

On Effective Through-Silicon Via Repair for 3-D-Stacked ICs

Li Jiang, *Student Member, IEEE*, Qiang Xu, *Member, IEEE*, and Bill Eklow, *Fellow, IEEE*

Abstract—3-D-stacked integrated circuits (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 3-D-stacked ICs must be as high as possible, requiring TSVs to be repairable. 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 cleanliness of silicon dies, rendering prior TSV redundancy solutions less effective. Furthermore, existing techniques consume a lot of redundant TSVs that are still costly in the current TSV process. This inefficient TSV redundancy can limit the amount of TSVs that is allowed to use and may even become the obstacle to commercial production. To resolve this problem, we present a novel TSV repair framework, including a hardware redundancy architecture that enables faulty TSVs to be repaired by redundant TSVs that are farther apart, the corresponding repair algorithm and the redundancy architecture construction. By doing so, the manufacturing yield for 3-D-stacked ICs can be dramatically improved, as demonstrated in our experimental results.

Index Terms—3-D stacking, redundancy, through-silicon vias (TSV) repair, yield enhancement.

I. INTRODUCTION

3-D 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 3-D-stacked IC (3-D SIC) products for CMOS image sensor camera modules are already in volume production [1], [2]. 3-D-stacked memory products have also recently been announced by various companies [3]–[5]. Moreover, various techniques with massive use of TSVs have been proposed to fully exploit the benefits of this emerging technology [6]–[8].

Manuscript received March 30, 2012; revised August 20, 2012; accepted October 22, 2012. Date of current version March 15, 2013. This work was supported in part by a research grant from Cisco Systems, Inc. This paper was recommended by Associate Editor G. Loh.

L. Jiang is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.

Q. Xu is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, and also with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: qxu@cse.cuhk.edu.hk).

B. Eklow is with Cisco Systems, Inc., San Jose, CA 95134 USA (e-mail: beklow@cisco.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2012.2228742

While 3-D SICs provide many benefits over traditional 2-D ICs and have been produced in certain applications, they can widely be adopted only when their design and manufacturing costs are commercially viable [9]. Among the various factors that affect the 3-D 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 losses in 3-D SICs:

- 1) *stack yield loss* caused by defects in one or more of the stacked dies;
- 2) *assembly yield loss* caused by defects occurred during the assembling process.

Various yield enhancement techniques for 3-D SICs have been proposed, as surveyed in [11]. To improve the stack yield of 3-D SICs, it is critical to conduct prebond testing to screen out defective dies so that only known good dies 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 yields [13]–[15].

The assembly process for 3-D 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 for enhancing assembly yields 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 on the winding level of the thinned wafer and the surface roughness and cleanliness 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 a 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 paper. Experimental results show that the proposed solution outperforms prior techniques, especially when the number of TSVs used in the 3-D SICs is large and/or the clustering effect is significant.

The remainder of this paper is organized as follows. Section II presents the preliminaries and motivation of this paper. 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 corresponding redundancy architecture construction. Section VI presents the experimental results for various hypothetical 3-D SICs. We then discuss several practical considerations to use the proposed technique in Section VII. Finally, we conclude this paper in Section VIII.

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 recent 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 (#Redundant TSVs/#Signal TSVs), repair capability, and hardware costs.

Samsung [3] presented a TSV redundancy strategy used to improve the yield of its 3-D 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 that 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 [17], for an $N \times N$ TSV grid used as NoC links, redundant rows or columns of TSVs are added for defect-tolerance. Suppose that 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 that M redundant columns or rows are added, the redundancy ratio of this technique is $M : N$, and it can tolerate any M TSV failure in each row or 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

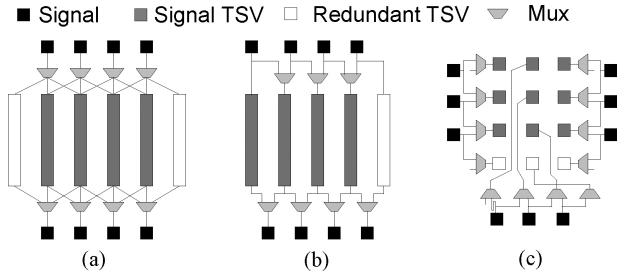


Fig. 1. Existing TSV redundancy solutions. (a) Signal switching [3]. (b) Signal shifting [18]. (c) Crossbar [17].

