On Effective TSV Repair for 3D-Stacked ICs

Li Jiang[†‡], Qiang Xu[†‡], and Bill Eklow[§]

[†]CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[§]Cisco Systems, San Jose, CA
Email: {ljiang,qxu}@cse.cuhk.edu.hk; beklow@cisco.com

Abstract—**3D-stacked ICs that employ through-silicon vias (TSVs) to connect multiple dies vertically have gained wide-spread interest in the semiconductor industry. In order to be commercially viable, the assembly yield for 3D-stacked ICs must be as high as possible, requiring TSVs to be reparable. Existing techniques typically assume TSV faults to be uniformly distributed and use neighboring TSVs to repair faulty ones, if any. In practice, however, clustered TSV faults are quite common due to the fact that the TSV bonding quality depends on surface roughness and cleaness of silicon dies, rendering prior TSV redundancy solutions less effective. To resolve this problem, we present a novel TSV repair framework, including a hardware architecture that enables faulty TSVs to be repaired by redundant TSVs that are farther apart, and the corresponding repair algorithm. By doing so, the manufacturing yield for 3D-stacked ICs can be dramatically improved, as demonstrated in our experimental results.**

I. Introduction

Three-dimensional (3D) technology that integrates multiple silicon dies with short and dense through-silicon vias (TSVs), which provide abundant interconnect bandwidth with improved performance and less communication energy, has gained great interest in the semiconductor industry. Early 3D-stacked IC (3D-SIC) products for CMOS image sensor camera modules are already in volume production [1, 2]. 3D-stacked memory products were also announced by various companies recently [3–5]. Moreover, various techniques with massive use of TSVs have been proposed to fully exploit the benefits of this emerging technology(e.g., [6–8]).

While 3D-SICs provide many benefits over traditional 2D ICs and have been produced in certain applications, they can be widely adopted only when their design and manufacturing costs are commercially viable [9]. Among the various factors that affect 3D-SIC product cost, manufacturing yield is one of the most (if not the most) crucial factors [10].

Generally speaking, there are two types of yield loss in 3D-SICs.

- *Stack yield loss* caused by defects in one or more of the stacked dies;
- *Assembly yield loss* caused by defects occurred during the assembling process;

Various yield enhancement techniques for 3D-SICs have been proposed, as surveyed in [11]. To improve the stack yield of 3D-SICs, it is critical to conduct pre-bond testing to screen out defective dies so that only known good dies (KGDs) are used to form the stacked ICs [12]. In addition, several die/wafer matching and inter-die repair strategies have also been proposed in the literature to enhance stack yield (e.g., [13–15]).

The assembly process for 3D-SICs involves many challenging manufacturing steps (e.g., wafer thinning and TSV bonding), which may cause various types of TSV faults [16]. Adding redundant TSVs to repair faulty ones is probably the most effective method to enhance assembly yield besides improving the manufacturing process itself. Several TSV redundancy design techniques have been proposed in the literature [3, 17, 18]. Despite different redundancy allocation strategies used in these works, they all assume uniformly-distributed TSV faults and use neighboring TSVs to replace faulty ones, if any. In practice, however, the bonding quality of TSVs depends not only on the bonding technology, but also the winding level of the thinned wafer and the surface roughness and cleaness of silicon dies. Consequently, if one TSV is defective during the bonding process, it is more likely that its neighboring TSVs are also faulty. Due to such clustering effect, earlier TSV repair techniques are less effective because a signal TSV and its neighboring redundant TSV may be defective at the same time.

In this paper, we propose a novel TSV repair framework to tackle the above problem. Instead of repairing faulty TSVs by their neighbor TSVs, our technique enables them to be repaired by redundant TSVs that are "distant". With the improved repair flexibility, our technique is suitable to repair clustered, faulty TSVs. To guarantee the timing correctness after repair, we also present a new repair algorithm in this work. Experimental results show that the proposed solution outperforms prior techniques, especially when the number of TSVs used in the 3D-SICs is large and/or the clustering effect is significant.

The remainder of this paper is as follows: Section II presents the preliminaries and motivation of this work. In Section III, we present the hardware architecture of our TSV repair framework. The corresponding repair algorithm is then shown in Section IV. Section V presents the experimental results for various hypothetical 3D-SICs. We then discuss several practical considerations to use the proposed technique in Section VI. Finally, we conclude this work in Section VII.

