layout of the address and data lines becomes important. For simplicity, we assume a regular TSV array of size 4×4 to demonstrate how to generate test patterns for SOF with Crosstalk. Furthermore, we assume a 1^{st} aggressor model, i.e., victims can only be affected by closest neighbor aggressors. Grouping the 4×4 matrix in four groups as shown in Fig. 3 allows us to test each group simultaneously. For example, when TSVs of group 1 are tested as victims it is assumed that the remaining TSVs act as aggressors. The same applies for the other three TSV groups. In general any k^{th} aggressor model can be used, where k the maximum TSV distance between victims and aggressors. Results reported in [25] show that restricting to k=1 is sufficient.

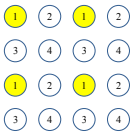


Fig. 3. TSV groups

TABLE V
SOF WITH CROSSTALK TEST PATTERNS FOR DATA LINES

OP	Operation	Address	Data
OP1	W	$Addr_1$	**0**000 0000 **0**000 0000
OP2	W	$Addr_2$	**1**010 0000 **1**010 0000
OP3	R	$Addr_1$	0000 0000 0000 0000
OP4	R	$Addr_2$	1010 0000 1010 0000
OP5	W	$Addr_1$	**1**111 1111 **1**111 1111
OP6	W	$Addr_2$	**0**101 1111 **0**101 1111
OP7	R	$Addr_1$	1111 1111 1111 1111
OP8	R	$Addr_2$	0101 1111 0101 1111

TABLE VI
SOF WITH CROSSTALK TEST PATTERNS FOR ADDRESS LINES

OP	Operation	Address	Data
OP1	W	00000000 0 0000000	Init_Data
OP2	W	00000000 1 0000000	$Data_1$
OP3	R	00000000 0 0000000	Init_Data
OP4	W	11111111 1 1111111	Init_Data
OP5	W	11111111 0 1111111	$Data_1$
OP6	R	11111111 1 1111111	Init_Data

SOF with Crosstalk at data lines: Table V shows the memory operations required to detect SOF with Crosstalk for TSV group 1. To sensitize such a fault, a transition must be created on the victim while keeping the aggressors stable. OP1-OP2 create a 0→1 transition from master to slave on the victim data lines, while keeping aggressors stable at 0. OP3 and OP4 make a similar transition, but from slave to master. In case the transition fails (during write or read) it will be detected during reading (OP3-OP4). In a similar manner, but with all data lines complemented, OP5-OP8 can be applied to detect the 1→0 transition fault on the victim. Similar patterns can be developed for the other remaining three groups.

SOF with Crosstalk at address lines: Table VI shows the test pattern to detect SOF with crosstalk for a single address line. OP1 initializes the memory by writing *Init_Data* to address 0. OP1 and OP2 create a 0→1 transition on the victim address line while the aggressors are kept stable at 0. OP3 expects *Init_Data* in case fault-free, and $Data_1$ if the victim line failed to make the 0→1 transition. Similarly, OP4-OP6 detect the reverse transition. These group of test patterns have to be repeated for each address line individually.

It is worth noting that the minimum set required to detect all static and dynamic faults targeted in this paper consist of only two test: (1) PDF with Crosstalk and (2) SOF with Crosstalk.

TABLE VII
TEST COST FOR STATIC FAULTS

Fault (set)	#mem ops.	# MIPS instr.	#MIPS cycles
Optimized SAF	32	45	57
Optimized static / Optimized Bridge (simple bridge)	48	109	137
Optimized static / Optimized Bridge (complex bridge)	62	137	179

TABLE VIII
TEST COST FOR DYNAMIC FAULTS

Fault (set)	#mem ops.	# MIPS instr.	#MIPS cycles
PDF at data lines	6	16	21
PDF at address lines	10	15	23
SOF at data lines	4	14	19
SOF at address lines	48	75	91
Crosstalk / (PDF + crosstalk) at data lines	24	58	66
Crosstalk / (PDF + crosstalk) at address lines	96	123	175
Stuck open fault (SOF) with Crosstalk at data lines	32	61	73
Stuck open fault (SOF) with Crosstalk at address lines	72	92	104

IV. EXPERIMENTAL RESULTS

A. Case Study

We simulate memory test patterns, for a memory die stacked on a logic die that consists of a MIPS64 processor, by using the MIPS64 simulator in [26]. The simulator can handle a maximum of L_d=64-bit data lines and L_a=12-bit address lines (lowest 3 bits are byte offset). The simulator supports three types of instructions: (1) ALU instructions such as add, subtract and shift, (2) Branch instructions such as branch if equal, and (3) Memory instructions such as load, store, etc; a complete reference can be found in [27].

The memory operations, which represent the test patterns, need to be translated into real MIPS instructions. An example for the SAF at data lines is provided in the code fragment below.

```
1. ori  r1,r0,0xFFFF    8. SD  R1, 0xFF8(R0)
2. dsll r1,r1,16        9. LD  R10,0XFF8(R0)
3. ori  r1,r1,0xFFFF   10. BNE R1,R10,SA0_DATA
4. dsll r1,r1,16       11. HALT
5. ori  r1,r1,0xFFFF
6. dsll r1,r1,16           SA0_DATA:
7. ori  r1,r1,0xFFFF       ;handle fault here
```

The test consists of 11 instructions. The first 7 instructions create the desired pattern FFFF FFFF FFFF FFFF in register R1 (similarly as in the left side of Table II). Instructions 8 and 9 contain the two memory operations in which R1 is written (SD) and read (LD) from memory. In case a stuck at fault is present a branch will be taken (instruction 10) to SA0_DATA.

Tables VII and VIII summarize the number of memory operations and clock cycles for all static and dynamic faults respectively. The tables provide for each fault the required number of memory operations, the number of MIPS instructions to execute those memory operations and finally, the number of MIPS cycles. For example, to test for all static faults requires only 179 MIPS cycles. The memory latency is 1 clock cycle.

B. Comparison with Prior Related Work

We compare our MBIT approach with BS, TL and the BIST methods [12,13] for several DfT requirements related to test quality (T1) and cost (T2).

T1 Test quality: The test methodology must support full controllability and observability and test for static and dynamic faults. In addition, diagnosis should identify faulty locations. Modifying test patterns for extra diagnosis or to target different faults is needed.

T2 Test cost: The DfT overhead should be as low as possible and preferably without DfT on the memory die. The test time should be cost-effective; i.e., the test time should be reasonable and scalable with the number of TSVs.

TABLE IX
COMPARISON BETWEEN INTERCONNECT TEST APPROACHES

Test Requirement	Boundary Scan	Test Logic	BIST [12]	BIST [13]	MBIT
T1 controllability/observability	Both	Memory outputs are only observable	both	both	Address lines are tested indirectly
T1 static/dynamic	Only static	Only static	crosstalk only	crosstalk only	Static + Dynamic
T1 detection/diagnosis	Support for both	Support for both	Support for both	Support for both	Support for both
T1 flexible test patterns	yes	yes, limited output controllability	no	no	yes
T2 area overhead	$2 \cdot (L_a + L_c + 2 \cdot L_d)$ BS cells (bottom/top die) + JTAG (top die)	$L_a + L_c + 2 \cdot L_d$ BS cells (bottom die) and test logic (top die)	9.8% with respect to TSV array	7% with respect to TSV array	No area overhead
T2 test cost (simple bridges)	$2 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	$(L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	not applicable	not applicable	$2 \cdot log_2(L_d + 2) + 2 \cdot L_a + 8$ at speed memory operations
T2b test cost (complex bridges)	$4 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	$2 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	not applicable	not applicable	$4 \cdot log_2(L_d + 2) + 2 \cdot L_a + 8$ at speed memory operations

Test quality comparison

Table IX summarizes the comparison between the five test methods. All approaches are in general able to control and observe the interconnects. TL has a limited controllability of memory outputs and MBIT propagates faults in address lines indirectly. BS and TL can be used for static faults only, while the approaches in [12] and [13] perform testing by hardwired state machines and target crosstalk faults only. MBIT is flexible enough to test for any fault. BS and TL can be modified for dynamic fault testing, but require extra hardware or complete cell modification [28,29]. BS interconnect testing has an additional limitation for the case where drivers and receiver cells cannot be tested simultaneously; in this case, approximately 75% of the drivers and receivers can be covered [4]. A similar problem exists in [12] and [13] as both solutions only can handle uni-directional lines. MBIT is able to test for both TSV drivers and receivers as patterns are applied in both directions. Diagnosis is possible for all cases, however, the schemes in [12,13] cannot apply flexible patterns as the BISTs are hardwired, while in TL some test patterns might not be applicable due to memory input output dependancy during test.

Test cost comparison

For a fair area overhead comparison, we assume a bottom die with default JTAG. In that case, the overhead for each method will be the following:

- BS: the overhead consists of the additional BS cells on both the bottom die and top die assigned to the interconnects, in total equal to $2 \cdot (L_a + L_c + 2 \cdot L_d)$. Here L_a presents the number of address line, L_c the number of control lines, L_d the number of data lines. Control and address lines require a single BS cell per wire, while bi-directional data lines are assumed to have two BS cells [30]. In addition to BS cells, the JTAG infrastructure on the top die is also part of the overhead.

- TL: the overhead includes the BS-cells on the bottom die of length L_a+L_c+$2 \cdot L_d$ and the test logic on top die.
- BIST [12,13]: the overhead consists in both methods of a state-machine, several flip-flops and other control logic such as muxes. In [13] its reported that the area overhead of the method in [12] approximates 9.8%, while their own equals 7%; both are measured with respect to the total TSV area. It is evaluated in 90 nm technology using 15 μm TSV diameters using a 64×16 TSV matrix.
- MBIT: no area overhead.

The test time for each of the approaches is as follows:

- BS: the total test time for BS depends on the number of test patterns and the length of the BS cells. For the True/Complement Algorithm, the number of test patterns equal $2 \cdot \lceil log_2(L_a + L_c + L_d + 2) \rceil$ to detect all static faults. The length of the BS cells equals $2 \cdot (L_a + L_c + 2 \cdot L_d)$. Therefore, the test time equals $4 \cdot (L_a + L_c + 2 \cdot L_d) \cdot \lceil log_2(L_a + L_c + L_d + 2) \rceil$ test clock cycles.
- TL: The test time reduces by a factor of two when compared to BS, due to half the number of BS cells.
- BIST [12,13]: The test time of the hardwired BISTs in [12,13] is much lower than other approaches. For example, the method in [13] requires 122 cycles (assuming 1 cycle per TSV row pattern) to detect all targeted faults in this paper (i.e., the test set PDF with crosstalk and SOF with crosstalk faults).
- MBIT: To detect all static faults 179 MIPS cycles are required (assuming complex bridges). To detect all static and dynamic faults (PDF with crosstalk and SOF with crosstalk faults), MBIT requires 66+175+73+104=418 at speed cycles (see Table VIII).

In conclusion, with respect to the area overhead MBIT performs best followed by BIST [12], BIST [13], TL and BS. If we compare MBIT with BS and TL with respect to test time considering the same MIPS memory (L_a=12, L_d=64 and for simplicity ignore control lines L_c=0), BS based testing would require 3920 test clock cycles and Test Logic based testing 1960 test clock cycles for True/Complement Algorithm. Moreover, if we assume an operational clock frequency of 500 MHz and test clock speed of 100 MHz the differences between the methods becomes more apparent. The total test time would be 0.36μs, 39.20μs and 19.6μs for MBIT, BS and TL respectively. If we compare MBIT with the hardwired BIST solutions for both dynamic and static faults, we see that MBIT is slower in test time (418 cycles for MBIT versus 122 cycles for BIST [13]), but has the flexibility of applying different test patterns and does not require additional DfT.

V. Conclusion

This paper proposed a new Memory Based Interconnect Test (MBIT) approach for 3D-SICs where memory is stacked on logic by testing interconnects through memory read and write operations. Our MBIT solution is able to perform at-speed testing and detect all static and dynamic faults. It has zero area overhead and allows flexible patterns to be applied. In addition, the required test time is much lower than traditional based solutions such as Boundary Scan, but is three times slower than hardwired BIST solutions. However BIST solutions have a large area overhead and cannot apply flexible patterns.

References

[1] P. Garrou *et al.*, *Handbook of 3D Integration*. John Wiley & Sons, 2008.

