Yield Improvement and Test Cost Optimization for 3D Stacked ICs

Said Hamdioui  Mottaqiallah Taouil

Abstract—Three-Dimensional Stacked IC (3D-SIC) is an emerging technology promising many benefits, such as heterogeneous integration, reduced latency and power consumption. Realizing higher compound yield and overall low cost are the driving forces of the success of such a technology. This paper addresses these two topics.

First, two yield improvement schemes will be discussed: wafer matching and layer redundancy. Wafer matching is a technique that can be applied when Wafer-to-Wafer (W2W) stacking is used to fabricate 3D-SICs; this stacking approach provides many advantages such as high throughput, thin wafer and small die handling, and high TSV density; however, it suffers from low compound yield as compared with other stacking processes. Layer redundancy, on the other hand, is based on adding redundant layer(s) to the stacked IC to replace the faulty irreparable dies in the stack. It can be applied only when similar dies are stacked as it is the case for stacked memories. Experiment results for both wafer matching and layer redundancy will be presented and compared; they show that both wafer matching and layer redundancy significantly improve the yield and therefore reduce the cost per 3D-SIC.

Second, test cost optimization will be covered. During the manufacturing of 3D-SICs, tests can be applied at different moments such as before the stacking process, during the creation of each partially stacked IC, after the creation of the complete stack, etc. This results into a huge number of test flows. A framework covering different test flows will be discussed. In addition, an appropriate cost model able to identify the most cost-effective test flow will be presented. The simulation results show that test flows with the pre-bond testing significantly reduce the overall cost, that a cheaper test flow does not necessary results in lower overall cost, and that the best cost-effective test flow strongly depends on the stack yield; hence, adapting the test according the stack yield is the best approach to use.

Index Terms—3D Stacked IC, Yield, Layer Redundancy, Test Flows

I. INTRODUCTION

3D-SIC technology is emerging as one of the main competitors to continue the trend of Moore’s Law [1–4]. Perhaps the most reliable way to achieve this is using 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, higher bandwidth communication (as TSVs cross dies along the surface of the chip), heterogeneous integration, etc.

Stacking dies comes with three flavors [4]: Wafer-to-Wafer (W2W), Die-to-Wafer (D2W) or Die-to-Die (D2D) stacking [3]; each with its own advantages and disadvantages. In D2D stacking, stacking is based on individual selected dies. The 3D-SIC yield can get high as Known-Good-dies (KGD) can be identified. Drawback of this approach is the low manufacturing throughput as each die has to be handled separately. In D2W stacking, this throughput is increased as stacking occurs on wafers. Yet, the freedom still exists to stack KGDs only. Finally, in W2W stacking complete wafers are stacked and bonded together. Therefore, it can handle cases with extremely small dies. W2W stacking requires the stacked dies to be of equal size. Stacked memories [2,5] are examples of 3D-SICs that could benefit from W2W stacking. Although W2W stacking results in highest throughput, it suffers from lower compound yield. Moreover, during the manufacturing of 3D-ICs, tests can be applied at different moments; for instance, when D2W stacking is used, a test can be applied (a) before the stacking process, (b) during the creation of each partially stacked IC, (c) after the creation of the complete stack, and (d) after packaging of the stack; each applied test may target interconnects, one or more dies, or both. This results into a huge number of test flows, each with a 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. In conclusion, yield improvement and test cost optimization are the drivers for overall 3D-SIC cost reduction.

Two schemes have been developed for 3D-SIC yield improvement: wafer matching and layer redundancy. In wafer matching, a software algorithm is used to identify wafers with equal or similar fault maps. Subsequently, these are stacked. Wafer matching was initially introduced by Smith et al. [6], where the authors compared the yield improvement of a single die SoC by mapping it into a 3D-SIC with two equal sized layers. More elaborated studies of wafer matching are presented in [7,8]. However, all this work considered wafer matching with static repositories, i.e., after wafer selection the repositories are not replenished unless they are empty. In our previous work [9], we considered running repositories and...
showed their superiority over static repositories. The second yield improvement technique, layer redundancy, is based on extending the concept of (memory) redundancy in the third dimension; the latter gives not only the option to use spare resources within a die (intra-layer), but also to use resources of neighbor dies (inter-layer) redundancy or to cover up for complete dies (layer redundancy). In [10,11], inter-layer redundancy is used by the authors to increase the stacked memory yield for different allocation algorithms. In our previous work [12], we analyzed layer-redundancy and proposed a cost improvement model to evaluate the scheme. Note that wafer matching is applicable only in case of W2W stacking while layer redundancy can be applied for any stacking process as long as the stacked die are similar.