II. Preliminaries and Motivation

To date, there is no public data on actual TSV failure rates. In fact, they can vary significantly among different foundries because the failure rate of a particular TSV technology depends on its technology maturity level and parameters such as TSV width/height and TSV pitch size. The common belief is that: while the TSV processing technology has advanced significantly over the past several years, TSV yield is still not satisfactory, requiring redundancy for defect-tolerance. Consequently, several TSV redundancy allocation strategies were presented in the literature which differ in terms of redundancy ratio ($\frac{\#Redundant\ TSVs}{\#Signal\ TSVs}$), repair capability, and hardware cost.
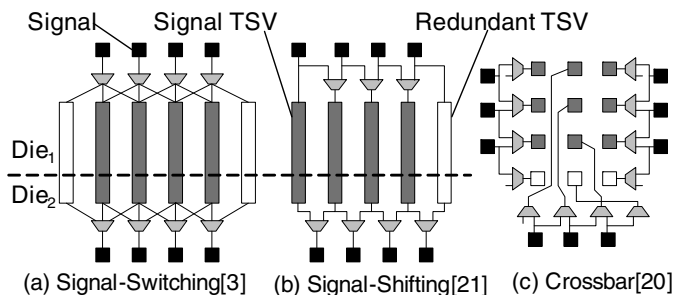
Fig. 1.   Existing TSV Redundancy Solutions

In [3], Samsung presented a TSV redundancy strategy used to improve the yield of its 3D memory product, wherein four signal TSVs and two spares are bundled together to form a group of six TSVs (see Fig. 1(a)). The redundancy ratio of this technique is 1:2 and it can tolerate any two TSV failures within the group.

Hsieh *et al.* [18] proposed to link signal TSVs in a TSV block with one spare TSV to form a TSV-chain (see Fig. 1(b)). If there is one defective TSV in a TSV block, signal shifting is conducted to repair it with the spare. Suppose each TSV block contains N TSVs, the redundancy ratio of this technique is $1 : N$, and it can tolerate one TSV failure in the block.

In *et al.* [17], for a $N \times N$ TSV grid used as NoC links, redundant rows or columns of TSVs are added for defect-tolerance. Suppose a redundant row is added (see Fig. 1(c)), each spare TSV is connected to the signal TSVs on its corresponding column through a crossbar and it can be used to repair any defective TSV on that column. Suppose M redundant columns/rows are added, the redundancy ratio of this technique is $M : N$, and it can tolerate any M TSV failure in each row/column in the grid.

While significant yield improvements were achieved in the above works, their analysis, in all cases, was based on the assumption that TSV defects are uniformly-distributed. This assumption may hold true for certain random defects such as void formation [19] and lamination due to thermal induced stress [20]. At the same time, however, many types of TSV defects appear during the imperfect bonding process. Oxidation or contamination of the bond surface, height variation of the TSVs, thinned dies warping [21] and bowing of a wafer can cause large alignment errors [22], leading to clustered, faulty TSVs.

Due to the above, it is likely that, while most signal TSV groups have very few faults and are reparable, there exists one or more signal TSV groups that are vulnerable to clustered faults and will become irreparable, if the redundancy ratio is large enough. On the other hand, since prior solutions mainly rely on neighboring spare TSVs for repair, they may suffer from the same clustered faults as the defective one, rendering low repair efficiency. Motivated by the above, in this work, we propose a novel repair framework that is effective to repair clustered TSV faults, as shown in the following sections.

III. PROPOSED TSV REDUNDANCY ARCHITECTURE

In order to handle clustered TSV fault, our solution is to offer more repair options for each defective TSV. In other words, we try to increase repair flexibility so that a defective TSV can be replaced by a spare that is distant.

A. Overall Structure

Inspired by the compensation path problems [23], the proposed architecture links the TSV pads by switches and wires, constructing
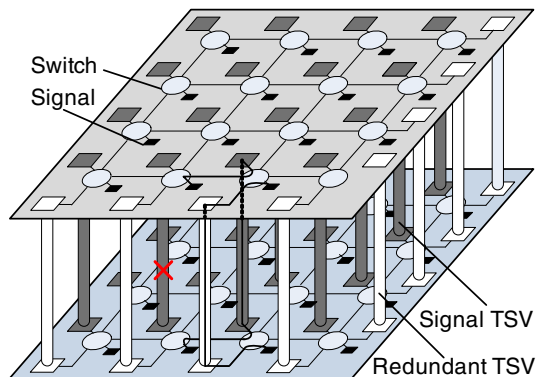

Fig. 2.   Proposed TSV Redundancy Architecture.

a TSV grid. Additionally, redundant TSVs are also linked to two borders of the grid (see Fig. 2). In this architecture, if one signal is disconnected due to a TSV fault (the one with "X" mark), the switches linking two pads of the faulty TSV reroute the signal through a neighbor fault-free TSV by way of their switches (see the solid line). Since the fault-free TSV is now attached to the previous rerouted signal, its original signal needs to be rerouted as well. This process continues until a redundant TSV on the border is used.