[2] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[3] (2013) P1838 - standard for test access architecture for three-dimensional stacked integrated circuits. [Online]. Available: http://standards.ieee.org/develop/project/1838.html

[4] S. Deutsch *et al.*, "Dft architecture and atpg for interconnect tests of jedec wide-i/o memory-on-logic die stacks," in *ITC*, 2012, pp. 1–10.

[5] H. Ehrenberg and B. Russell, "IEEE Std 1581- A standardized test access methodology for memory devices," in *ITC*, 2011, pp. 1–9.

[6] E.J. Marinissen and Y. Zorian, "Testing 3D chips containing through-silicon vias," in *ITC*, Nov. 2009, pp. 1–11.

[7] K. Chakrabarty *et al.*, "TSV defects and TSV-induced circuit failures: The third dimension in test and design-for-test," in *IRPS*, April 2012, pp. 5F.1.1 –5F.1.12.

[8] A. Papanikolaou *et al.*, *Three Dimensional System Integration*. Springer US, 2011.

[9] F. Ye and K. Chakrabarty, "TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation," in *49th ACM/EDAC/IEEE DAC*, June 2012, pp. 1024–1030.

[10] A.E. Engin and S.R. Narasimhan, "Modeling of crosstalk in through silicon vias," *IEEE Trans. on Electromagnetic Compatibility*, vol. PP, no. 99, pp. 1 –10, 2012.

[11] C. Liu *et al.*, "Full-chip tsv-to-tsv coupling analysis and optimization in 3d ic," in *48th ACM/EDAC/IEEE DAC*, 2011, pp. 783–788.

[12] V. Pasca *et al.*, "Configurable Thru-Silicon-Via interconnect Built-In Self-Test and diagnosis," in *12th LATW*, 2011, pp. 1–6.

[13] Y.J. Huang *et al.*, "Post-bond test techniques for TSVs with crosstalk faults in 3D ICs," in *VLSI-DAT*, 2012, pp. 1–4.

[14] L. Chen *et al.*, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," in *Design Automation Conference*, 2001, pp. 317–322.

[15] S. Kannan *et al.*, "Fault Modeling and Multi-Tone Dither Scheme for Testing 3D TSV Defects," *J. Electronic Testing*, vol. 28, no. 1, 2012.

[16] C.W. Kuo and H.Y. Tsai, "Thermal stress analysis and failure mechanisms for through silicon via array," in *ITherm*, June 2012, pp. 202–206.

[17] X. Liu *et al.*, "Failure mechanisms and optimum design for electroplated copper Through-Silicon Vias," in *ECTC*, May 2009, pp. 624 –629.

[18] M. Jung *et al.*, "Full-chip through-silicon-via interfacial crack analysis and optimization for 3D IC," in *ICCAD*. Piscataway, NJ, USA: IEEE Press, 2011, pp. 563–570.

[19] E. Beyne and I. Dewolf. (2013, July) Failure analysis for 3d tsv systems. [Online]. Available: http://www.sematech.org/meetings/archives/3d/10124/pres/Beyne.pdf

[20] E.J. Marinissen, "Testing tsv-based three-dimensional stacked ics," in *DATE*, march 2010, pp. 1689 –1694.

[21] D. Jung *et al.*, "Disconnection failure model and analysis of tsv-based 3d ics," in *EDAPS*, Dec 2012, pp. 164–167.

[22] M. Cuviello *et al.*, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," in *ICCAD*, 1999, pp. 297–303.

[23] P. Goel and M.T. McMahon, "Electronic chip-in-place test," in *DAC*. Piscataway, NJ, USA: IEEE Press, 1982, pp. 482–488. [Online]. Available: http://dl.acm.org/citation.cfm?id=800263.809248

[24] P.T. Wagner, "Interconnect testing with boundary scan," in *International Test Conference (ITC '87)*, Sep. 1987, pp. 52–57.

[25] R. Weerasekera *et al.*, "Compact modelling of through-silicon vias (tsvs) in three-dimensional (3-d) integrated circuits," in *3D-IC*, 2009, pp. 1–8.

[26] (2013) Winmips64. [Online]. Available: http://indigo.ie/ mscott/

[27] (2013) Mips64 architecture for programmers volume ii: The mips64 instruction set. [Online]. Available: http://scc.ustc.edu.cn/zlsc/lxwycj/200910/W020100308600769158777.pdf

[28] M. Tehranipour *et al.*, "Testing soc interconnects for signal integrity using boundary scan," in *VLSI Test Symposium*, 2003, pp. 158–163.

[29] S. Park and T. Kim, "A new ieee 1149.1 boundary scan design for the detection of delay defects," in *DATE*, 2000, pp. 458–462.

[30] N.K. Jha and S. Gupta, *Testing of Digital Systems*. Cambridge, United Kingdom: Cambridge University Press, 2003.

Post-Bond Interconnect Test and Diagnosis for 3D Memory Stacked on Logic

Mottaqiallah Taouil, *Student Member, IEEE,* Mahmoud Masadeh, and Said Hamdioui, *Senior Member, IEEE,*

***Abstract*—Three-dimensional stacked IC (3D-SIC) technology based on Through-Silicon Vias (TSVs) provides numerous advantages as compared to traditional 2D-ICs. A potential application is memory stacked on logic, providing enhanced throughput, and reduced latency and power consumption. However, testing the TSV interconnects between the two dies is challenging as both the memory and the logic dies might come from different providers. Currently, no standard exists and the proposed solutions fail to address dynamic and time-critical faults (at speed testing). In addition, memory vendors have not been in favor to put additional DfT structures such as JTAG for interconnect testing on their memory devices. This paper proposes a new Memory Based Interconnect Test (MBIT) approach for 3D memories stacked on logic (e.g. CPU's). A structural approach is used to develop fault models, their detection conditions, and test and diagnosis patterns. The test patterns are applied by read and write instructions to the memory and are validated by a case study where a 3D memory is assumed to be stacked on a MIPS64 processor. The main benefits of the MBIT approach are: (1) zero area overhead, (2) the ability to detect both static and dynamic faults and perform at speed testing, (3) flexibility in applying any test pattern, as this can be executed by the CPU on the logic die (4) extreme short test execution time, and (5) the ability to perform interconnect diagnosis.**

Keywords: *interconnect testing, 3D-SIC, memory-on-logic*

I. Introduction

The popularity of 3D Stacked ICs (3D-SICs) is rising among industry and research groups [1]. 3D-SICs based on Through Silicon Vias (TSVs) are emerging as one of the main competitors to continue the trend of Moore's Law [2]. Stacking dies with vertical interconnects possess many benefits [1], such as (a) low latency between adjacent dies, (b) reduced power consumption, (c) high bandwidth communication, (d) improved form factor and package volume density, (e) heterogeneous integration, etc.

One of the main applications that utilizes the mentioned benefits is the stacking of memory (DRAM) on logic (CPU). After stacking, a post-bond interconnect test is required to test interconnects (TSVs + μ-bumps) between the memory and logic dies. This is not straightforward as (1) stacked dies may come from different providers (IP confidentiality), (2) memory providers are reluctant to integrate DfT such as JTAG for interconnect testing, and (3) even with DfT support, obtaining high coverage for dynamic faults is still challenging.

M. Taouil and S. Hamdioui are with the Department of Computer Engineering, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands. E-mail: {M.Taouil, S.Hamdioui}@Tudelft.nl

Currently, no standard exists to test interconnects in memories stacked on logic. However, some test approaches are being under development. IEEE P1838 [3] is currently an ongoing standard that develops DfT for general stacked ICs; it is based on the presence of Boundary Scan (BS) cells in all dies. Wide I/O [4] also supports interconnect testing using BS. However, (DRAM) memory vendors are not always in favor of integrating JTAG on their devices [5]. Other approaches such as the IEEE P1581 [5], originally for 2D ICs, can be extended in the third dimension. In test mode, the memory is bypassed and interconnects are tested by creating a direct logic function between the inputs and outputs of the memory. IEEE P1581 prefers a JTAG compliant logic chip, i.e., the test logic on the memory chip can function with a logic chip that supports JTAG. This standard can be mapped to 3D-SICs by having (a) the bottom die (logic) JTAG compliant and (b) the test logic residing on the top die (memory). This approach, referred to as Test Logic (TL) based interconnect testing, also requires additional DfT test logic on the memory die. In addition to the undesired DfT on the memory die, both the BS and TL based test methods are unable to provide at speed testing required to target dynamic faults. Testing for dynamic faults is crucial, as 3D interconnects are expected to suffer from speed and timing related faults [6–11]. BS and TL can both be used to perform diagnosis for static faults.

In addition to the lack of standards, limited dedicated test solutions with at-speed testing capability for TSV crosstalk faults have been published. In [12,13], authors present hardware BIST approaches to test the TSV interconnects. Both methods are not flexible in altering test patterns and require additional DfT area; it is assumed to be 7% for [13] and 9.8% for [12] when considering 90 nm technology using 15 μm TSV diameters. In [14] the authors test the memory interconnects using the embedded CPU. They target crosstalk faults in planar dies. However, the authors did not address diagnosis. Moreover, the layout of a TSV array in 3D-SICs differs from wire connections in planar ICs.

This paper proposes a post-bond Memory Based Interconnect Test (MBIT) methodology being able to test interconnects between memory and logic dies by performing read and write operations from the logic die (CPU) to the memory dies. It provides a classification of interconnect defects, and compiles them into fault models. In addition, it discusses the test pattern generation for these faults and uses the proposed MBIT to implement them. For diagnosis purposes, several algorithms are presented allowing

to perform maximum fault diagnosis. MBIT does not require any DfT area as it reuses existing components in the stack. It supports at-speed testing and detects static and dynamic faults. Moreover, it is very flexible in altering test patterns simply by modifying software instructions and has a extreme short test execution time.

The remainder of the paper is organized as follows. Section II presents defects, fault models and detection conditions for 3D interconnects. Section III provides the test pattern generation for the targeted fault models. Section IV discusses the diagnosis algorithms for the fault models under consideration. Section V provides the simulation results and compares our methodology with the state-of-the-art. Finally, Section VI concludes this paper.

II. Defect, Fault Models and Detection

This section describes first general terminology regarding defects, faults and fault detection. Thereafter, it presents an overview of typical defects in the 3D interconnects. Each defect is subsequently compiled in its fault abstraction. Finally, detection conditions are given for each fault.

A. Terminology

Next, we describe briefly the keywords that are used in the rest of the paper. A *defect* is a physical imperfection that may cause a chip to malfunction. Defects are typically modeled at a higher abstraction level by *faults*. A fault may represent one or more defects with the same or similar fault behavior. A collection of faults with similar properties are grouped in a *fault model*. Faults can be detected by applying a sequence of test vectors; the obtained test responses are compared with golden fault-free responses. The fraction of detectable faults, the *fault coverage*, indicates the quality of the test. In case the fault coverage of the test algorithm is insufficient faulty chips might pass the test; they are referred to as *test escapes*.

In this work, a *test algorithm* is considered to be a sequence of memory write and read operations (i.e., test vectors) applied to target interconnect faults. Subsequently, a *test response* presents the logic value on the interconnects retrieved using a read operation. The difference between the expected fault-free response and the actual *test response* is called the *fault syndrome*. Note that not all input vectors trigger faulty test responses in the presence of faults. In case a test pattern does not trigger a fault to be visible at the output, we speak of an *aliasing syndrome*. Finally, a *comfounding syndrome* refers to the case where the presence of two or more faults lead to the same faulty *test response* for a given test vector. Our objective in this paper is to provide high quality tests at low cost, while also being able to perform *maximum diagnosis*, i.e., not only identifying the fault location but also the fault type.

B. Defects in Interconnects

Interconnects in 3D-stacking are a potential source of defects inherent to the manufacturing steps such as TSV fabrication/filling, bonding etc. Defects may occur both in TSVs and micro-bumps and a list of common defects are given next.