Optimizing test cost is another challenge that can significantly contribute to the overall cost reduction. Choosing an optimal and efficient test flow requires the analysis for such flows using an appropriate test cost model. Research on this topic is still in its infancy stage and very limited work is published [13–15]. In [13], 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 [14], 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 [15], 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 [16], 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 [17].

This paper discusses yield improvement and test cost optimization for 3D-SICs. It overviews wafer matching and layer redundancy and shows by means of simulation how these schemes can significantly improve the overall yield. It further presents an analysis of different test flows and shows, based on simulation performed using appropriate cost model, how different test flows can result into different cost and that the cheapest test flow does not necessary results in lower overall 3D-SIC cost. The rest of the paper is organized as follows. Section II and Section III present wafer matching and layer redundancy, respectively, and their impact on yield improvement. Section IV discusses the possible test flows for 3D-SICs and their impact both on test cost and overall cost. Finally, Section V concludes the paper.

II. WAFER MATCHING FOR YIELD IMPROVEMENT

In this section we analyze the yield improvement due to wafer matching. Section II-A briefly describes the wafer matching problem. Section II-B discusses the different aspects that contribute to the definition of our wafer matching framework. Finally, Section II-C shows some simulation results.

A. Problem description

The problem of W2W 3D-SICs can be defined as follows (see Figure 1): Given, (a) number of repositories each with wafers, (b) fault maps for all the wafers (based on pre-bond testing), and (c) a production size of m 3D-SICs, maximize the overall compound yield for all m 3D-SICs, by selecting appropriate wafers for the n-layer 3D-SICs from the repositories [9]. Note that 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.

B. Matching framework

The wafer matching framework defines the possible scenarios that can be used to realize the wafer matching either for static or running repositories; each scenario is a combination of a matching process and a matching criterion; these are explained next.

1) Matching Process: It defines a step-by-step process for the realization of wafer matching. Therefore, it determines the number of repositories and the number of wafers that are considered at a time. Referring back to Figure 1, and depending on the number of involved repositories, 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. Hence, this is an iterative process.
- **All-Layers (AL)**: all repositories are used at once for the matching process. 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 matching step, two cases can be distinguished:

- **Wafers-Wafer (WbW)**: In each step of the wafer matching process, only one wafer from each repository is
involved (with no regard to the remaining wafers) and wafers contributing to the best match are selected. This process is regarded as greedy.

- **All-Wafers (AW):** all wafers from all involved repositories are matched in each step of the wafer matching process. As the process considers all possible outcomes for all $k$ wafers, this process is considered to be exhaustive.

The above combinations can be compiled into five possible wafer matching processes [9]: (1) LbL;WbW, (2) WbW;LbL, (3) LbL;AW, (4) AL;WbW and (5) AL;AW.

2) **Matching Criterion:** When performing wafer matching, one intends to maximize the matching good/faulty dies, and/or minimize the matching between good and bad dies. These can be compiled into three criterion:

- **Max(MG).** The best wafer pair combinations is selected 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).** The best wafer pair combinations is selected 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.

Table I gives an the matching framework; it shows all possible combinations of matching processes and repository types; a valid combination can be used with any matching criteria, resulting into a matching scenarios; matching scenarios already addressed in the literature are represented by their references.

**C. Simulation Results**

In order to show the added value of wafer matching and superiority of running repository over static repositories, we will consider the simulation of ‘Best Pair (BP)’ based WbW;LbL scenario running repository [9] and compare it with the state-of-the art. In the BP-based matching process, the wafers from the first two repositories are matched in pairs without any selection restrictions. 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. All matching criteria are considered.

The experiments are based on the reference process in [8]; e.g., a defect density is considered to be $d_0 = 0.5 \text{ defects/cm}^2$ and the defect clustering parameter $\alpha = 0.5$, etc.

The simulation results are summarized in Table II for $n = 3$ and $k = 25$ with 590 dies per wafer. In the table, the first column provides the varied parameter (wafer yield) of the simulation; the second column reports the compound yield of the related work; the third column presents the compound yield of 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. It should be noted that the yield value given in the second and the third column are the max value obtained when the three matching criteria were used. From the table, we can conclude that running repositories lead to a higher compound yield and that the lower the wafer yield, the higher yield improvement.