TSVs are usually fabricated in a "regular" manner and grouped as bundles in many 3D-SIC designs [5, 8], these regularly-placed TSVs can be naturally linked together to construct the proposed TSV redundancy architecture. In case that TSVs are not regularly placed, we can also map them into a logical TSV grid and apply our repair architecture (discussed later). Note that, while more hardware resources are needed in the proposed architecture (i.e., additional switches and wires) when compared to earlier TSV redundancy techniques, this hardware cost is well justified by the yield improvement brought with our solution, as shown in our experimental results.

B. Switch Design and Repair Path Routing

The switch design depends on the placement of redundant TSVs. Take Fig. 2 as an example, the redundant TSVs are placed on the east and south borders of the TSV grid. As a result, we constrain signals to only route from two directions (from west to east or from north to south). Fig. 3(a) shows the schematic of the corresponding switch design. The original signal and its corresponding TSV have two ports in the switch (denoted as *signal port* and *TSV port*). In addition, there are four other ports connecting other switches far apart from four different directions (denoted as *linking ports*). The design principle
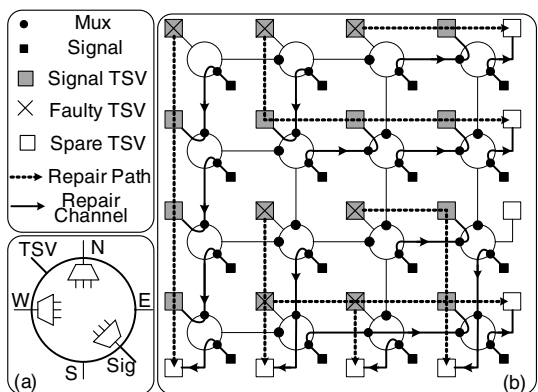

Fig. 3.   Switch Design and Routing Capability. (a)Switch Design; (b)TSV grid with Edge-Disjoint Repair Path.

is that the *signal port* and two *linking ports* (**N**orth and **W**est) have a mux capable of linking to the *TSV port* and the remaining *linking ports* (**E**ast and **S**outh).

We use Fig. 3(b) as an example (4×4 grid) to introduce the concept of *repair path* and to show its routing capability. Initially the mux of *signal port* connects to *TSV port*. Once the faulty TSV is detected, the signal port reroutes to another fault free TSV by reconfiguring the connectivity of the switch. This type of physical connection between signal and TSV is represented as a **repair channel** (see the solid arrow). Starting from any faulty TSV, there must be a succession of continuous *repair channels* finally terminating to a redundant TSV. We denoted this virtual connection from a faulty TSV to a redundant TSV as **repair path**. For example, the three clustered faults on the top find three disjointed *repair paths* to redundant TSVs (see dashed arrow). The four clustered faults on the bottom also find four disjointed *repair paths*. It is worth noting that the design of the switch guarantees that the *repair paths* can intersect with each other without any contradiction as long as port connections within the switch have no conflict (see the three clustered faults in the bottom).

IV. PROPOSED REPAIR ALGORITHM

After the fault maps of each TSV grid is obtained via testing [1], a repair algorithm is essential to analyze whether the TSV grid is reparable and generate *repair paths* for each faulty TSV, if possible.

A. Problem Analysis

Consider the TSV grid as a directed graph, wherein each vertex represents a TSV and its corresponding switch while the directed edge connecting two vertices is the wire between two switches (the edge direction depends on the constraint of signal routing directions). Our problem is to find *edge-disjoint repair paths* for all faulty TSVs, and we can employ the Maximum Flow method [25] to find them. To be specific, we first assign each edge in the graph with an unit capacity "1" to construct a directed flow graph. By adding a super source node that points to those faulty TSVs and merging all the spare TSVs into a target node, the TSV grid is repairable only if the weight of maximum flow is equal to the number of faulty TSVs (see Fig. 4(a)).

While the above problem formulation and its corresponding solution is simple and effective, it does not take the additional delay introduced during repair into consideration. To guarantee the timing correctness of the circuit after repair without necessarily changing our problem formulation completely, we translate the timing constraint for each "to-be-repaired" fault as length constraints in the flow graph. That is, each wire/mux in the TSV grid is associated with a length weight, and given a length constraint for each signal, the length of *repair channel* (the distance between the signal with the faulty TSV and its reconnected TSV) cannot violate the length constraint.

With the above, directly using maximum flow method to solve our problem is not applicable because: (i). the "flow" in maximum flow method has no sense of length; (ii). if two *repair paths* intersect in the same TSV (e.g. dashed path and doted path intersected in TSV V in Fig. 4(a)), a decision has to be made, that one *repair path* should possess the TSV while the other one is bypassed, and this decision has to consider the timing constraint. This is not a concern in the original maximum flow method. Before introducing our repair algorithm in detail, we prove this problem is a NP problem first, as shown in the following.