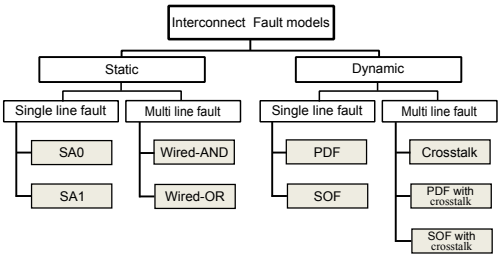


Fig. 1. Fault Model Classification for interconnects

Defects related to Through-Silicon-Via (TSV):

- D1 Pinhole defects occur along TSV walls and cause a short (low resistance path) between TSVs and the substrate; This may cause degradation of the signal quality in terms of strength and speed [6,7,9,15].
- D2 An incomplete fill of TSVs (voids) may originate from insufficient wetting during plating. Voids cause partial opens and lead to higher TSV resistance [6,7,9,15].
- D3 Coefficient of thermal expansion (CTE) mismatch between TSV metal (most likely copper) and substrate may lead to TSV cracks and sidewall delamination. Both lead to increased path resistance [9,15–18].
- D4 Pinch-off of TSVs during plating could lead to increased TSV resistance or partial opens [7].
- D5 Missing contacts between TSVs and the transistors or metal layer cause opens [7,8].
- D6 TSV misalignment with μ-bumps increases the resistance and causes (partial) opens [7,9,15].
- D7 Crosstalk between different TSVs [9,10].

Defects related to μ-bumps:

- D8 Damage in underlying BEOL [19].
- D9 Weak bonding due to buckled thinned Si chip [19].
- D10 Variation in TSV heights may cause tin to be squeezed out from μ-bump causing shorts between μ-bumps [19,20].
- D11 Electromigration may cause voids and cracks in the joints, resulting in higher resistive μ-bumps, or opens [21].
- D12 Cracks in μ-bumps due to CTE mismatch between copper, silicon, and silicon-oxide [7].

C. Faults and Fault Models

Interconnect fault models can be classified into static and dynamic faults. Static faults are fixed and time independent, while dynamic faults may change over time. Fig. 1 shows a classification of the faults. A defect can cause a single line or a multi line fault. Each fault is depicted in Figure 2 and explained next. Static faults include:

- Stuck-at-Fault (SAF). There are two types of SAF faults: stuck-at-0 (SA0) and stuck-at-1 (SA1) as depicted in Fig. 2(b). A SAF fault can be caused by defect D1.
- Bridge fault. *Simple* bridge faults include wired-AND (Fig. 2(c)) and wired-OR (Fig. 2(d)) faults. *Complex* bridge faults also exists, such as A dominate-AND B in

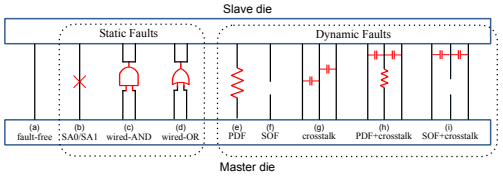


Fig. 2. Static and dynamic faults

which wire A is fault-free and where B takes the value A ∩ B. A bridge fault can be caused by defect D10.

Dynamic faults include:

- Path Delay Fault (PDF): A partial open line defect increases the line delay (Fig. 2(e)). It can affect both rising or falling delay time. PDF faults can be caused by defects D2, D3, D6, D8, D11, D12.
- Stuck Open Fault (SOF): This is caused by a complete open line defect (Fig. 2(f)). SOFs can be caused by defects D5, D6, D8, D9, D11, D12. Note that we assumed SOF to be a dynamic fault as the logic value on the floating end may change over time.
- Crosstalk Fault: Faults on victim lines are caused by crosstalk from aggressive neighbors (Fig. 2(g)). Several crosstalk faults exists as described by the Maximum Aggressor (MA) fault model [22] such as (1) glitch-up, (2) glitch-down, (3) falling delay, and (4) rising delay. Each fault has a specific behavior, while it represents the same phenomena. Defect D7 may cause crosstalk faults.
- PDF with Crosstalk (PDFC): Faults due to partial resistive opens (victims) are affected by crosstalk from neighbors (Fig. 2(h)). PDF with Crosstalk faults can be caused by combinations of crosstalk and PDF faults.
- SOF with Crosstalk (SOFC): Faults due to complete open lines (victims) are affected by crosstalk from neighbors (Fig. 2(i)). SOF with Crosstalk faults can be caused by combinations of crosstalk and SOF faults.

The dynamic faults embody most of the defects and therefore their detection is essential to guarantee high product quality.

D. Detection Conditions

The detection conditions of each fault are described next. In general, the detection process adheres to the following steps:

1) Fault sensitization (activation): create a difference between faulty and fault-free circuits.
2) Fault propagation: make the fault visible at the outputs.
3) Fault justification: backtrack the values to the primary inputs of the circuit, such that the inputs sensitize the fault.

Fault propagation and justification for address and data lines are dissimilar. Data lines can be controlled and observed directly through writing and reading. Therefore, fault propagation and justification are straightforward. However, address lines are uni-directional and fault propagation must be performed indirectly by utilizing data lines (e.g., by writing and reading different values to different addresses). Control lines such as write or read enable are tested implicitly. For fault sensitization, special sequences and/or transitions are required for each fault.

SAF (SA0/SA1): A stuck-at-fault forces a wire to a specific value; either 0 (SA0) or 1 (SA1). Therefore, to sensitize a SAF fault an opposite value must be applied to the wire.
Bridge fault (Wired-AND/Wired-OR): To sensitize a bridge fault, two opposite values must be specified on each pair of lines. Simple bridge faults such as the ones depicted in Fig. 2(c) and (d) require at least one of the two patterns 0-1 or 1-0 as inputs. More complex bridges such as A dominate-AND B require both 0-1 and 1-0 inputs on each pair of wires for fault sensitization.
PDF: We assume that a path delay fault is caused by a low to moderate resistive open defect, violating the normal operation with at most one additional clock cycle. Faults that create larger delays, i.e., more than one extra clock cycle, are considered as SOFs. To sensitize PDF faults, both 0→1 and 1→0 transitions should be applied on each line.
SOF: For SOFs we assume that during short time periods the non-driven part of floating lines remain unchanged. Therefore, to sensitize these faults either a 0→1 or 1→0 transition is needed.
Crosstalk: We consider only the relevant crosstalk faults of the MA fault model. These are the *rising* and *falling delay* faults. To sensitize a *rising delay* fault, a victim must undergo a 0→1 transition, while the aggressors simultaneously make a 1→0 transition. The reverse applies for *falling delay* faults.
PDFC: The fault sensitization for PDF faults with crosstalk consists of two simultaneous transitions. Up or down transitions are needed to sensitize delay faults at the resistive victims, while opposite transitions are needed on the aggressors to maximize the stress. The sensitization of this fault is equivalent to falling and rising delay faults.
SOFC: The fault sensitization for this fault requires both a 0→1 transition on the victim (i.e., the wire that contains the SOF), while keeping the aggressors unchanged at 0 and a 1→0 transition on the victim while keeping the aggressors unchanged at 1. Keeping the aggressors unchanged reduces the coupling with the floating part of the SOF, hence it minimizes the contribution of the aggressors to the transition on the floating part of the victim.

III. Test Pattern Generation

In this section, we generate test patterns for each fault. We assume the presence of a single fault at a time. In addition, during explanation we assume that the number of data lines is L_d=16 and that of address lines L_a=16. Nevertheless, the approach is scalable to any number of data and address lines. In this work we target the testing of TSVs, micro-bumps as well as TSV drivers and receivers on both the memory and logic die.

A. Static Faults

In this section, we will present the test patterns of static faults. A SAF fault may happen in data lines or address lines. A bridge fault may happen: (1) between data lines, (2) between address lines, and (3) between data and address

TABLE I
SAF TEST PATTERNS FOR DATA LINES

SA0 at data lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	$Addr_x$	F F F F
OP2	*R*	$Addr_x$	F F F F

SA1 at data lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	$Addr_y$	0 0 0 0
OP2	*R*	$Addr_y$	0 0 0 0

TABLE II
SAF TEST PATTERNS FOR ADDRESS LINES

SA0 at address lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	0000 0000 0000 0000	Init_Data
OP2	*W*	0000 0000 0000 000**1**	$Data_1$
OP3	*W*	0000 0000 0000 00**1**0	$Data_2$
...	...	...	...
OP16	*W*	0**1**00 0000 0000 0000	$Data_{15}$
OP17	*W*	**1**000 0000 0000 0000	$Data_{16}$
OP18	*R*	0000 0000 0000 0000	Init_Data

SA1 at address lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	1111 1111 1111 1111	Init_Data
OP2	*W*	1111 1111 1111 111**0**	$Data_1$
OP3	*W*	1111 1111 1111 11**0**1	$Data_2$
...	...	...	...
OP16	*W*	1**0**11 1111 1111 1111	$Data_{15}$
OP17	*W*	**0**111 1111 1111 1111	$Data_{16}$
OP18	*R*	1111 1111 1111 1111	Init_Data

lines. Each category is described next.

SAF at data lines: Table I shows the memory operations required to detect SA0 (left table) and SA1 faults (right table) on data lines. The tables consists of four columns; the first column shows the operation number (OP#); the second column the type of operation, i.e., read (R) or write (W); the third and fourth columns show the address value presented in binary value (if applicable) and the data value presented in hexadecimal value (if applicable), respectively. For example, the detection of SA0 or SA1 in any data line requires only two memory operations consisting of a write followed by a read irrespective of the number of data lines. Any address ($Addr_x$) may be used for this. SAF faults will be detected during OP2.

SAF at address lines: Table II shows the test patterns to detect both SA0 (left table) and SA1 faults (right table) in address lines. Detecting SA0/SA1 faults at address lines is more complex as they affect the memory address. For example, writing a value to the address "all 1's" and subsequently reading from this address will not detect any SA0 fault in the address lines; this is because both the write and read addresses are affected in the same way and detecting the fault is not guaranteed. To test memory address lines, each address line should be tested separately. For example, by using a walking-1 sequence for SA0 as depicted in the left table. Address 0 of the memory is first initialized to *Init_Data* during OP1. During write operations OP2 to OP17 (with different data than *Init_Data*), any SA0 in the address lines will overwrite *Init_Data* of address 0. Therefore, read operation OP18 is able to detect any SA0 fault. The same applies for SA1 faults, but with complement addresses. Detecting each of the SAF faults at address lines requires L_a+2 memory operations. Note that *Data*$_1$ to *Data*$_{16}$ can have the same value.

Bridges between data lines: The detection of wired-AND and wired-OR bridges between data lines requires that each pair of data lines must be set to at least one of the combinations 0-1 or 1-0. Modified Counting Sequence (MCS) satisfies this requirement at a cost of $\lceil log_2(L_d+2) \rceil$ test patterns [23]. The total number of memory operations required to execute such test patterns equals $2 \cdot \lceil log_2(L_d+2) \rceil$ memory operations; each MCS pattern is written and thereafter read using any address. The effectiveness of these patterns is proven in literature [23]. Complex bridge faults, such as dominant-AND require setting each pair of data lines to both combinations 0-1 and 1-0. The True/Complement Algorithm [24] can be used for this; it consists of $2 \cdot \lceil log_2(L_d+2) \rceil$ test patterns resulting into $4 \cdot \lceil log_2(L_d+2) \rceil$ memory operations.

Bridges between address lines: Wired-And and wired-OR faults between address lines must be considered separately.
Wired-AND bridge fault: Wired-AND bridge faults can be detected by a walking-1 pattern, similar to the detection of SA0 faults in address lines (left side of Table II); due to wired-AND fault operations OP2 till OP17 will overwrite *Init_Data* of OP1; the fault is detected by OP18.
Wired-OR bridge fault: Wired-OR bridge faults can be detected by a walking-0 pattern, similar to the detection of SA1 faults in address lines (right side of Table II). It is worth noting that the walking-1 sequence for wired-AND faults and walking-0 for wired-OR detect both simple and complex bridge faults (see Section II-C). Each sequence consists of L_a+2 memory operations.

Bridges between data and address lines: Bridge faults may cause data or address lines to flip. Each category is described next.

Bridge faults that flip data lines: The left side of Table III provides the memory operations that detect wired-AND bridge faults that lead to faulty data lines. Any wired-AND fault between a data line and an address line will cause the data line to flip to zero, which is thereafter easily detectable. These two test patterns are similar to those of SA0 in data lines when $Addr\_x$ of Table I is set to value 0. The right side of Table III provides the test patterns that detect wired-OR bridge faults that lead to faulty data lines. Here, the operations take the complement values of those of wired-AND. Any wired-OR fault between a data line and an address line will cause the data line to flip to one, which is thereafter easily detectable. These two test patterns are similar to those of SA1 in data lines when $Addr\_y$ of Table I is set to a value of all 1's.