**III. LAYER REDUNDANCY FOR YIELD IMPROVEMENT**

We will considerer 3D stacked memories as an example to show the added value of layer redundancy for yield improvement. First, a classification of 3D memory redundancy scheme is provided. Second, a model for layer redundancy and its associated cost are presented. Finally, some simulation results are discussed.

**A. Classification of 3D redundancy**

Traditionally, yield improvement for 2D memories is based on the use of spare rows and/or columns [18]. 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 (i.e., spare rows and/or columns) within each layer (similar to that in planar memories) 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 from the die they belong to only, but also from neighbor dies as it is the case for Tezzaron memories [5]. In [10], inter-layer redundancy is used to increase the stacked memory yield for different allocation algorithms for D2W stacking.

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 in it.

In this paper, we present the yield improvement due to layer redundancy [12].
B. Redundancy Based Yield Model and Its Cost

Let assume a 3D memory with a stack size \( n \), and a number of redundant layers \( r \); the complete stack size will be \( s = n + r \). Lets denote the overall compound yield of the stack with a size \( n \) with \( Y(n) \). Further, assume that the stacking yield is composed of two parameters: the interconnect (TSV) yield \( Y_{INT} \) and the stacked-die yield \( Y_{SD} \). The latter present the fraction of good dies that enter the stack that do not get corrupted during stacking. Given these parameter, a yield model will be given both for memories without and 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.

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

(1)

Note that a 3D stacked memory with \( n \) layers requires \( n-1 \) stacking steps.

b) Memories with layer redundancy: In this case, if \( n \) or more layers out of the stacked \( s = n + r \) 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:

\[
p(i) = \binom{s}{i} \cdot Y_D^i \cdot (1 - Y_D)^{s-i}
\]  

(2)

We extend the symbol \( Y(n) \) for non-redundant memories to \( Y(n, s) \) in order to express the yield of an \( s \) layered stack with \( r = s - n \) redundant layers.

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

(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.

In order to evaluate either it is cost-wise justified to use layer redundancy, the associated increased cost has to be taken into consideration. The manufacturing costs \( C_m(s) \) for a stack size of \( x \) \((x = n \) and \( x = s \)) for non-redundant, respectively, redundant memories can be formulated by:

\[
C_m(x) = s \cdot C_w + (x - 1) \cdot C_{3D}
\]  

(4)

where \( C_w \) the wafer cost and \( C_{3D} \) the cost related to 3D stacking processes including TSV, back side processing, bonding processing, etc.

C. Experimental Results

In order to appropriately evaluate the added value of layer redundancy, 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. 5 and Eq. 6 respectively.

\[
C_{GD}(n) = C_m(n)/Y(n)
\]  

(5)

\[
C_{GD}(n, s) = C_m(s)/Y(n, s)
\]  

(6)

By using the above equations, the relative improvement of the price of a good 3D-SIC with layer redundancy over the one without layer redundancy can be calculated:

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

(7)

Here, Eq. 4 is substituted for \( C_m(s) \) and \( C_m(n) \).

The simulation has been performed for various values of \( n \) and \( s \); we further assume the following values for the parameters involved in the equations [12]: die yield \( Y_D = 85\% \), stacked-die yield \( Y_{SD} = 99\% \), and interconnect yield \( Y_{INT} = 97\% \). Figure 2 shows the results for various values of \( n \) and \( s \). The column on the right side shows the yield improvement \( (Y(n, s)/Y(n)) \) in \%. The left part shows the cost ratio for \( 0.1 \leq \frac{C_{GD}(n, s)}{C_{GD}(n)} \leq 0.9 \), i.e., the 3D processing cost lies between 10\% and 90\% of the wafer cost. The following can be concluded from the figure:

- Layer redundancy improves the memory yield irrespectively for all considered cases. E.g., an improvement of 39.42\% is obtained in case of \( r = 1 \) and \( n = 3 \).
- The impact of the ratio \( \frac{C_{GD}(n, s)}{C_{GD}(n)} \) 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 made. The figure clearly shows that for the parameters considered in this case, the achieved yield improvement using layer redundancy results in lower cost per good stack.