[1]Testing is out of the scope of this paper. Interested readers may refer to [18, 21, 24] for more details.
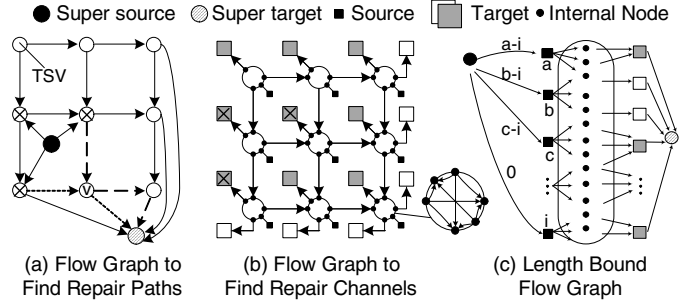

Fig. 4. Problem Transformation.

First, we transform the flow graph by setting all signals as sources and all the fault-free TSVs as targets (see Fig. 4(b)). At the same time, all the ports within switches become internal vertices while all the wires between them are edges in the graph. The edge between TSV and the *TSV port* guarantees that each TSV is used only once, and the length weight is put on these edges. The problem now becomes how to find edge-disjoint paths from all signals to TSVs under the length constraint. Second, we rearrange the flow graph as shown in Fig. 4(c), where each source is labeled with length constraint ($a...f$), among which we suppose f is the minimum length constraint. Then we add a super source/target and links to all the sources/targets. For each link from super source to source, we manipulate a length weight that is the source's length constraint minus the minimum length constraint (e.g., $a - f$). Thus, the original problem becomes a NP *maximum length-bounded flow problem*, which is to find a maximum flow between one source and one target where the length of all flows are bounded by a length constraint [26].

B. Repair with Length Constraint

For the sake of simplicity, in this work, we assume that there is a unified length constraint C, i.e., the one for the most critical signal.

Our heuristic is shown in Fig. 5, and the basic idea is to divide sources into groups and apply bounded search for each group. First, we initialize the flow graph by removing those edges that link signals with faulty TSVs and their *TSV port* (line 1). Consider that the *repair path* can only go towards east and south directions. It is better to conduct a bounded search for those sources from west/north to east/south. This is because- the targets and edges chosen for preceding sources are no longer available for the latter sources, thus reducing solution space. At the same time, the sources in the same diagonal lines have no edge connection to each other, which makes it a perfect choice to group them together (line 2). For each source in a group, we first find all possible candidate *repair channels* (c_j) including available edges and targets that satisfie the length constraint using breadth first search (lines 4-5). To avoid an extremely large solution space during exhaustive search, we constrain the candidate *repair channels* in terms of their number of edges. The search bound iterative increases from "1" as long as it does not exceed some pre-defined maximum bound (line 6). Generally speaking, the larger the search bound is, the more edges the *repair channel* occupies, leaving less solution space for consequent groups. Thus, the *repair channels* with less edges are preferred. We then apply an exhaustive search to find the *repair channels* for each source in this group such that, there is no conflict on edges and targets (line 7). Once such a combination of *repair channels* is found, we confirm this solution by recording the *repair channels* and updating the graph, i.e. deleting the chosen edges and targets in this solution (lines 8-9). We then continue with the next group of sources (line 10). Otherwise, the search bound is

```
Input : G = (V, E⃗, w), S, T, L
Output : C = {c_k(s_k, t_k), s_k ∈ S, t_k ∈ T}
1    Initialize G, remove faulty TSVs;
2    Classify diagonal sources into groups P = {p_i};
3    For each group {p_i}
4        For each source s_j ∈ p_i
5            BFS: find all possible repair channels from s_j to t ∈ T,
                 satisfying the length bound L and put into c_j;
6        For search bound sb from 1 to maximum bound;
7            find a repair channel and target for each source
                 such that no conflict exists;
8            If success
9                confirm the solution and update G;
10               break;
11           Else If not success and sb = MAX_hops(c_j)
12               The grid is Irreparable;
13   The grid is reparable;
```

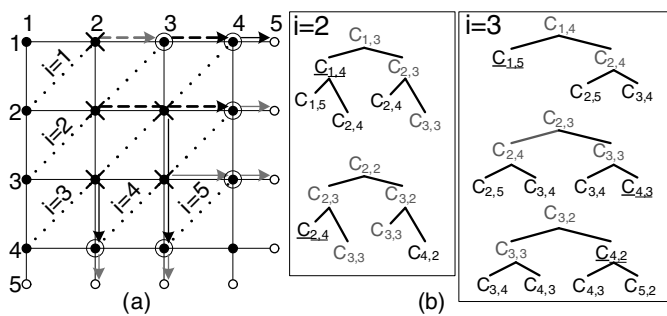Fig. 5.   Proposed Alogrithm

Fig. 6.   (a) Example TSV Grid; (b) Search Procedure Demonstration

increased and the search continues. If no such non-conflict *repair channels* can be found even with the maximum bound, the TSV grid is deemed irreparable (lines 11-12). Otherwise, the heuristic returns the successful repair solution (line 13).