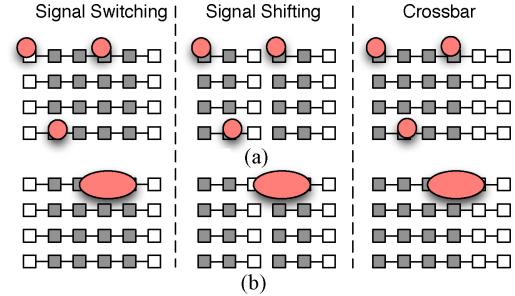


Fig. 2. Results of existing repair schemes assuming 1/2 redundancy ratio. (a) Random defect. (b) Clustered defect.

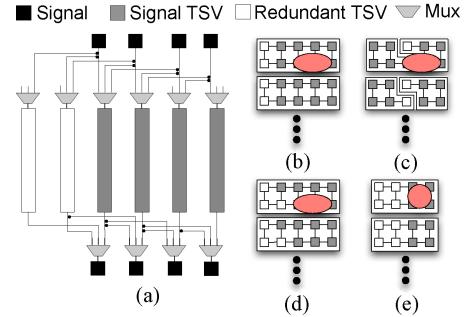


Fig. 3. Existing clustered faults aware repair scheme [23]. (a) TSV grouping. (b) Impact of clustered faults. (c) Shrink group size. (d), (e) Increase redundancy ratio.

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 repairable, there exists one or more signal TSV groups that are vulnerable to clustered faults and will become irreparable, although 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. We use two examples in Fig. 2 to demonstrate the above concern. For simplicity, the bundled signal TSVs and redundancies are linked by a line in the fault map. Although the redundancy ratios of the prior solutions are all the same (1:2), their repair efficiencies behave differently after the clustered TSV faults occur. In Fig. 2(a), all the prior solutions can successfully repair the random TSV faults that are sparsely located. While in Fig. 2(b), all the existing techniques fail to repair the three clustered TSV faults due to the lack of redundancies nearby.

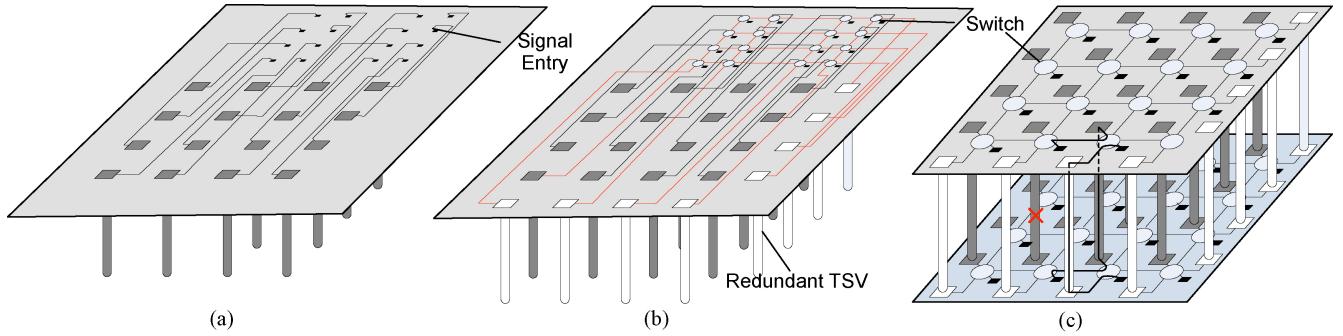


Fig. 4. (a) Example physical layout of original TSV bundle and their signal entries. (b) Physical implementation of proposed TSV redundancy architecture. (c) Conceptual view of the proposed TSV redundancy architecture.

Recently, the clustered TSV faults get noticed by several works [23], [24]. In [23], signal TSVs and redundant TSVs are grouped together as a crossbar scheme. Each signal TSV has r repair candidates if the group is assigned with r redundant TSVs [e.g., Fig. 3(a)]. In order to tolerate the clustered faults [e.g., Fig. 3(b)], they either choose to shrink the group size [e.g., Fig. 3(c)] or increase the redundancy ratio [e.g., Fig. 3(d) and (e)]. In order to maximize the yield and minimize the multiplexors, the group size and redundancy ratio are determined through probability analysis of successful repair. The derived optimal solution, however, is still not cheap, especially in terms of redundancy cost. That is because the repair structure is fixed in advance, rendering an inflexible repair and hence most of the redundant TSVs remain useless. Motivated by the above, in this paper, we propose a novel repair framework that is effective and flexible to repair clustered TSV faults with least redundancy costs.

