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

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

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

(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

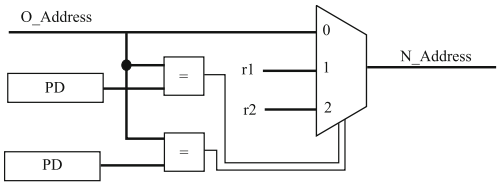

Fig. 1 Layer replacement circuit

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

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 $A = 93$ 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; entries 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 $0.1 \leq \frac{C_{3\mathrm{D}}}{C_{\mathrm{w}}} \leq 0.9$, 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 $\frac{C_{3\mathrm{D}}}{C_{\mathrm{w}}}$ 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.

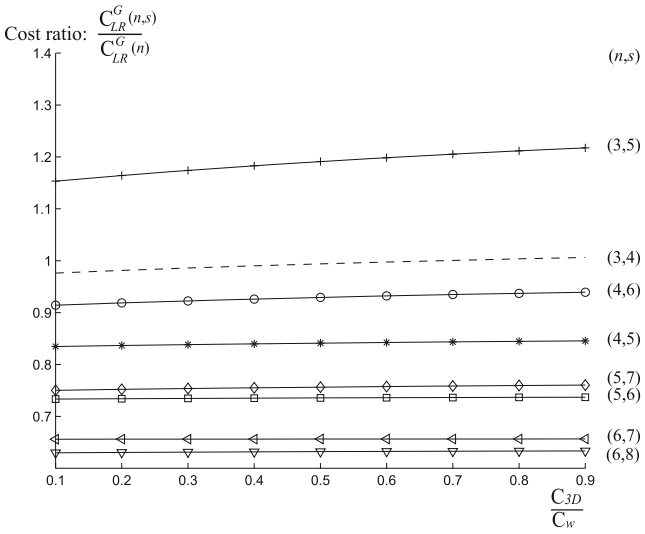

Fig. 2 Impact of layer redundancy on the cost ratio $\frac{C^G_{\mathrm{LR}}(n,s)}{C^G(n)}$

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 $\frac{C_{3\mathrm{D}}}{C_{\mathrm{w}}} = 0.3$. 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

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	$\frac{Y_{WM}(n)-Y(n)}{Y(n)}$ (%)	$\frac{Y_{LR}(n,s)-Y(n)}{Y(n)}$ (%)	r	$\frac{C^G_{WM}(n)-C^G(n)}{C^G(n)}$ (%)	$\frac{C^G_{LR}(n,s)-C^G(n)}{C^G(n)}$ (%)	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 $Y_{M,BP}$ 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 $C_{COM}(s)$ 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.

$$C_{COM}(s) = C_{COM,m}(s) + C_{COM,t}(s) + C_{COM,p}(s) \quad (32)$$

$$C_{COM,m}(s) = s \cdot C_w + (s-1) \cdot C_{3D} \quad (33)$$

$$C_{COM,t}(s) = C_{COM,t,pre}(s) + C_{COM,t,post}(s) + C_{COM,t,final}(s) \quad (34)$$

$$C_{COM,t,pre}(s) = s \cdot d \cdot t_{int} \quad (35)$$

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 VLSl 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.