Let us demonstrate how our heuristic works using an example fault map shown in Fig. 6(a). The groups are those signals in diagonal lines (doted lines). To simplify the demonstration, we adopt a flow graph like Fig. 4(a) and index each node with the row and column numbers ($C_{x,y}$). We mark the faulty TSV as cross ("X") and use a circle to denote the node whose TSV is possessed by previous signals. When $i = 1$, $C_{1,2}$ has faulty TSV and it finds a *repair channel* (dashed gray arrows) to the TSV in node $C_{1,3}$ while the other node in this group finds the *repair channel* to its own TSV. When $i = 2$, two signals without their original TSVs find the *repair channels* (dashed black arrows) bounded by 2. The search process is shown in Fig. 6(b). The nodes with fault TSV or possessed TSV are shown in grey color and the non-conflicting *repair channels* end on nodes $C_{1,4}$ and $C_{2,4}$ respectively (underlined). After confirming this solution, the edges and TSVs possessed during $i = 2$ are labeled in grey color and no longer available. The process continues for group $i = 3$, and three more *repair channels* are found for this group (solid black arrows in Fig. 6(a)).

V. EXPERIMENTAL RESULTS

A. Experimental Setup

We set up three types of sample chips (small/medium/large) based on TSV count, that is, thousands [3]/tens of thousands [17]/hundreds of thousands [7] of TSVs. We group TSVs as bundles (16×16 for

small and medium chip and 32×32 for large chip) and place them randomly on the chip. We vary the TSV failure rate ranging from 0.1% to 1%. In the experiments, we use three repair schemes as baseline solutions for comparison. The crossbar scheme [17] with 0.25 redundancy ratio is denoted as "8:2", which indicates that eight signal TSVs and two redundant TSVs are bundled together. Similarly, the signal-shifting scheme [18] and signal-switching [3] scheme with a 0.5 redundancy ratio are denoted as "2:1" and "4:2" respectively. We assume that these schemes always place redundant TSVs near signal TSVs, and hence have no length constraint. The proposed scheme allocates one column and one row of redundant TSVs on two borders of each TSV grid, denoted as "$R \times C : R + C$". For ease of comparison, we set up two TSV grids, "$4 \times 4 : 8$" and "$8 \times 8 : 16$", with redundancy ratio 0.5 and 0.25, respectively.

We conduct three set of experiments. The first two experiments compare the above repair schemes without timing constraint. First, Poisson Distribution is used to inject defective TSVs. Secondly, to study repair efficiency on clustered TSV faults, we adopt a commonly-used Compound Poisson Distribution [27]. The clustering effect parameter (denoted as $Alpha$) is varied under a fixed mean failure rate. Finally, we evaluate the proposed length bounded search algorithm. For the sake of simplicity, the distance between neighboring TSVs within the TSV bundle are all set equally. The length constraint is decreased from 4 to 1 time(s) of this distance value (denoted as "$-T(4)$ to $-T(1)$").

B. Results and Analysis

Fig. 7(a)-(c) show the yield comparison under Poisson Distribution. With the increase of TSV failure rate, the yield with the three baseline repair schemes drops with different gradients while the maximum-flow based scheme can still keep a very high yield. To be specific, the repair efficiency of signal-shifting scheme is less than other solutions because it can only tolerate one fault in its repair unit. The crossbar is better than the preceding one but worse than signal-switching scheme. The signal-switching scheme performs better for small chip and medium chips with up to 98% yield, but in the case of large chips, the yield drops nearly 10%.

From the above results, we can also observe that the redundancy ratio is not the only dominating factor for the final yield, e.g., the 8:2 repair scheme is better than 2:1 scheme, even though its redundancy ratio is only half of the latter scheme. This is because the flexibility for repair has a significant impact on TSV repair efficiency. Let us consider a TSV bundle containing 8 signal TSVs, with 8:2 repair scheme, as long as there are no more than two TSV failures, we can successfully repair them. With 2:1 repair scheme, however, if there are two TSV failures within the same repair unit, this TSV bundle becomes irreparable. When the TSV failure rate is not very high, the possibility to have more than two TSV failures in the bundle (and can be repaired with 2:1 repair scheme) is lower than the possibility to have irreparable double TSV failures. Therefore, 8:2 repair scheme results in higher yield when compared to 2:1 scheme.