Moreover, a comparison of layer redundancy and wafer matching reveals that for \( n \geq 3 \) the former leads to better results (up to 4X) [12]. E.g., for \( n = 6 \), layer redundancy reduces the cost with 38.45\% and wafer matching with only 10.27\%; see [12] for more experimental results and analysis.
IV. TEST COST OPTIMIZATION

This section will show how using appropriate test flow can significantly contribute to test cost reduction. First, test moments and test flows are presented. Thereafter, a cost model allowing the evaluation of test flows is presented. Finally, simulation results are covered.

A. Test moments and test flows

Figure 3(a) shows the conventional 2D test flow for planar wafers [19]; 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, 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 moments can be defined for partial stacks. At each moment a distinction can be made between die tests and interconnect tests. For 3D-SICs, four test moments can be distinguished in time as depicted in Figure 3(b).

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 [4]) as well.

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.

3) $T_{po}$: one post-bond test. This test can be applied after the complete stack is formed. $T_{pr}$ can be applied to save unnecessary assembly and packaging costs.

4) $T_{f}$: one final test can be applied after assembly and packaging to ensure the required product quality.

The test flow framework can be extracted from the above test moments, which consist in total of $2n$ tests. 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 $2^n (T_{pr}) \times 4^{n-2} (T_{mi}) \times 2 (T_{po}) \times 2 (T_{f}) = 2^{3n-2}$. 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) [16]. For example, if we restrict mid-bond testing to be one of the following:

- Test for the interconnect between the top dies ($i_t= \text{top interconnect}$) only.
- Test for the top dies ($d= \text{top dies}$) only.
- Test for both the top interconnect and top dies ($i_t, d$), or
- None ($n$).

Then, $T_{mi}$ will be reduced to $T_{mi} \in \{i_t, d, i_t d, n\}$.

In [16], reasonable assumptions were made such that the test flows were reduced to eight for D2W stacking. They are given in Table III; e.g., TF1 denotes a test flow based on no pre-bond test ($T_{pr}$), no mid-bond ($T_{mi}$) and $T_{po} = i_t d$, where $i_t$ denotes a test for all interconnect and $d$ 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) $T_{mi} \in \{i_t, d, i_t d, n\}$ and (c) $T_{po}$ can be any of $\{i_t, d, i_t d, n\}$ 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 [16].

B. Cost model

To evaluate the impact different test flows on the overall cost and identify the optimal one, an appropriate cost model has to be built. Therefore, the test cost of each flow should be identified. However, this is by far not enough to produce a fair comparison. Other cost classes have to be considered as well. For example, a pre-bond TSV test requires additional hardware and prevents faulty dies to enter in the stack; this increases the chip area resulting in less dies per wafer. Hence, it impact the overall cost of 3D-SIC.

In our work [16,17] we proposed an appropriate cost model; its block diagram is shown in Figure 4. The model considers five classes of input parameters:

- Design: defines the cost related to the design of the IC.
- Manufacturing: defines parameters/cost 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.
- Packaging: The cost of 3D-SIC packaging.
- Test: defines the cost related to DFT and test/ test flows such as that of testing dies and interconnects. Test flows have a large impact on this cost since they determine when and what to test for.
- Logistics: The cost related to move dies or wafers between fabs (e.g., between wafer fab and 3D stacking fab); this cost is different for fab-less, fab-lite and IDM’s.
C. Simulation results

In this section we briefly review some experimental results of simulating the test flows of Table III, using the cost model depicted in Figure 4; only the manufacturing, packaging and test inputs are considered. We will present the impact of test flows on the overall 3D-SICs for different stack yields; other experimental results (e.g., for different stack size, die yield, etc.) can be found in [12]. Nevertheless, both results show similar trends. Note that the parameters used for the simulation are the same as those described in Section II-C.

Figure 5 depicts the overall 3D cost versus stacked yield (i.e., interconnect yield $Y_{INT}$ and stacked-die yield $Y_{SD}$) 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.

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.

V. Conclusions

This paper discusses yield improvement and test cost reduction for 3D-SICs; both significantly contribute to the reduction of overall 3D-SIC cost. Two yield improvement schemes for 2W2 stacked 3D-SICs were covered: wafer matching and layer redundancy. Experimental results show that wafer matching improves the yield with up to 59% as compared with random stacking for lower wafer yield, and that layer redundancy even outperforms wafer matching. Moreover, different test moments and test flows were defined and analyzed for test cost reduction using an appropriate cost model. The results show that test flows with pre-bond testing are by far much better, especially for wafers/dies with lower yield.

REFERENCES