Bridges that flip address lines: The left part of Table IV provides the test patterns needed for the detection of wired-AND bridges that cause address lines to flip. A walking-1 pattern on the address lines ensures the detection of these types of faults. OP1 initializes the memory word with address "all 0's" to data "all 1's" (FFFF in hex); note that this address is not impacted by wired-AND faults. Any address line that suffers from a wired-AND with a data line will cause the address line to flip to zero during the walking-1 sequence (OP2 up to OP17). This will overwrite the initialization. Therefore, read OP18 results in a data value of FFFF for non-faulty interconnects and 0000 in case a fault is present. These test patterns are similar to those in the left side of Table II used to detect SA0 faults in address lines when *Init_Data* = FFFF and $Data_x$ = 0000. The right part of Table IV provides the test patterns needed to detect wired-OR bridges that cause address lines to flip. Here, all address and data values are the complements of the wired-AND patterns. The memory operations are the same as those for SA1 faults in address

TABLE III
BRIDGE FAULTS TEST PATTERNS THAT FLIP DATA LINES

Wired-AND that flip data lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	0000 0000 0000 0000	F F F F
OP2	*R*	0000 0000 0000 0000	F F F F

Wired-OR that flip data lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	1111 1111 1111 1111	0 0 0 0
OP2	*R*	1111 1111 1111 1111	0 0 0 0

TABLE IV
BRIDGE FAULTS TEST PATTERNS THAT FLIP ADDRESS LINES

Wired-AND that flip address lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	0000 0000 0000 0000	F F F F
OP2	*W*	0000 0000 0000 000**1**	0 0 0 0
OP3	*W*	0000 0000 0000 00**1**0	0 0 0 0
...	...	...	...
OP16	*W*	0**1**00 0000 0000 0000	0 0 0 0
OP17	*W*	**1**000 0000 0000 0000	0 0 0 0
OP18	*R*	0000 0000 0000 0000	F F F F

Wired-OR that flip address lines			
OP#	**Op.**	**Address**	**Data**
OP1	*W*	1111 1111 1111 1111	0 0 0 0
OP2	*W*	1111 1111 1111 111**0**	F F F F
OP3	*W*	1111 1111 1111 11**0**1	F F F F
...	...	...	...
OP16	*W*	1**0**11 1111 1111 1111	F F F F
OP17	*W*	**0**111 1111 1111 1111	F F F F
OP18	*R*	1111 1111 1111 1111	0 0 0 0

TABLE V
PDF TEST PATTERNS

PDF at data lines			
OP#	**Op.**	**Address**	**Data**
OP1	W	$Addr_1$	0 0 0 0
OP2	W	$Addr_2$	F F F F
OP3	W	$Addr_1$	0 0 0 0
OP4	R	$Addr_1$	0 0 0 0
OP5	R	$Addr_2$	F F F F
OP6	R	$Addr_1$	0 0 0 0

PDF at address lines			
OP#	**Op.**	**Address**	**Data**
OP1	W	1111 1111 1111 1111	Init_Data
OP2	W	1111 1111 1111 1111	Init_Data
OP3	W	0000 0000 0000 0000	$Data_1$
OP4	W	1111 1111 1111 1111	$Data_2$
OP5	R	1111 1111 1111 1111	$Data_2$
OP6	W	0000 0000 0000 0000	Init_Data
OP7	W	0000 0000 0000 0000	Init_Data
OP8	W	1111 1111 1111 1111	$Data_1$
OP9	W	0000 0000 0000 0000	$Data_2$
OP10	R	0000 0000 0000 0000	$Data_2$

lines (right part of Table II) under the condition that *Init_Data* = 0000 and $Data_x$ = FFFF.

B. Dynamic Faults

Dynamic faults consist of single and multi-line faults. For single line faults the same general approach as static faults can be used in which data lines are tested in parallel and address lines individually. However, for multi-line faults the physical layout of the address and data lines becomes important. We assume that no dynamic coupling faults can occur between address and data lines. This can be guaranteed if power supply TSVs are placed between the address and data lines.

PDF Faults

PDF faults at data lines: Table V shows the test patterns to detect PDF faults in data (left side). These patterns guarantee the detection of delays within a timing violation of a single clock cycle. The write operations OP1-OP3 consist of a 0→1 followed by a 1→0 transition from master to slave, while read operations OP4-OP6 carry out the same transitions but from slave to master. Any rising or falling transition fault, from master to slave or vice versa, is visible at the output during the reads OP4-OP6. In total, 6 memory operations are required.

PDF faults at address lines: Detecting a path delay fault at an address line requires 0→1 and 1→0 transitions. The test patterns are shown at the right side of Table V. The first part of the table (OP1-OP5) tests the 0→1 transitions, and the second part (OP6-OP10) tests the 1→0 transitions. OP1 and OP2 ensure a fault-free initialization of the address consisting of "all 1's" to *Init_Data*, even in the presence of PDF faults, as PDF faults are assumed to violate the timing with at most one clock cycle. 0→1 transitions are subsequently created on all address lines by OP3 and OP4 to test for rising delay faults. Lines that fail to make this transition at OP4 are detected by OP5. In the fault free case, i.e. if all transitions occurred, $Data_2$ is expected. However, the presence of a PDF fault will result in reading $init\_data$, as at least one transition created at OP3-OP4 fails. Detecting 1→0 transition faults (falling PDF faults) using OP6-OP10 is performed in a similar way but with complement addresses. In total, 10 memory operations are required.

TABLE VI
SOF TEST PATTERNS

SOF at data lines			
OP#	**Op.**	**Address**	**Data**
OP1	W	$Addr_1$	0000 0000 0000 0000
OP2	W	$Addr_2$	1111 1111 1111 1111
OP3	R	$Addr_1$	0000 0000 0000 0000
OP4	R	$Addr_2$	1111 1111 1111 1111

SOF at address lines			
OP#	**Op.**	**Address**	**Data**
OP1	W	0000 0000 0000 000**0**	$Data_1$
OP2	W	0000 0000 0000 000**1**	$Data_2$
OP3	R	0000 0000 0000 000**0**	$Data_1$
OP4	R	0000 0000 0000 000**1**	$Data_2$

SOF Faults

The stuck open fault represents a complete open line. Such a fault may occur at data lines or address lines; both are described next.

SOF faults at data lines: Test patterns to detect stuck open faults at data lines are shown at the left part of Table VI. It contains two write operations (OP1 and OP2) and two read operations (OP3 and OP4); each pair of operations creates 0→1 transitions. During the two read operations, we assume that floating data lines on the master's side could carry logic values belonging to one of the following three cases:

1) The line maintains a stable value **0** during both read operations. OP4 detects this fault.
2) The line maintains a stable with value **1** during both read operations. OP3 detects this fault. Note that the floating line could maintain this logic value 1 due to the write operation OP2.
3) The value on the line changes during the read operations (i.e., from value 1 during OP3 to value 0 during OP4 due to leakage). Note that the master floating data line is last set to 1 before reading operation OP3 and OP4 start. This fault is detected by both OP3 and OP4.

Only 4 memory operations are required to execute this test.

SOF faults at address lines: A stuck open fault at address lines can be tested based on walking patterns. Similarly to SOFs in data lines, two 0→1 transitions are created. However, in this case each bit line must be tested separately as shown at the right part of Table VI for the LSB bit only. Any floating address line will be detected as the 0→1 transitions will not change the floating end (on the slave side) of the address lines. In total, $4{\cdot}L_a$ memory operations are required to execute this test. Note that the presented patterns also detect PDF faults that violate the clock cycle more than one clock cycle.

Crosstalk/PDFC faults

As victims of crosstalk faults are affected by its neighbors it is important to consider the physical layout of the address and data lines. For simplicity, we assume a regular TSV array of size 4×4 to demonstrate how to generate test patterns for these types of faults. Furthermore, we assume a 1^{st} aggressor model, i.e., victims can only be affected by closest

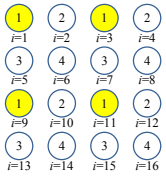


Fig. 3. TSV groups: victim Group 1 and aggressor Groups 2, 3 and 4.

TABLE VII
CROSSTALK/PDFC TEST PATTERNS AT DATA LINES

Crosstalk falling delay at victim data lines			
OP#	**Operation**	**Address**	**Data**
OP1	Write	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
OP2	Write	$Addr_2$	**0**1**0**1 1111 **0**1**0**1 1111
OP3	Read	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
OP4	Read	$Addr_2$	**0**1**0**1 1111 **0**1**0**1 1111
...	...	...	...
OP13	Write	$Addr_1$	0000 **0**1**0**1 0000 **0**1**0**1
OP14	Write	$Addr_2$	1111 **1**0**1**0 1111 **1**0**1**0
OP15	Read	$Addr_1$	0000 **0**1**0**1 0000 **0**1**0**1
OP16	Read	$Addr_2$	1111 **1**0**1**0 1111 **1**0**1**0

Crosstalk rising delay at victim data lines			
OP#	**Operation**	**Address**	**Data**
OP1	Write	$Addr_1$	**0**1**0**1 1111 **0**1**0**1 1111
OP2	Write	$Addr_2$	**1**0**1**0 0000 **1**0**1**0 0000
OP3	Read	$Addr_1$	**0**1**0**1 1111 **0**1**0**1 1111
OP4	Read	$Addr_2$	**1**0**1**0 0000 **1**0**1**0 0000
...	...	...	...
OP13	Write	$Addr_1$	1111 **1**0**1**0 1111 **1**0**1**0
OP14	Write	$Addr_2$	0000 **0**1**0**1 0000 **0**1**0**1
OP15	Read	$Addr_1$	1111 **1**0**1**0 1111 **1**0**1**0
OP16	Read	$Addr_2$	0000 **0**1**0**1 0000 **0**1**0**1

TABLE VIII
OPTIMIZED CROSSTALK/PDFC TEST PATTERNS AT DATA LINES

OP#	Operation	Address	Data
OP1	W	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
OP2	W	$Addr_2$	**0**1**0**1 1111 **0**1**0**1 1111
OP3	W	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
OP4	R	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
OP5	R	$Addr_2$	**0**1**0**1 1111 **0**1**0**1 1111
OP6	R	$Addr_1$	**1**0**1**0 0000 **1**0**1**0 0000
...	...	...	...
OP19	W	$Addr_1$	0000 **0**1**0**1 0000 **0**1**0**1
OP20	W	$Addr_2$	1111 **1**0**1**0 1111 **1**0**1**0
OP21	W	$Addr_1$	0000 **0**1**0**1 0000 **0**1**0**1
OP22	R	$Addr_3$	0000 **0**1**0**1 0000 **0**1**0**1
OP23	R	$Addr_2$	1111 **1**0**1**0 1111 **1**0**1**0
OP24	R	$Addr_1$	0000 **0**1**0**1 0000 **0**1**0**1

neighbor aggressors. In general any k^{th} aggressor model can be used, where k the maximum TSV distance between victims and aggressors. Results reported in [25] show that restricting to k=1 is sufficient. Fig. 3 shows us how to group the 4×4 matrix in four groups using the 1^{st} aggressor model; it allows us to test each group simultaneously. For example, when TSVs of Group 1 are tested as victims it is assumed that the remaining TSVs act as aggressors. The victims for Group 1 include TSVs with i=1, 3, 9 and 11 (see Fig. 3). The same applies for the other three TSV groups.

Crosstalk/PDFC faults in data and address lines are described next.

Crosstalk faults/PDFC at data lines: Test patterns to detect falling/rising delay faults at the data lines are shown at the left/right part of Table VII. The victims undergo a falling/rising transition, while the aggressors undergo the opposite transition. The patterns in the table are only shown for TSVs belonging to Group 1 (OP1-OP4) and Group 4 (OP13-OP16), where it is assumed that the MSB (LSB) of the pattern present the value of TSV with i=1 (i=16). OP1 and OP2 initiate a 1→0 or 0→1 transition to test for falling or rising delay faults on the victim data lines, while aggressors make opposite transitions. The data value is presented in a binary number for a better clarification. Note that the opposite transitions on the victim and aggressors are executed in both directions from master to slave (OP1 and OP2) and vice versa (OP3 and OP4). In a similar way, test patterns can be created for the other two groups.