Fig. 8(a)-(c) show the yield comparison under Compound Poisson Distribution. We fix the TSV failure rate in this experiment and vary the cluster parameter $Alpha$. It can be observed from the figure that with the increase of clustering effect, the benefits provided by the proposed TSV redundancy scheme become more significant, which proves the effectiveness of our solution for clustered faulty TSVs.

Fig. 9(a)-(c) show the bounded search heuristic with different length constraints. Both 4×4 and 8×8 grids with "-T(1)" length constraint suffer from notable yield loss. This is expected because the *repair path* can only reach one hop of TSVs for repair and hence it
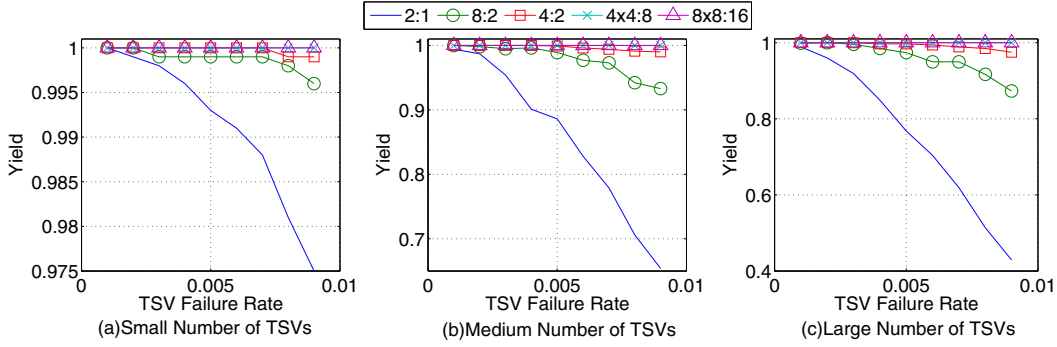
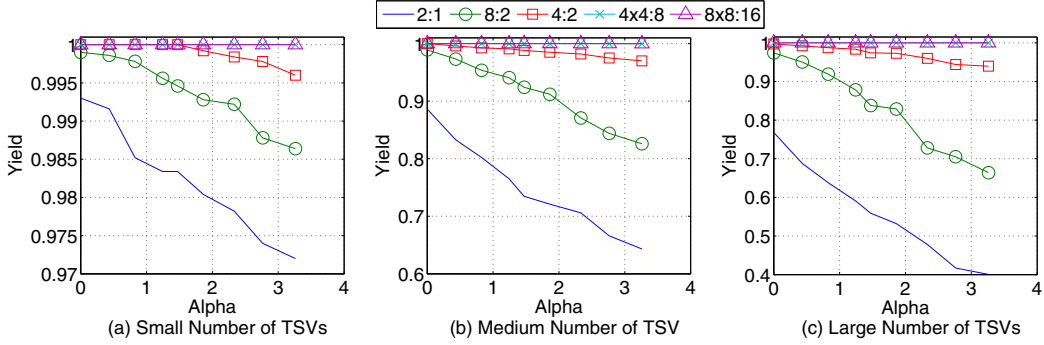Fig. 7. Comparison of Repair Schemes under Poisson Distribution.

Fig. 8. Compound Poisson Distribution with Fixed TSV Failure Rate as 0.5%.
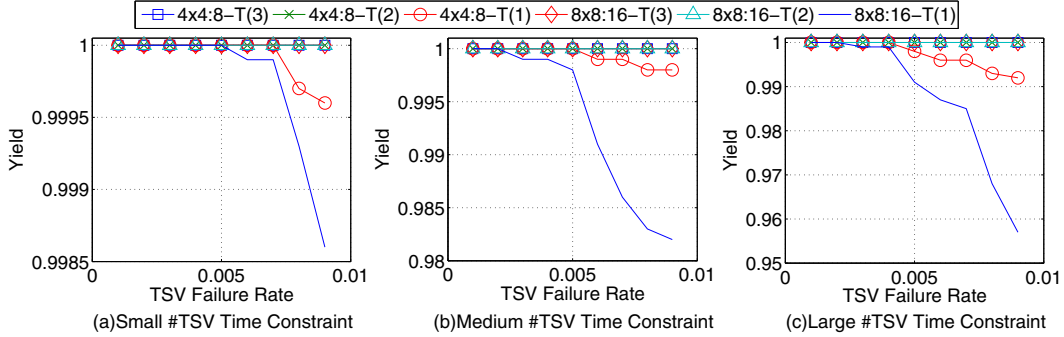

Fig. 9. The Impact of Length Constraints under Poisson Distribution.

is vulnerable to any TSV faults in close proximity. Compared to the "$8 \times 8 - T(1)$" scheme, the "$4 \times 4 - T(1)$" scheme has better repair efficiency. This is not only because of its higher redundancy ratio, but also the less amount of clustered faults in a grid. An interesting observation is that, as long as the more-than-one-hop *repair path* is allowed, the proposed repair scheme can guarantee a very high yield under our experimental settings.