III. PROPOSED TSV REDUNDANCY ARCHITECTURE

In order to handle the 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. In this section, we first demonstrate the overall architecture and then detail the switch design and signal re-routing mechanism.

A. Overall Structure

An example layout of TSVs is shown in Fig. 4(a), wherein the signal connecting TSV (noted as signal entry) is located nearby. Around each TSV, there is a keep-out-zone that no logic gate can be placed. Thus, the signal can only be routed into this TSV bundle by wires. It first injects a switch into the TSV-signal link and places the redundant TSVs along two borders of the TSV bundle [Fig. 4(b)]. Each redundancy TSV links to a switch by wires (shown in red color). Note that the exact placement of these redundancy TSVs can be tuned to facilitate the placement and routing in physical design. This physical layout is mapped to a logical TSV grid [see Fig. 4(c)] for clarity. In the remainder of this paper, we will use this logical redundancy architecture instead.

Inspired by the compensation path problem [25], in this architecture, if one signal is disconnected due to a TSV

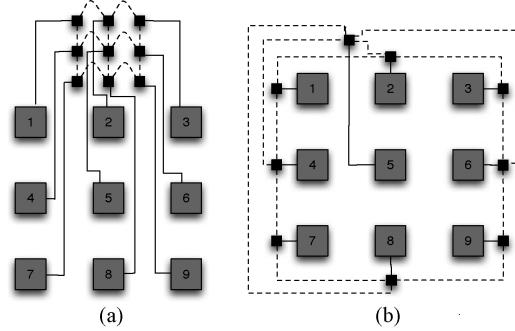


Fig. 5. Switch design and routing capability. (a) Switch design. (b) TSV grid with edge-disjoint repair path.

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 3-D SIC designs [5], [8]; those regularly placed TSVs can be naturally linked together to construct the proposed TSV redundancy architecture. In the 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, e.g., in Fig. 4(c), which are placed on the east and south borders of the TSV grid. Thus, we constrain signals to route from two directions (from west to east or from north to south). Fig. 5(a) shows the schematic of the corresponding switch design. The signal entry and its original 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

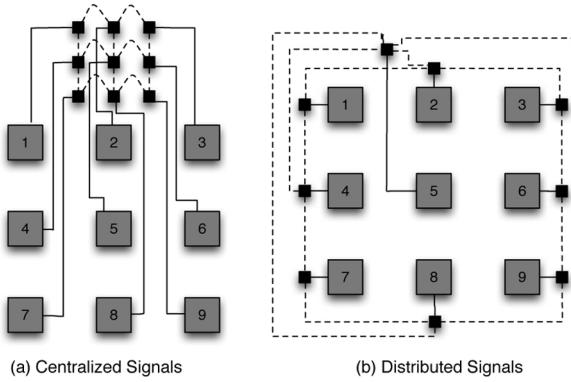


Fig. 6. Timing issue caused by signal rerouting. (a) Centralized signals. (b) Distributed signals.

ports). The design principle is that the signal port and two linking ports (north and west) have a mux capable of linking to the TSV port and the remaining linking ports (east and south).

We use Fig. 5(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 in a redundant TSV. We denote 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 the 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 are obtained via testing,¹ a repair algorithm is essential to analyze whether the TSV grid is repairable and generate repair paths for each faulty TSV, if possible.

A. Maximum Flow Method Based Approach

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 [27] 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

¹Testing is beyond the scope of this paper. Interested readers may refer to [18], [21], and [26] for more details.

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. 7(a)].

B. Additional Delay Due to Signal Rerouting

While the above problem formulation and its corresponding solution are simple and effective, it does not take the additional delay introduced during repair into consideration. First, let us take a look at the extra delay caused by the switch inserted between each TSV and its signal entry. Each switch has three Muxes (i.e., six gates). Assuming the average area of a gate is 3125F^2 , the size of the switch is approximately $0.9\mu\text{m} \times 0.9\mu\text{m}$ in 65 nm technology. Comparing to the TSV size (in an order of magnitude of $10\mu\text{m}$), it is negligible. To reroute to the neighboring TSV, one signal only need to pass four gates in maximum. Conclusively, it is not the extra hardware in the proposed architecture that contributes the extra delay. Instead, the wire of accessing the neighboring TSV mainly determines the extra delay.