The test patterns in Table VII (both the left and right part) can be optimized further; the result is presented in Table VIII. Here, only 6 test patterns per group suffice to create all down (OP1 and OP2 from master to slave, OP4 and OP5 from slave to master) and up (OP2 and OP3 from master to slave, OP5 and OP6 from slave to master) transitions for the victims and their opposite ones for the aggressors. Therefore, the number of memory operations to detect crosstalk at data lines is 6 (patterns) · 4 (groups) = 24 operations.

TABLE IX
CROSSTALK/PDFC TEST PATTERNS AT ADDRESS LINES

Crosstalk falling delay at victim address lines			
OP#	**Operation**	**Address**	**Data**
OP1	Write	1111 1111 1111 1111	Init_Data
OP2	Write	0000 0000 0000 000**1**	$Data_1$
OP3	Write	1111 1111 1111 111**0**	$Data_2$
OP4	Read	1111 1111 1111 1111	Init_Data

Crosstalk rising delay at victim address lines			
OP#	**Operation**	**Address**	**Data**
OP1	Write	0000 0000 0000 0000	Init_Data
OP2	Write	1111 1111 1111 111**0**	$Data_1$
OP3	Write	0000 0000 0000 000**1**	$Data_2$
OP4	Read	0000 0000 0000 0000	Init_Data

TABLE X
SOFC TEST PATTERNS FOR DATA LINES

OP#	Operation	Address	Data
OP1	W	$Addr_1$	**0**0**0**0 0000 **0**0**0**0 0000
OP2	W	$Addr_2$	**1**0**1**0 0000 **1**0**1**0 0000
OP3	R	$Addr_1$	0000 0000 0000 0000
OP4	R	$Addr_2$	1010 0000 1010 0000
...	.	...	...
OP17	W	$Addr_1$	**1**1**1**1 1111 **1**1**1**1 1111
OP18	W	$Addr_2$	**0**1**0**1 1111 **0**1**0**1 1111
OP19	R	$Addr_1$	1111 1111 1111 1111
OP20	R	$Addr_2$	0101 1111 0101 1111
...	.	...	...

Crosstalk/PDFC faults at address lines: Each address line TSV needs to be tested individually for crosstalk/PDFC faults. As each victim line is tested separately, we assume the rest of the lines to be aggressors. The left side of Table IX shows the test patterns required to detect a falling delay fault at the LSB address bit. OP1 initializes the memory value at the address of "all 1's" with *Init_Data*. OP2 and OP3 create a 1→0 transition on the victim address line and a 0→1 transition on the aggressors (i.e., all other address lines). In case the LSB bit of the address lines fails to make the transition due to crosstalk, OP3 overwrites the initialization value *Init_Data*. Therefore, the read operation (OP4) from the address consisting of "all 1's" results in data value *Init_Data* in case the address line is fault-free and in $Data_2$ in case a fault is present. Similarly, but with opposite patterns, the right side of Table IX shows the patterns to detect rising delay faults. In total 8·L_a memory operations are required to detect crosstalk/PDFC faults in all address lines.

SOFC Faults

A SOFC fault may happen at data or address lines. The two cases are described below.

SOFC at data lines: Table X shows the memory operations required to detect SOFC faults for TSVs located in Group 1. To sensitize such a fault, a transition must be created on the victims, while keeping the aggressors unchanged. OP1 and OP2 create this 0→1 transition from master to slave for the victim lines of Group 1, while keeping the aggressors (TSVs from Group 2, 3, and 4) unchanged at 0. OP3 and OP4 make the same transitions, but from slave to master. In case any transition fails (during either writing or reading), it will be detected during read operations OP3 and OP4

TABLE XI
SOFC TEST PATTERNS FOR ADDRESS LINES

OP#	Operation	Address	Data
OP1	W	0000 0000 0000 000**0**	Init_Data
OP2	W	0000 0000 0000 000**1**	$Data_1$
OP3	R	0000 0000 0000 000**0**	Init_Data
OP4	W	1111 1111 1111 111**1**	Init_Data
OP5	W	1111 1111 1111 111**0**	$Data_1$
OP6	R	1111 1111 1111 111**1**	Init_Data

as failed transitions are directly visible at the data output. Similarly, but with complemented data lines, OP17-OP20 can be applied to detect 1→0 transition faults on the victims with aggressors kept unchanged at 1. Similar patterns can be developed for the other remaining three groups. For the SOFC patterns, we changed the order of patterns to reduce the background noise. We assume first that all patterns with zero background are tested followed by patterns with a one background. For example, in Table X, OP1-OP4 present the patterns for Group 1 with zero background, while patterns OP16-OP20 have a one background. In total 8 (patterns) · 4 (groups) =32 memory operations are required.

SOFC at address lines: Table XI shows the test patterns required to detect a SOFC fault for the LSB address bit line; each address line must be tested separately. OP1 initializes the memory at address "all 0's" by writing a particular data value (*Init_Data*). OP2 and OP3 create subsequently a 0→1 transition on the victim address line, while the aggressors are kept unchanged at 0. In the fault-free case, the data *Init_Data* is expected at OP3. However, if a SOFC fault occurs, faulty output $Data_1$ is expected. Similarly, OP4-OP6 detect the opposite transition. In total $6{\cdot}L_a$ memory operations are required.

It is worth noting that the *minimum set* required to detect *all* static and dynamic faults targeted in this paper consist of only four tests: PDFC for data lines (PDFC_D), PDFC for address lines (PDFC_A), SOFC for data lines (SOFC_D), SOFC for address lines (SOFC_A); see Table VIII to XI, respectively. The proof of this is given in the next section.

IV. DIAGNOSIS

This section presents a methodology to perform MBIT fault diagnosis. This post-bond diagnosis can take place either off-line or on-line due to the availability of the CPU. Similarly as for test patterns, we perform diagnosis using the available CPU. We propose several algorithms to perform the interconnect diagnoses. First, we analyze the diagnosis capabilities of the minimum test set required for the detection of faults targeted in this paper. Subsequently, we augment this set with additional test patterns to realize *maximum diagnosis* if needed; more test patterns for diagnosis might be required in case faulty responses cannot uniquely identify the fault type or location. The diagnosis process consists of three steps:

1) Initialization: This step is required to obtain proper fault-free address and data reference values to perform data and address diagnosis, respectively. For example, to apply the minimum test set for data faults (i.e., PDFC_D of Table VIII and SOFC_D of Table X) two addresses ($Addr_1$ and $Addr_2$ in the tables) with a fault free behavior must be identified. We refer to these addresses as $A1$ and $A2$, respectively. Similarly, two data values $D1$ and $D2$ have to be identified to apply the PDFC_A (Table IX) and SOFC_A (Table XI) test patterns. Note that in Table IX $Data_1$ and $Data_2$ can have the same value.
2) Data diagnosis: Apply PDFC_D and SOFC_D test patterns using $A1$ and $A2$. If maximum diagnosis is not achieved, new diagnosis patterns should be added.
3) Address diagnosis: Apply PDFC_A and SOFC_A test patterns using $D1$ and $D2$. If maximum diagnosis is not achieved, new diagnosis patterns should be added.

Next, these steps will be elaborated in detail.

```
Alg. 1: [A1, A2, D1, D2] = Initialize()

    //initialization of A1, A2, D1 and D2
 1. A1 = 0;
 2. D1 = 0;

 3. for i=1:La // for all address bits
 4.     for j=1:Ld // for all data bits

 5.         A2 = sll(1, i-1);
 6.         D2 = sll(1, j-1);

            // verify current A1, A2, D1, D2
 7.         write A1  D1;
 8.         write A2  D2;
 9.         write A1  D1;

10.         read  A1  x;
11.         read  A2  y;
12.         read  A1  x;

13.         if (x==D1 and y==D2)
14.             return(A1, A2, D1, D2);
15.         endif

16.     endfor

17. endfor

18. signal_error();
```

Fig. 4. Pseudo code of the proposed find_A1A2_D1D2() algorithm.

A. Step 1: Initializing $A1$, $A2$, $D1$, $D2$

In this section, we present the algorithm *initialize()* to identify appropriate values for $A1$, $A2$, $D1$ and $D2$ as shown in Figure 4. Such values require to be distinctive by only a single bit. The algorithm initializes $A1$ and $D1$ to all 0's (lines 1 and 2 of the algorithm) and subsequently performs a walking 1 sequence on both $A2$ and $D2$ (performed by shift left operations on lines 5 and 6, respectively), until it finds two fault free address and data values ($A1$, $A2$, $D1$ and $D2$). To obtain such values, fault free up- and down-transitions (without a specific order) need to be made for the two address and two data values. This is performed by creating both write (lines 7 till 9) and read transitions (lines 10 till 12) on the address and data lines. When the read responses match the write responses

TABLE XII
COMPLETE READ OPERATIONS OF OPTIMIZED TEST

Read Patterns for PDFC		
Group	**Index**	**Data**
1	R4	1010 0000 1010 0000
1	R5	0101 1111 0101 1111
1	R6	1010 0000 1010 0000
2	R10	0101 0000 0101 0000
2	R11	1010 1111 1010 1111
2	R12	0101 0000 0101 0000
3	R16	0000 1010 0000 1010
3	R17	1111 0101 1111 0101
3	R18	0000 1010 0000 1010
4	R22	0000 0101 0000 0101
4	R23	1111 1010 1111 1010
4	R24	0000 0101 0000 0101

Read Patterns for SOFC		
Group	**Index**	**Data**
1	R3	0000 0000 0000 0000
1	R4	1010 0000 1010 0000
2	R7	0000 0000 0000 0000
2	R8	0101 0000 0101 0000
3	R11	0000 0000 0000 0000
3	R12	0000 1010 0000 1010
4	R15	0000 0000 0000 0000
4	R16	0000 0101 0000 0101
1	R19	1111 1111 1111 1111
1	R20	0101 1111 0101 1111
2	R23	1111 1111 1111 1111
2	R24	1010 1111 1010 1111
3	R27	1111 1111 1111 1111
3	R28	1111 0101 1111 0101
4	R31	1111 1111 1111 1111
4	R32	1111 1010 1111 1010

(the check is performed on line 13), appropriate values for $A1$, $A2$, $D1$ and $D2$ are returned on line 14 and the algorithm ends. In case all address and/or all data lines are faulty no appropriate values can be identified and an error is signaled (line 18). Note that this algorithm is able to produce suitable values in the simultaneous presence of both address and data lines faults. The worst case complexity of the algorithm equals $\Omega(L_a \cdot L_d)$.

B. Step 2: Diagnosis of Data Lines

To diagnose faults in data lines, we first apply the minimum test set for data line fault detection, i.e., tests PDFC_D (Table VIII) and SOFC_D (Table X). The read operations of these tests are repeated for convenience and are tabulated at the left and right side of Table XII, respectively. Both tables contain three columns; the first column shows the TSV group under test, the second column shows the index of the read operation denoted by R (which is the same as OP in the original tables), and the third column shows the expected fault-free test response.

Next, we analyze the impact of these patterns in the presence of faults. Tables XIII and XIV show the expected fault responses (similar to fault syndromes) when the patterns are applied while assuming faults to be present. The first columns of the tables list the simulated fault; the second columns the fault location expressed by the TSV group (the failing TSV bit line index can easily be derived); the remainder columns shows for each read operations whether or not the fault can be detected; all the read indexes are taken from Table XII; e.g., 'column 4' in Table XIII denotes read R4 in the left part of Table XII.

We will first discuss the results of the PDFC_D patterns shown in Table XIII. We explain the table by giving examples for each fault. SA0 faults in data lines of Group 1 are expected to fail at read operations R4, R6, R11, R17, and R23. For example, if the MSB-bit of the data line (belonging to Group 1) contains a SA0, faulty test responses will be directly visible on R4, R6, R11, and R23 as during those read a 1 is expected. Similar conclusions can be made for the remainingSA0 and SA1 faults and for the other three groups.