VI. DISCUSSION

A. TSV grid Construction

TSVs are usually bundled together in a 3D-SIC design, and the proposed TSV redundancy scheme can be directly applied to such designs if we treat each TSV bundle as a grid. However, sometimes the shape of the TSV bundle may not be suitable for efficient repair (e.g., a 1×32 TSV bundle). There are also designs with irregularly-placed TSVs. Under the above circumstances, it is essential to construct "logical TSV grids" for repair.

We suggest a "cut and merge" strategy for TSV grid construction when TSVs are bundled together but their shapes are not suitable for our repair scheme. Given a rough size of TSV grid based on the TSV failure rate and TSV redundancy ratio, we cut the TSV bundle according to the grid's size. For TSV bundles with high aspect-ratio (e.g., 1×32), we can first merge two columns into one column and the cluster becomes 2×16 grid. Such merging process continues until the shape is proper. It should be noted that, the merging process would increase the length of edges in the logical TSV grids.

When TSVs are placed irregularly, we propose a simple mapping method as follows. First, we divide the layout into blocks until the number of TSVs in each block is roughly equal to the size of logical TSV grid. Then, we index each TSV with its X-/Y-coordinate (see Fig. 10(a)), which indicates their relative positions amongst TSVs. Next, we construct grids and place TSVs into them according to the index obtained earlier. After that, we map them into the corresponding logical TSV grids (e.g., 4×2 in Fig. 10(b) and 3×3 in Fig. 10(c) in
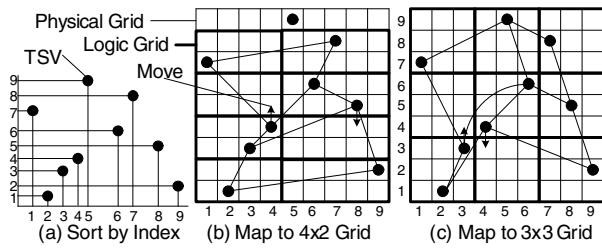
Fig. 10.   Mapping Irregular Placed TSVs.

bold). Finally, we move TSVs so that each cell of the logical grids is assigned with one TSV only. Since the physical grid maintains the relative position of TSVs, we are able to find a good TSV grid construction with few TSV movements.

There are some other considerations during TSV grid construction. For example, it would be beneficial to have timing critical signals with shorter *repair channels* in a grid so that we can have more repair candidates, which will be explored in our future research work.

B. Cost Analysis

Under the same redundancy ratio, the proposed repair scheme has higher hardware cost when compared to other TSV repair solutions, i.e., our redundancy scheme requires three 1-to-3 muxes for each TSV, while signal-shifting and signal-switching have one 1-to-3 mux for each signal TSV. However, the extra cost is justified by the corresponding significant TSV yield improvement. It should be noted that, the amount of redundancy is configurable in the proposed architecture. That is, we can adjust the redundancy ratio by either varying the TSV grid size or changing the redundancy allocation scheme (e.g., we can allocate redundant TSVs only on one border). From this perspective, to achieve the same TSV yield, the required redundant TSVs with the proposed repair scheme is much less than existing solutions, rendering less hardware overhead.

Finally, the runtime of the proposed repair algorithm is very small. For large dies with hundreds of thousands TSVs, it takes only tens of millisecond to obtain the repair solution, if any.

VII. Conclusion

In this paper, we propose a novel TSV redundancy architecture and the corresponding repair algorithm for yield enhancement of 3D-stacked ICs. When compared to prior techniques, the proposed solutions enable faulty TSVs to be repaired by spares that are distant, thus is suitable for repairing clustered TSV faults. Experimental results demonstrate the effectiveness of the proposed technique.

VIII. Acknowledgements

References

[1] V. Suntharalingam, et al. Megapixel CMOS image sensor fabricated in three-dimensional integrated circuit technology. In *IEEE Int'l Solid-State Circuits Conf.*, pp. 356–357, 2005.

[2] H. Yoshikawa, et al. Chip scale camera module (CSCM) using through-silicon-via (TSV). In *IEEE Int'l Solid-State Circuits Conf.*, pp. 476–477, 2009.

[3] U. Kang, et al. 8 Gb 3-D DDR3 DRAM using through-silicon-via technology. *IEEE Journal of Solid-State Circuits*, 45(1):111–119, Jan. 2010.