Furthermore, the wire length of rerouting the signal is determined by the layout of signal entries. We classify their layout into two scenarios, one of which is mentioned in Fig. 4 and denoted as centralized signal entries [see Fig. 6(a)], wherein the switches are assumed to be near the signal entry, and thus omitted for clarity. Applications such as 3-D NoC and 3-D-stacked memory prefer to connect TSVs in this way for massive data transformation. Under this circumstance, the extra delay after repairing depends on the different distances of the original TSV and repairing TSV away from the respect signal entries. And this difference is roughly equivalent to the distance between the two TSVs (e.g., repairing TSV 2 using TSV 5). While the extra the delay after repairing the TSV would even decrease in other cases (e.g., repairing TSV 1 by TSV 2). Assuming the TSV distance is in the same order of its pitch ($10\mu\text{m}$), this extra delay is limited and negligible, except for some critical signals with tight timing margin.

For the other scenario, the signal entries are distributed near to their respect TSVs [see Fig. 6(b)]. In this situation, the proposed TSV redundancy architecture confronts with timing issues since we have to take the distance between signal entries (switches) into consideration. For example, repairing TSV 5 by TSV 8 leads the signal to access as far as half of the bundle's perimeter away. A repairing is available only if the extra delay of rerouting the signal would not violate the timing constraint. As a result, it is essential to consider timing constraint in the proposed TSV repair algorithm.

C. Problem Analysis Due to Timing Constraint

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.

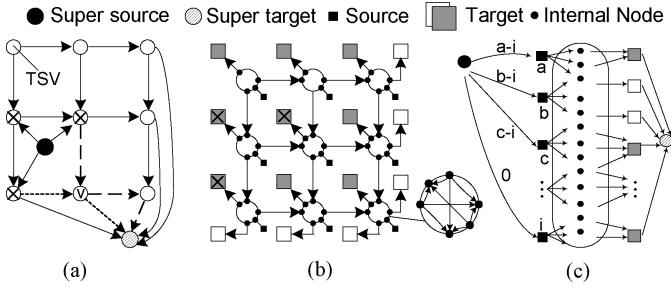


Fig. 7. Problem transformation. (a) Flow graph to find repair paths. (b) Flow graph to find repair channels. (c) Length bound flow graph.

With the above, directly using maximum flow method to solve our problem is not applicable because: 1) the flow in the maximum flow method has no sense of length, and 2) if two repair paths intersect in the same TSV [e.g., dashed path and dotted path intersected in TSV V in Fig. 7(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 that this problem is an NP problem first, as shown in the following.

First, we transform the flow graph by setting all signals as sources and all the fault-free TSVs as targets [see Fig. 7(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. 7(c), where each source is labeled with length constraint ($a \dots i$), among which we suppose i 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., $a - i$). Thus, the original problem becomes an NP-complete maximum length-bounded flow problem, which is to find a maximum flow between one source and one target where the lengths of all flows are bounded by a length constraint [28].

D. Repair With Length Constraint

For the sake of simplicity, in this paper, 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. 8, 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 channel can only go toward 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