For bridge faults we assume two TSVs from different groups to be involved. For example, simple AND-bridges between

TABLE XIII
DATA LINE FAULT RESPONSE USING PDFC PATTERNS

Fault	**Group**	**4**	**5**	**6**	**10**	**11**	**12**	**16**	**17**	**18**	**22**	**23**	**24**
SA0	G1	x		x		x			x			x	
SA0	G2		x		x		x		x			x	
SA0	G3		x			x		x		x		x	
SA0	G4		x			x			x		x		x
SA1	G1		x		x		x	x		x	x		x
SA1	G2	x		x		x		x		x	x		x
SA1	G3	x		x	x		x		x		x		x
SA1	G4	x		x	x		x	x		x		x	
Bridge-AND	G1-G2	1	2	1	2	1	2						
Bridge-OR	G1-G2	2	1	2	1	2	1						
Bridge-AND	G1-G3	1	3	1				3	1	3			
Bridge-OR	G1-G3	3	1	3				1	3	1			
Bridge-AND	G1-G4	1	4	1							4	1	4
Bridge-OR	G1-G4	4	1	4							1	4	1
Bridge-AND	G2-G3				3	2	3	2	3	2			
Bridge-OR	G2-G3				2	3	2	3	2	3			
Bridge-AND	G2-G4				4	2	4				2	4	2
Bridge-OR	G2-G4				2	4	2				4	2	4
Bridge-AND	G3-G4							4	3	4	3	4	3
Bridge-OR	G3-G4							3	4	3	4	3	4
PDF/PDFC Fall	G1		x			x				x			x
PDF/PDFC Fall	G2			x		x				x			x
PDF/PDFC Fall	G3			x			x		x				x
PDF/PDFC Fall	G4			x			x			x		x	
PDF/PDFC Rise	G1	x		x		x			x			x	
PDF/PDFC Rise	G2		x		x		x		x			x	
PDF/PDFC Rise	G3		x			x		x		x		x	
PDF/PDFC Rise	G4		x			x			x		x		u
SOF	G1	z	o	z	o	z	o	o	z	o	o	z	o
SOF	G2	o	z	o	z	o	z	o	z	o	o	z	o
SOF	G3	o	z	o	o	z	o	z	o	z	o	z	o
SOF	G4	o	z	o	o	z	o	o	z	o	z	o	z
SOFC	All												

Group 1 (G1) and Group 2 (G2) will lead to faulty responses at read instructions R4, R5, R6, R10, R11 and R12 (denoted by the 1's and 2's in the table). A complex bridge fault affects only one of the groups; for example, complex AND-bridge faults between G1 and G2 causing faults to appear only in G1 will be predicted at R4, R6 and R11 (denoted by the 1's in table).

Note that the applied PDFC_D test can not distinguish between PDF and PDFC faults. As PDFC faults have stronger detection conditions their tests automatically also detect PDF faults. Falling delay faults for Group 1 are expected to be detected at read operation R5, R12, R18, and R24, while rising delay faults at R4, R6, R11, R17 and R23. SOF faults theoretically can fail in all read operations. The letters *z* and *o* present the SOF fault types that can be detected; a *z* presents a floating zero and *o* presents a floating one. However, SOF faults might behave as SA0 or SA1 faults during short time intervals; therefore, their fault response might overlap (partially) with SAF faults. The last entry of the table shows that SOFC faults cannot be detected irrespective of the TSV group.

Similarly, Table XIV shows the expected responses of the SOFC_D patterns in the presence of faults. The table is constructed in a similar manner; the main difference with the previous table is that this test test can detect SOFC faults in stead of PDFC faults. The two tables clearly show that the PDFC_D and SOFC_D patterns detect all faults considered in this paper as previously mentioned.

Next, we analyze the combined responses for diagnosis, i.e., unique fault location and fault type. To identify the faulty locations is relatively easy as the TSV group of the failing data lines is known beforehand. The failed indexes can be directly obtained by xor-ing the test response with the golden reference value. Careful analysis of the table reveals that *each fault* has a unique signature, except for SOF. A SOF fault might show

TABLE XIV
DATA LINE FAULT RESPONSE USING SOFC PATTERNS

Fault	Group	3	4	7	8	11	12	15	16	19	20	23	24	27	28	31	32
SA0	G1		x							x		x	x	x	x	x	x
SA0	G2				x					x	x	x		x	x	x	x
SA0	G3						x			x	x	x	x	x		x	x
SA0	G4								x	x	x	x	x	x	x	x	
SA1	G1	x		x	x	x	x	x	x		x						
SA1	G2	x	x	x		x	x	x	x				x				
SA1	G3	x	x	x	x	x		x	x						x		
SA1	G4	x	x	x	x	x	x	x									x
Bridge-AND	G1-G2		1		2						2		1				
Bridge-OR	G1-G2		2		1						1		2				
Bridge-AND	G1-G3		1				3				3				1		
Bridge-OR	G1-G3		3				1				1				3		
Bridge-AND	G1-G4		1						4		4						1
Bridge-OR	G1-G4		4						1		1						4
Bridge-AND	G2-G3				2		3						3		2		
Bridge-OR	G2-G3				3		2						2		3		
Bridge-AND	G2-G4				2				4				4				2
Bridge-OR	G2-G4				4				2				2				4
Bridge-AND	G3-G4						3		4						4		3
Bridge-OR	G3-G4						4		3						3		4
PDF Fall	G1										x						
PDF Fall	G2												x				
PDF Fall	G3														x		
PDF Fall	G4																x
PDF Rise	G1		x														
PDF Rise	G2				x												
PDF Rise	G3						x										
PDF Rise	G4								x								
PDFC	All																
SOF	G1	o	z	o	o	o	o	o	o	z	o	z	z	z	z	z	z
SOF	G2	o	o	o	z	o	o	o	o	z	z	z	o	z	z	z	z
SOF	G3	o	o	o	o	o	z	o	o	z	z	z	z	z	o	z	z
SOF	G4	o	o	o	o	o	o	o	z	z	z	z	z	z	z	z	o
SOFC	G1		x														
SOFC	G2				x								x				
SOFC	G3						x								x		
SOFC	G4								x								x

unpredictable test responses; fault signatures that do not match any of the targeted faults are assumed to be of SOFs. Special care is also required for the PDF and PDFC faults. In case a fault matches the signature of PDF/PDFC faults in Table XIII, but not the PDF faults of Table XIV, the fault is identified as a PDFC fault. However, if the fault matches both PDF/PDFC and PDF signatures of the two tables, then the fault is assumed to be a PDF fault. Note that the detection condition of a PDFC fault is stronger than that of a PDF fault.

C. Step 3: Diagnosis in Address Lines

The test patterns that detect PDFC_A and SOFC_A faults in address lines (see tables IX and XI) target a single fault at a time. Therefore, these tests can easily diagnose the location of faulty interconnects. Drawback, however, is that comfounding syndromes are expected to occur as many faults will trigger the same faulty test response; this makes it more complex to identify the fault type. Therefore, additional instructions are required to identify this fault type.

The algorithm that contains these additional instructions is shown in Figure 5. At the higher level, it distinguishes faults in address lines into three categories: (i) SA0, AND-bridges and rising PDF faults, (ii) SA1, OR-bridges and falling PDF faults, and (iii) PDFC and SOFC faults. The algorithm (referred to as diag_interconnect()) has three inputs: two data values $D1$ and $D2$ obtained using the algorithm of Figure 4, and an array F containing the faulty bit response of the applied minimum test set for the targeted address faults (PDFC_A and SOFC_A); the value is set high in this array if at least one of the two tests failed.

For each faulty address line i (denoted as $F[i]$) the algorithm identifies which of the category it belongs to. Address $A1$ is initialized to 0 (line 1) and the faulty address line is activated in address $A2$ (line 3). By creating a zero to one transition on the faulty line i using addresses A1 and A2 (lines 5 and 6) followed by a read (line 7), SA0, AND bridges

```
Alg. 2: [type] = diag_interconnect(D1, D2, F)

//initialization of A1, A2
 1. A1 = 0;

 2. for i = 1:La   // La is number of address bits
 3.    A2 = sll(1, i-1);

 4.    if (F[i])

 5.       write A1 D1;  // write addr = 0...0
 6.       write A2 D2;  // write to bit under test
 7.       read  A2 x;   // verify bit under test

 8.       write not(A1) D1;  // write addr = 1...1
 9.       write not(A2) D2;  // write to tested bit
10.       read  not(A2) y;  // verify tested bit

11.       if (x!=D2)
12.          type[i] = diag_SA0_AND_RPDF(i, D1, D2);
13.       endif

14.       if (y!=D2)
15.          type[i] = diag_SA1_OR_FPDF(i, D1, D2);
16.       endif

17.       if (x==D1 and y==D2)
18.          type[i] = diag_PDFC_SOFC(i);
17.       endif
18.    endif
19. endfor
```

Fig. 5. Pseudo code of the proposed diag_interconnect() algorithm.

and rising PDF (RPDF) delays can be identified as faults that fail the read operation (lines 11-13). Similarly, by creating an opposite transition on the address lines (lines 8-10), SA1, OR-bridges and falling PDF faults can be identified (lines 14-16). If none of the previous faults occurred for given $F[i] = 1$, then the faulty address line i must be either PDFC or SOFC (line 17 and 18).

The algorithm in Figure 6 uses a straightforward method to differentiate between a static (SA0 or AND-bridge) fault and a RPDF fault. A rising transition is created first on the faulty address line i by addresses $A1$ and $A2$ on line 3 and 4; subsequently, by writing again to address A2 (line 5) static faults can be distinguished from the RPDF fault. If a RPDF fault is present the transition on line 3-4 will not happen in time; this overwrites the value at address $A1$ to D2. The next write on line 5 will write the value $D1$ correctly to $A2$ (note that the PDF fault is not active anymore in this cycle). Therefore, if address $A1$ is read (line 6) the value of D2 is expected in the presence of a RPDF fault; this is checked in line 7 and 8. In case a static fault (SA0 or AND-bridge) is present, the value of address $A2$ is mapped onto address $A1$. The expected read value on line 6 will be correct. However, further analysis are needed to distinguish between SA0 and AND-bridge faults.

To discriminate between the SA0 and AND-bridge faults we will change addresses $A1$ and $A2$ in such a way that the AND-bridge fault is not sensitized anymore. We achieve this by setting only the direct neighbors of the faulty line i to 1 in both $A1$ and $A2$ (lines 10-12). By creating a transition now on

```
Alg. 3: [fault_type] = diag_SA0_AND_RPDF(i, D1, D2)

 1. A1 = 0;
 2. A2 = sll(1, i-1);

 3. write A1  D1; // write addr of all zero's
 4. write A2  D2; // write to bit under test
 5. write A2  D1; //
 6. read  A1  x; // read bit under test

 7. if (x != D1)  //  RPDF fault?
 8.    return RPDF_FAULT;

09.  // set all neighbors of line i high in A1
10.  for (all j that are direct neighbors of i)
11.     A1 = A1 | (1<<j);

     //activate all bits also in A2
12.  A2 = A2 | A1;

13.  write A1  D1;
14.  write A2  D2;
15.  read  A1  x;

16.  if (x == D2)
17.     return SA0_FAULT;
19.  else
19.     return BRIDGE_OR;
20.  endif
```

Fig. 6. Pseudo code of the proposed diag_SA0_AND_RPDF algorithm.

```
Algorithm 4: [type] = diag_SA1_OR_FPDF(i, D1, D2)

 1. A1 = 1111...111;
 2. A2 = sll(1, i-1);
 3. A2 = not(A2);

 4. write A1  D1; // write addr of all zero's
 5. write A2  D2; // write to bit under test
 6. write A2  D1; //
 7. read  A1  x; // read bit under test

 8. if (x != D1)  //  FPDF fault?
 9.    return FPDF_FAULT;

    // set all neighbors of i low in A1
10. A1 = 0;
11. for (all x that are direct neighbors of i)
12.    A1 = A1 | (1<<x);
13. endfor
14. A1 = not(A1);

    //activate all neighbor bits also in A2
15. A2 = A2 & not(A1);

16. write A1  D1;
17. write A2  D2;
18. read  A1  x;

19. if (x == D2)
20.    return SA1_FAULT;
21. else
22.    return BRIDGE_AND;
23. endif
```

Fig. 7. Pseudo code of the proposed diag_SA1_OR_FPDF algorithm.

the faulty bit address i, while its direct neighbors are kept at 1, we can differentiate between SAF fault or bridge faults. In the presence of a SA0, the addresses $A1$ and $A2$ on lines 12 and 14 will have the same value. However, this is not the case for a bridge-AND as the fault will not be sensitized. Further diagnoses can be made to identify the precisely identify the index of the other faulty line by having a walking 0 on the neighbors of address line i. For the sake of simplicity, they are not included in this paper.