[4] T. Mitsuhashi et al. Development of 3-D-packaging process technology for stacked memory chips. In *Proc. Enabling Technologies for 3-D Integration Symp.*, pp. 155–162, 2006.

[5] M. Kawano, et al. A 3D packaging technology for 4 Gbit stacked DRAM with 3 Gbps data transfer. In *IEEE Int'l Electron Devices Meeting*, pp. 1–4, 2006.

[6] G.H. Loh. 3D-stacked memory architectures for multi-core processors. In *Proc. Int'l Symp. on Computer Architecture*, pp. 453–464, 2008.

[7] H.W. Dong, N.H. Seong, D.L. Lewis, and H.-H.S. Lee. An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth. In *IEEE Int'l Symp. on High Performance Computer Architecture*, pp. 1–12, 2010.

[8] T. Zhang, et al. A customized design of DRAM controller for on-chip 3D DRAM stacking. In *Proc. IEEE Conf. on Custom Integrated Circuits*, pp. 1–4, 2010.

[9] X.Y. Dong and Y. Xie. System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs). In *Proc. IEEE Asia South Pacific Design Automation Conf.*, pp. 234–241, 2009.

[10] G. Smith, L. Smith, S. Hosali, and S. Arkalgud. Yield considerations in the choice of 3D technology. In *Int'l Symp. on Semiconductor Manufacturing*, pp. 1–3, 2007.

[11] Q. Xu, L. Jiang, H. Li, and B. Eklow. Yield enhancement for 3D-stacked ICs: Recent advances and challenges. In *Proc. IEEE Asia South Pacific Design Automation Conf.*, to appear, 2012.

[12] H.-H.S. Lee and K. Chakrabarty. Test challenges for 3D integrated circuits. *IEEE Design Test of Computers*, 26(5):26–35, sept.-oct. 2009.

[13] L. Jiang, R. Ye, and Q. Xu. Yield enhancement for 3D-stacked memory by redundancy sharing across dies. In *Proc. Int'l Conf. on Computer-Aided Design*, pp. 230–234, 2010.

[14] C.W. Chou, Y.J. Huang, and J.F. Li. Yield-enhancement techniques for 3D random access memories. In *Int'l Symp. on VLSI Design Automation and Test*, pp. 104–107, 2010.

[15] C. Ferri, S. Reda, and R.I. Bahar. Strategies for improving the parametric yield and profits of 3D ICs. In *Proc. Int'l Conf. on Computer-Aided Design*, pp. 220–226, 2007.

[16] J.U. Knickerbocker, et al. Three-dimensional silicon integration. *IBM Journal of Research and Development*, 52(6):553–569, nov. 2008.

[17] I. Loi, et al. A low-overhead fault tolerance scheme for TSV-based 3D network on chip links. In *Proc. Int'l Conf. on Computer-Aided Design*, pp. 598–602, 2008.

[18] A.C. Hsieh, et al. TSV redundancy: Architecture and design issues in 3D IC. In *Proc. Design, Automation, and Test in Europe Conf. Exhibition*, pp. 166–171, 2010.

[19] B. Kim, C. Sharbono, T. Ritzdorf, and D. Schmauch. Factors affecting copper filling process within high aspect ratio deep vias for 3D chip stacking. In *Proc. Electronic Components and Technology Conf.*, pp. 6–pp, 2006.

[20] A.P. Karmarkar, X. Xu, and V. Moroz. Performanace and reliability analysis of 3D-integration structures employing through silicon via (TSV). In *IEEE Int'l Reliability Physics Symp.*, pp. 682–687, 2009.

[21] E.J. Marinissen and Y. Zorian. Testing 3D chips containing through-silicon vias. In *Proc. IEEE Int'l Test Conf.*, pp. 1–11, 2009.

[22] A. W. Topol, et al. Three-dimensional integrated circuits. *IBM Journal of Research and Development*, 50(4.5):491–506, july 2006.

[23] J. S. N. Jean, H.C. Fu, and S.Y. Kung. Yield enhancement for WSI array processors using two-and-half-track switches. In *Proc. Int'l Conf. on Wafer Scale Integration*, pp. 243–250, 1990.

[24] L. Jiang, L. Huang and Q. Xu. Test architecture design and optimization for three-dimensional SoCs. *Proc. Design, Automation, and Test in Europe Conf. Exhibition*, pp. 220-225, 2009.

[25] A.V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):783–797, 1998.

[26] G. Baier, et al. Length-bounded cuts and flows. *Automata, languages and programming*, pp. 679–690, 2006.

[27] I. Koren and Z. Koren. Defect tolerance in VLSI circuits: techniques and yield analysis. *Proc. of the IEEE*, 86(9):1819–1838, 1998.