Input : $G = (V, \vec{E}, w), S, T, \mathcal{L}$
Output : $C = \{c_k(s_k, t_k), s_k \in S, t_k \in T\}$

```

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  $s_j \in p_i$ 
5     BFS: find all possible repair channels from  $s_j$  to  $t \in T$ ,
           satisfying the length bound  $\mathcal{L}$  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. 8. Proposed algorithm.

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 satisfy the length constraint using the breadth first search (lines 4, 5). To avoid an extremely large solution space during an exhaustive search, we constrain the candidate repair channels in terms of their number of edges or hops. The search bound iteratively increases from 1 as long as it does not exceed some predefined 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 increased and the search continues. If no such nonconflicting 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, as shown in Fig. 9(a). The groups are those signals in diagonal lines (dotted lines). To simplify the demonstration, we adopt a flow graph such as Fig. 7(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. 9(b). The nodes with fault TSV or possessed TSV are shown in grey color and the nonconflicting repair channels end on nodes $C_{1,4}$ and $C_{2,4}$, respectively (underlined). After confirming this solution, the

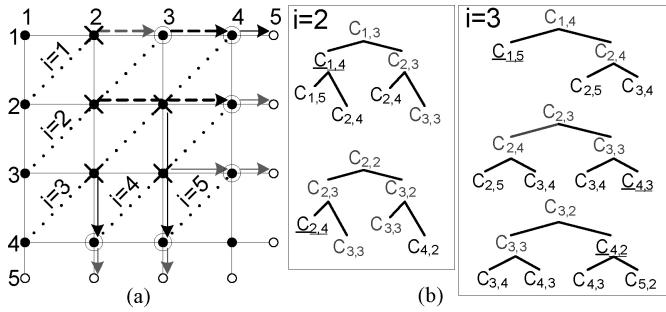


Fig. 9. (a) Example TSV grid. (b) Search procedure demonstration.

edges and TSVs possessed during $i = 2$ are labeled in grey color and are no longer available. The process continues for group $i = 3$, and three more repair channels are found for this group (solid black arrows).

V. TSV REDUNDANCY ARCHITECTURE CONSTRUCTION

In order to integrate the proposed TSV redundancy technique into the design flow, we plan to insert redundant TSVs and supporting infrastructures, i.e., muxes and wires, right after the TSVs planning but prior to the placement and detail routing. During this process, the key step is to construct the TSV redundancy architecture, i.e., to determine the distance of neighboring TSVs, which influences the reparability in two contradictory aspects. From the perspective of the whole TSV grid, a higher routability indicates a higher reparability since each signal is more flexible to find a replacement. Thus, it is preferred to minimize the distance between neighboring TSVs so that more replacement candidates can be reached. On the other side, it is likely that the presence of a single TSV fault increases the chance of more defective TSVs in close vicinity [23], [24]. Once a TSV is found faulty, we are unwilling to see that its successive neighboring TSVs are also faulty, blocking up its repair paths. In that sense, it seems better, on the contrary, to maximize the distance between neighboring TSVs as long as this distance fulfills the length constraint. In order to investigate the impact of this distance, we first conduct the probability analysis on the reparability of TSV redundancy grid assuming that the clustered TSV faults are spatially correlated. Then, a topology mapping strategy is proposed for TSV grid construction to enhance the reparability.

A. Defect Probability Model With Spatial Correlation

To analyze the probability of successful repair, i.e., reparability, it is essential to derive a probability model of the clustered TSV faults. The compound Poisson distribution [32] is widely accepted to model the clustering effect, in which the defect count follows Poisson distribution compounded with a Gamma function presenting the distribution of defect density. To model the spatial correlation, a center-satellite model [30] is proposed, where the distributions of the cluster centers are described by a 2-D distribution function and the distribution of the satellites (defects) with respect to the cluster center is also described by a 2-D distribution function. In this case, the defect probabilities in the regions near defect clusters

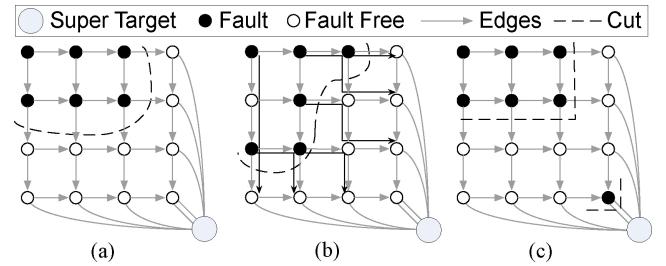


Fig. 10. Reparability condition. (a) Irreparable example. (b) Reparable example. (c) Minimal orthogonal cut equals to min-cut.

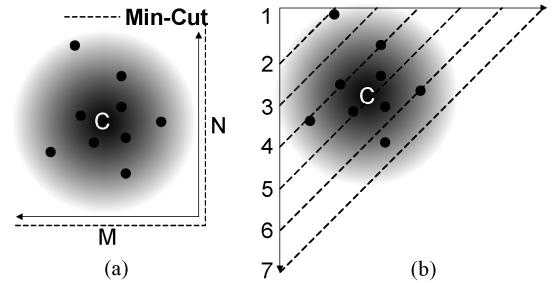


Fig. 11. Defect probability model. (a) Maximum flow method. (b) Length bounded search heuristic.