Similarly, but with reversed address patterns, the algorithm in Figure 7 can diagnose SA1, OR-bridges and FPD faults. The last function diagnose_PDFC_SOFC() in the algorithm diagnose_interconnect() of Figure 5 to identify between PDFC and SOFC faults can be easy performed by identifying which of the two test failed, i.e., either the PDFC_A test or SOFC_A test as these two faults have unique appearance.

V. Experimental Results

A. Case Study

We simulate memory test patterns, for a memory die stacked on a logic die that consists of a MIPS64 processor, by using the MIPS64 simulator in [26]. The simulator can handle a maximum of L_d=64-bit data lines and L_a=12-bit address lines (lowest 3 bits are byte offset). The simulator supports three types of instructions: (1) ALU instructions such as add, subtract and shift, (2) Branch instructions such as branch if equal, and (3) Memory instructions such as load, store, etc; a complete reference can be found in [27].

The memory operations, which represent the test patterns, need to be translated into real MIPS instructions. An example for the SAF at data lines is provided in the code fragment below.

```
1. ori  r1,r0,0xFFFF   8. SD  R1, 0xFF8(R0)
2. dsll r1,r1,16       9. LD  R10,0XFF8(R0)
3. ori  r1,r1,0xFFFF  10. BNE R1,R10,SA0_DATA
4. dsll r1,r1,16      11. HALT
5. ori  r1,r1,0xFFFF
6. dsll r1,r1,16          SA0_DATA:
7. ori  r1,r1,0xFFFF      ;handle fault here
```

The test consists of 11 instructions. The first 7 instructions create the desired pattern FFFF FFFF FFFF FFFF in register R1 (similarly as in the left side of Table II). Instructions 8 and 9 contain the two memory operations in which R1 is written (SD) and read (LD) from memory. In case a stuck at fault is present a branch will be taken (instruction 10) to SA0_DATA.

Tables XV and XVI show the number of memory operations and clock cycles for all static and dynamic faults respectively. The tables provide for each fault the required number of memory operations, the number of MIPS instructions to execute those memory operations and finally, the number of MIPS cycles. For example, to test for all static faults considering simple bridge faults in this case study requires only 137 MIPS instruction cycles, and 179 cycles in case complex bridges are targeted. The memory latency is 1 clock cycle.

Diagnosis

Figure 8 shows the Matlab configuration we used to verify the diagnosis algorithms. The configuration consists of three

TABLE XV
TEST COST FOR STATIC FAULTS

Fault (set)	#mem ops.	# MIPS instr.	#MIPS cycles
SA0 at Data line/Wired-AND (between Address and Data) flip data	2	11	17
SA1 at Data line	2	4	10
SA0 at Address line/ wired-AND between address lines/wired-AND (between Address and Data) flip address	14	18	24
SA1 at Address line/Wired-OR (between Address and Data) flip address/ Wired-OR between address lines	14	17	23
Optimized SAF	32	45	57
Wired-OR (between Address and Data) flip data	4	7	15
Data bridges (Wired-OR and wired-AND) (simple)	14	63	81
Data bridges (Wired-OR and wired-AND) (complex)	28	98	130
Optimized static / Optimized Bridge (simple bridge)	48	109	137
Optimized static / Optimized Bridge (complex bridge)	62	137	179

TABLE XVI
TEST COST FOR DYNAMIC FAULTS

Fault (set)	#mem ops.	# MIPS instr.	#MIPS cycles
PDF at data lines	6	16	21
PDF at address lines	10	15	23
SOF at data lines	4	14	19
SOF at address lines	48	75	91
Crosstalk / PDFC at data lines	24	58	66
Crosstalk / PDFC at address lines	96	123	175
SOFC at data lines	32	61	73
SOFC at address lines	72	92	104

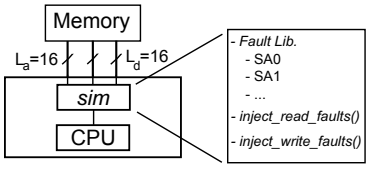


Fig. 8. Diagnosis Simulation Platform

main blocks: (i) the CPU, (ii) the fault injection unit, and (iii) the memory. We emulate a simple CPU that is able to perform read and write operations to the memory. The fault injection unit uses a fault dictionary to define and distinghuish between the faults of Section II, such as SA0 and bridge faults. Each read and write operation to the memory is intercepted by the fault injection unit and addresses or data values may be changed based on faults that are present.

Using this approach, we first verified the content of Tables XIII, XIV containing the fault signatures for data faults, and interconnect faults, i.e., simulating all faults available in fault dictionary. Subsequently, by applying the memory read and write operations we identified that first all faults are detectable and second by using a post-analysis script that each fault can be maximally diagnosed.

B. Comparison with Prior Related Work

We compare our MBIT approach with BS, TL and the BIST methods [12,13] for several DfT requirements related to test quality (T1) and cost (T2).

- T1 Test quality: The test methodology must support full controllability and observability and test for static and dynamic faults. In addition, diagnosis should identify faulty locations. Modifying test patterns for extra diagnosis or to target different faults is needed.
- T2 Test cost: The DfT overhead should be as low as possible and preferably without DfT on the memory die. The test time should be cost-effective; i.e., the test time should be reasonable and scalable with the number of TSVs.

TABLE XVII
COMPARISON BETWEEN INTERCONNECT TEST APPROACHES

Test Requirement	Boundary Scan	Test Logic	BIST [12]	BIST [13]	MBIT
T1 controllability/observability	Both	Memory outputs are only observable	both	both	Address lines are tested indirectly
T1 static/dynamic	Only static	Only static	crosstalk only	crosstalk only	Static + Dynamic
T1 detection/diagnosis	Support for both	Support for both	Support for both	Support for both	Support for both
T1 flexible test patterns	yes	yes, limited output controllability	no	no	yes
T2 area overhead	$2 \cdot (L_a + L_c + 2 \cdot L_d)$ BS cells (bottom/top die) + JTAG (top die)	$L_a + L_c + 2 \cdot L_d$ BS cells (bottom die) and test logic (top die)	9.8% with respect to TSV array	7% with respect to TSV array	No area overhead
T2 test cost (simple bridges)	$2 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	$(L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	not applicable	not applicable	$2 \cdot log_2(L_d + 2) + 2 \cdot L_a + 8$ at speed memory operations
T2b test cost (complex bridges)	$4 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	$2 \cdot (L_a + L_c + 2 \cdot L_d) \cdot log_2(L_a + L_c + L_d + 2)$ test clock cycles	not applicable	not applicable	$4 \cdot log_2(L_d + 2) + 2 \cdot L_a + 8$ at speed memory operations

Test quality comparison

Table XVII summarizes the comparison between the five test methods. All approaches are in general able to control and observe the interconnects. TL has a limited controllability of memory outputs and MBIT propagates faults in address lines indirectly. BS and TL can be used for static faults only, while the approaches in [12] and [13] perform testing by hardwired state machines and target crosstalk faults only. MBIT is flexible enough to test for any fault. BS and TL can be modified for dynamic fault testing, but require extra hardware or complete cell modification [28,29]. BS interconnect testing has an additional limitation for the case where drivers and receiver cells cannot be tested simultaneously; in this case, approximately 75% of the drivers and receivers can be covered [4]. A similar problem exists in [12] and [13] as both solutions only can handle uni-directional lines. MBIT is able to test for both TSV drivers and receivers as patterns are applied in both directions. Diagnosis is possible for all cases, however, the schemes in [12,13] cannot apply flexible patterns as the BISTs are hardwired, while in TL some test patterns might not be applicable due to memory input output dependancy during test.

Test cost comparison

For a fair area overhead comparison, we assume a bottom die with default JTAG. In that case, the overhead for each method will be the following:

- BS: the overhead consists of the additional BS cells on both the bottom die and top die assigned to the interconnects, in total equal to $2 \cdot (L_a + L_c + 2 \cdot L_d)$. Here L_a presents the number of address line, L_c the number of control lines, L_d the number of data lines. Control and address lines require a single BS cell per wire, while bi-directional data lines are assumed to have two BS cells [30]. In addition to BS cells, the JTAG infrastructure on the top die is also part of the overhead.
- TL: the overhead includes the BS-cells on the bottom die of length $L_a + L_c + 2 \cdot L_d$ and the test logic on top die.
- BIST [12,13]: the overhead consists in both methods of a state-machine, several flip-flops and other control logic such as muxes. In [13] its reported that the area overhead of the method in [12] approximates 9.8%, while their own equals 7%; both are measured with respect to the total TSV area. It is evaluated in 90 nm technology using 15 μm TSV diameters using a 64×16 TSV matrix.

- MBIT: no area overhead.

The test time for each of the approaches is as follows:

- BS: the total test time for BS depends on the number of test patterns and the length of the BS cells. For the True/Complement Algorithm, the number of test patterns equal $2 \cdot \lceil log_2(L_a + L_c + L_d + 2) \rceil$ to detect all static faults. The length of the BS cells equals $2 \cdot (L_a + L_c + 2 \cdot L_d)$. Therefore, the test time equals $4 \cdot (L_a + L_c + 2 \cdot L_d) \cdot \lceil log_2(L_a + L_c + L_d + 2) \rceil$ test clock cycles.
- TL: The test time reduces by a factor of two when compared to BS, due to half the number of BS cells.
- BIST [12,13]: The test time of the hardwired BISTs in [12,13] is much lower than other approaches. For example, the method in [13] requires 122 cycles (assuming 1 cycle per TSV row pattern) to detect all targeted faults in this paper (i.e., the test set PDF with crosstalk and SOF with crosstalk faults).
- MBIT: To detect all static faults 179 MIPS cycles are required (assuming complex bridges). To detect all static and dynamic faults (PDF with crosstalk and SOF with crosstalk faults), MBIT requires 66+175+73+104=418 at speed cycles (see Table XVI).

In conclusion, with respect to the area overhead MBIT performs best followed by BIST [12], BIST [13], TL and BS. If we compare MBIT with BS and TL with respect to test time considering the same MIPS memory (L_a=12, L_d=64 and for simplicity ignore control lines L_c=0), BS based testing would require 3920 test clock cycles and Test Logic based testing 1960 test clock cycles for True/Complement Algorithm. Moreover, if we assume an operational clock frequency of 500 MHz and test clock speed of 100 MHz the differences between the methods becomes more apparent. The total test time would be 0.36μs, 39.20μs and 19.6μs for MBIT, BS and TL respectively. If we compare MBIT with the hardwired BIST solutions for both dynamic and static faults, we see that MBIT is slower in test time (418 cycles for MBIT versus 122 cycles for BIST [13]), but has the flexibility of applying different test patterns and does not require additional DfT.

VI. Conclusion

This paper proposed a new Memory Based Interconnect Test (MBIT) approach for 3D-SICs where memory is stacked on logic by testing interconnects through memory read and write operations. Our MBIT solution is able to perform at-speed testing and detect all static and dynamic faults. It has zero area overhead and allows flexible patterns to be applied. In addition, the required test time is much lower than traditional based solutions such as Boundary Scan, but is three times slower than hardwired BIST solutions. However BIST solutions have a large area overhead and cannot apply flexible patterns.

References

[1] P. Garrou, C. Bower, and P. Ramm, *Handbook of 3D Integration*. John Wiley & Sons, 2008.

[2] G. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[3] (2013) P1838 - standard for test access architecture for three-dimensional stacked integrated circuits. [Online]. Available: http://standards.ieee.org/develop/project/1838.html

[4] S. Deutsch, B. Keller, V. Chickermane, S. Mukherjee, N. Sood, S. Goel, J. Chen, A. Mehta, F. Lee, and E. Marinissen, "Dft architecture and atpg for interconnect tests of jedec wide-i/o memory-on-logic die stacks," in *ITC*, 2012, pp. 1–10.

[5] H. Ehrenberg and B. Russell, "Ieee std 1581- a standardized test access methodology for memory devices," in *ITC*, 2011, pp. 1–9.

[6] E. Marinissen and Y. Zorian, "Testing 3d chips containing through-silicon vias," in *ITC*, Nov. 2009., pp. 1–11.

[7] K. Chakrabarty, S. Deutsch, H. Thapliyal, and F. Ye, "Tsv defects and tsv-induced circuit failures: The third dimension in test and design-for-test," in *IRPS*, April 2012.

[8] A. Papanikolaou, D. Soudris, and R. Radojcic, *Three Dimensional System Integration*. Springer US, 2011.

[9] F. Ye and K. Chakrabarty, "Tsv open defects in 3d integrated circuits: Characterization, test, and optimal spare allocation," in *49th ACM/EDAC/IEEE DAC*, June 2012, pp. 1024–1030.

[10] A. E. Engin and S. R. Narasimhan, "Modeling of crosstalk in through silicon vias," *IEEE Trans. on Electromagnetic Compatibility*, vol. PP, no. 99, pp. 1 –10, 2012.

[11] C. Liu, T. Song, J. Cho, J. Kim, J. Kim, and S.-K. Lim, "Full-chip tsv-to-tsv coupling analysis and optimization in 3d ic," in *48th ACM/EDAC/IEEE DAC*,, 2011, pp. 783–788.

[12] V. Pasca, L. Anghel, and M. Benabdenbi, "Configurable thru-silicon-via interconnect built-in self-test and diagnosis," in *Test Workshop (LATW), 2011 12th Latin American*, 2011, pp. 1–6.

[13] Y.-J. Huang, J.-F. Li, and C.-W. Chou, "Post-bond test techniques for tsvs with crosstalk faults in 3d ics," in *VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on*, 2012, pp. 1–4.

[14] L. Chen, X. Bai, and S. Dey, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," in *Design Automation Conference*, 2001, pp. 317–322.

[15] S. Kannan, B. C. Kim, and B. Ahn, "Fault modeling and multi-tone dither scheme for testing 3d tsv defects," *J. Electronic Testing*.

[16] C.-W. Kuo and H.-Y. Tsai, "Thermal stress analysis and failure mechanisms for through silicon via array," in *ITherm*, June 2012, pp. 202–206.

[17] X. Liu, Q. Chen, P. Dixit, R. Chatterjee, R. Tummala, and S. Sitaraman, "Failure mechanisms and optimum design for electroplated copper through-silicon vias (tsv)," in *ECTC*, May 2009, pp. 624 –629.

[18] M. Jung, X. Liu, S. K. Sitaraman, D. Z. Pan, and S. K. Lim, "Full-chip through-silicon-via interfacial crack analysis and optimization for 3d ic," in *ICCAD*.

[19] (2013, July). [Online]. Available: http://www.sematech.org/meetings/archives/3d/10124/pres/Beyne.pdf

[20] E. Marinissen, "Testing tsv-based three-dimensional stacked ics," in *DATE*, march 2010, pp. 1689 –1694.

[21] D. H. Jung, J. Kim, H. Kim, J. J. Kim, J. Kim, and J. S. Pak, "Disconnection failure model and analysis of tsv-based 3d ics," in *EDAPS*, Dec. 2012.

[22] M. Cuviello, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," in *ICCAD*, 1999, pp. 297–303.

[23] P. Goel and M. T. McMahon, "Electronic chip-in-place test," in *DAC*, June 1982, pp. 482–488.

[24] P. T. Wagner, "Interconnect testing with boundary scan," in *International Test Conference (ITC '87)*, Sep. 1987, pp. 52–57.

[25] R. Weerasekera, M. Grange, D. Pamunuwa, H. Tenhunen, and L.-R. Zheng, "Compact modelling of through-silicon vias (tsvs) in three-dimensional (3-d) integrated circuits," in *3D-IC*, 2009, pp. 1–8.

[26] (2013) Winmips64. [Online]. Available: http://indigo.ie/ mscott/

[27] (2013) Mips64 architecture for programmers volume ii: The mips64 instruction set. [Online]. Available: http://scc.ustc.edu.cn/zlsc/lxwycj/200910/W020100308600769158777.pdf

[28] M. Tehranipour, N. Ahmed, and M. Nourani, "Testing soc interconnects for signal integrity using boundary scan," in *VLSI Test Symposium*, 2003, pp. 158–163.

[29] S. Park and T. Kim, "A new ieee 1149.1 boundary scan design for the detection of delay defects," in *DATE*.

[30] N.K.Jha and S. Gupta, *Testing of Digital Systems*. Cambridge,United Kingdom: Cambridge University Press, 2003.

List of Publications

International Journals

1. **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Impact on the Overall Die-to-Wafer 3D Stacked IC Cost," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 1, pp. 15-25, Feb. 2012.

2. **M. Taouil** and S. Hamdioui, "Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 28, no. 4, pp. 523-534, Aug. 2012.

3. **M. Taouil**, S. Hamdioui and E.J. Marinissen, "Yield Improvement for 3D Wafer-to-Wafer Stacked ICs Using Wafer Matching," *submitted to ACM Transactions on Design Automation of Electronic Systems (TODAES), pp. 1–24, 2014*.

4. **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Post-Bond Interconnect Test and Diagnosis for 3D Memory Stacked on Logic," *submitted to IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 1–12, 2014*.

International Conferences

1. **M. Taouil**, S. Hamdioui, J. Verbree, and E.J. Marinissen, "On Maximizing the Compound Yield for 3D Wafer-to-Wafer Stacked ICs," in *International Test Conference (ITC)*, Austin, TX, USA, Nov. 2010, pp. 1-10.

2. **M. Taouil**, S. Hamdioui, K. Beenakker, and E.J. Marinissen, "Test Cost Analysis for 3D Die-to-Wafer Stacking," *19th IEEE Asian Test Symposium (ATS)*, Shanghai, China, Dec. 2010, pp. 435–441.

3. **M. Taouil** and S. Hamdioui, "Stacking Order Impact on Overall 3D Die-to-Wafer Stacked-IC Cost," *14th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Cottbus, Germany, April 2011, pp. 335–340.

4. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "How Significant will be the Test Cost Share for 3D Die-to-Wafer Stacked-ICs?" *6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Athens, Greece, April 2011, pp. 1–6.

5. **M. Taouil** and S. Hamdioui, "Layer Redundancy Based Yield Improvement for 3D Wafer-to-Wafer Stacked Memories," *European Test Symposium (ETS)*, Trondheim, Norway, May 2011, pp. 45–50.

6. S. Hamdioui and **M. Taouil**, "Yield Improvement and Test Cost Optimization for 3D Stacked ICs", *20th Asian Test Symposium (ATS)*, New Delhi, India, Nov. 2011, pp. 480–485. *(invited paper)*

7. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "On Modeling and Optimizing Cost in 3D Stacked-ICs," *6th IEEE International Design and Test Workshop (IDT)*, Beirut, Lebanon, Dec. 2011, pp. 24–29.

8. **M. Taouil** and S. Hamdioui, "On Optimizing Test Cost for Wafer-to-Wafer 3D-Stacked ICs," *7th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Tunis, Tunisia, May 2012, pp. 1–6.

9. M. Lefter, G.R. Voicu, **M. Taouil**, M. Enachescu, S. Hamdioui, and S.D. Cotofana, "Is TSV-based 3D Integration Suitable for Inter-die Memory Repair?" *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, March 2013, pp. 1251-1254.

10. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Using 3D-COSTAR for 2.5D Test Cost Optimization," *IEEE International 3D Systems Integration Conference (3DIC)*, San Fransisco, CA, USA, Oct. 2013, pp. 1–8.

11. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Impact of Mid-Bond Testing in 3D Stacked ICs," *16th IEEE Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, New York, NY, USA, Oct. 2013, pp. 178–183.

12. **M. Taouil**, M. Lefter, and S. Hamdioui, "Exploring Test Opportunities for Memory and Interconnects in 3D ICs," *International Design and Test Symposium (IDT)*, Marrakesh, Morocco, Dec. 2013, pp. 1–6.

13. **M. Taouil**, M. Masadeh, S. Hamdioui, and E.J. Marinissen, "Interconnect Test for 3D Stacked Memory-on-Logic,"*Design, Automation & Test in Europe (DATE)*, Dresden, Germany, March 2014, pp. 1–6.

14. **M.Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "Quality versus Cost Analysis for 3D Stacked ICs," *32nd IEEE VLSI Test Symposium (VTS)*, Napa, CA, USA, April 2014, pp. 1–6.

15. E.J. Marinissen, B. de Wachter, K. Smith, J. Kiesewetter, **M.Taouil**, and S. Hamdioui, "Direct Probing on Large-Array Fine-Pitch Micro-Bumps of a Wide-I/O Logic-Memory Interface," *International Test Conference (ITC)*, Seattle, WA, Oct. 2014, pp. 1–10.

International Workshops

1. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "Impact of Test Flows on the Cost in 3D Die-to-Wafer Stacking," *First IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-Test)*, Austin, TX, USA, Nov. 2010.

2. **M. Taouil**, S. Hamdioui, and E.J. Marinissen, "Test Cost Modeling for 3D-Stacked ICs," *Second IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-Test)*, Anaheim, CA, USA, Sept. 2011.

3. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "3D-COSTAR: A Cost Model For 3D Stacked ICs," *Third IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits (3D-Test)*, Anaheim, CA, USA, Nov. 2012.

4. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "3D-COSTAR: A Cost Model for 3D Stacked ICs," *3D Workshop in conjunction Design Automation & Test in Europe (DATE)*, Grenoble, France, March 2013.

5. **M. Taouil**, S. Hamdioui, E.J. Marinissen, and S. Bhawmik, "2.5D Test Cost Optimization using 3D-COSTAR," *3D Workshop in conjunction Design Automation & Test in Europe (DATE)*, Dresden, Germany, March 2014.

Other Publications

1. G.K. Kuzmanov, **M. Taouil**, "Reconfigurable Sparse/Dense Matrix-Vector Multiplier," *International Conference on Field-Programmable Technology (FPT)*, Sidney, Australia, Dec. 2009, pp. 483–488.

2. C. van der Bok, **M. Taouil**, P. Afratis, I. Sourdis, "The TU Delft Sudoku Solver on FPGA," *International Conference on Field-Programmable Technology (FPT)*, Sidney, Australia, Dec. 2009, pp. 526–529. **2nd prize FPT-2009 Design Competition Award**

3. L. Hasan, Z. Al-Ars, **M. Taouil**, “High Performance and Resource Efficient Biological Sequence Alignment”, 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Sept. 2010, pp. 1767–1770.

4. L. Hasan, Z. Al-Ars, **M. Taouil**, K.L.M. Bertels, “Performance and Bandwidth Optimization for Biological Sequence Alignment”, 5th IEEE International Design & Test Workshop (IDT), Abu Dhabi, UAE, Dec. 2010, pp. 155–160.

5. M.S. Khan, S. Hamdioui, **M. Taouil**, H. Kukner, P. Raghavan, F. Catthoor, “Impact of Partial Resistive Defects and Bias Temperature Instability on SRAM Decoder Reliablity”, *International Design & Test Symposium (IDT)*, Doha, Qatar, Dec. 2012, pp. 1–6.

6. S. Hamdioui, **M. Taouil**, N.Z.B. Haron, “Testing Open Defects in Memristor-Based Memories”, *IEEE Transactions on Computers (TC)*, vol. 99, pp. 1–14, 2013.

Curriculum Vitae

Mottaqiallah Taouil was born 1985 in Al Hoceima, Morocco. He attended secondary education at Veurs College te Leidschendam, where he obtained a diploma in 2003. Furthermore, he received the B.Sc. diploma in electrical engineering in 2007 from Delft University of Technology, the Netherlands. Thereafter, he completed in 2009 his M.Sc. degree (with cum Laude) at Computer Engineering laboratory from the same institution. Then, in November 2009, he joined the Department of Software and Computer Technology at the Faculty of Electrical Engineering, Mathematics, Computer Science at Delft University of Technology, the Netherlands, to pursue the Ph.D. degree under the supervision of dr. ir. Said Hamdioui. His research interests include Reconfigurable Computing, Embedded Systems, VLSI design & test, reliability, 3D stacked ICs, 3D architectures, design for testability, yield analysis and memory testing.