are higher than other regions. Approximately, this defect probability is inversely proportional to the distance from the existing defects [31]. If there are already N_c defects (regarded as cluster centers), the defect probability of TSV_i , p_i can be expressed as

$$p_i \propto \sum_{j=1}^{N_c} \left(\frac{1}{d_{ic}} \right)^\alpha \quad (1)$$

where d_{ic} is the distance between TSV_i and existing cluster center and α is the clustering effect.

B. Reparability Analysis

To analyze the reparability, we first investigate the metrics to determine whether the proposed repair scheme can survive the clustered TSV faults. As stated in the previous section, in order to repair the TSV faults, we have to find edge-disjointed paths from the faulty TSVs to redundant TSVs located in the borders. According to the max-flow min-cut theorem, the minimum cut of the induced flow graph must not be less than the number of faulty TSVs. Fig. 10(a) shows an irreparable fault map, in which the value of the min-cut (dashed curve) is less than the number of faulty TSVs. Whereas the fault map in Fig. 10(b) has a min-cut equal to the number of faulty TSVs, indicating a reparable solution (black solid arrows). The fault map in Fig. 10(c) shows that, on the contrary, although the min-cut of these seven sources is equal to seven, the fault map is still irreparable. As a result, we have the following theory.

Theorem 1: The fault map is reparable if and only if there is no such a subset of sources whose min-cut is less than the size of this subset.

Proof: To repair all the faults, we need to find a repair path, and an edge-disjointed path for each the source. If the number of sources in any subset is larger than the min-cut of

this subset, then the number of edge-disjointed path is also larger than the min-cut. Since the repair scheme constrains that the repair path can only go toward east and south, based on the min-cut maximum-flow theorem, there is no way in which we can find all edge-disjointed paths for each source that can pass through the min-cut. Thus, the fault map is irreparable. ■

With the above theorem, we can model the reparability of the TSV grid by investigating the probability that any fault cluster is reparable. The proposed two repair algorithms are modeled, respectively. As in [30], the probability of the cluster center (c) is assumed to be Poisson distributed. The defect probability of satellite TSVs p_i is given in (1). In order to model the reparability of the maximum flow based algorithm, we assume the size of the fault cluster is M by N and hence the min-cut of the fault cluster is $M + N$ [see Fig. 11(a)]. The random variable X denotes the number of defective TSVs among $M \times N$ TSVs in total. As mentioned above, this fault cluster is reparable only if the number of faulty TSVs in the cluster is less than the min-cut $M + N$. First, we randomly select x ($x \leq M + N$) TSVs from the total $M \times N$ TSVs as a faulty TSVs set F_x , resulting in totally $\binom{M \times N}{x}$ combinations. We obtain the probability of each combination C_i by calculating the product of defective probabilities for the faulty TSVs selected in F_{x,C_i} and the nondefective probabilities for the rest of TSVs that are not selected ($\Omega - F_{x,C_i}$). Then, the probabilities of all the combinations are accumulated for this x . Finally, the reparability is formulated by cumulating the derived probabilities of all possible x s as

$$P(X \leq M + N) = \sum_{x=0}^{M+N} \left[\sum_i \left[\prod_{k \in F_{x,C_i}} p_k \prod_{j \in \Omega - F_{x,C_i}} (1 - p_j) \right] \right] \quad (2)$$

where p_k and p_j are the defective probabilities of TSV k and j obtained from (1). For the sake of simplification, we just assume an average defective probability and (2) can be approximated as follows:

$$P(X \leq M + N) = \sum_{x=0}^{M+N} \left[\binom{M \times N}{x} p^x (1 - p)^{M \times N - x} \right] \quad (3)$$

where p is the average defective probability.

In order to apply the min-cut maximum-flow theorem to the proposed length bounded search heuristic, we can draw a set of right triangles with their hypotenuses representing the min-cuts. These right triangles are sorted in ascending order of the size until the last one embodies the whole fault cluster [see Fig. 11(b)]. Empirically, the set of right triangles represent the repair process that the TSV groups in the diagonal lines (i.e., the hypotenuses) are repaired iteratively. Once a faulty TSV is repaired according to the heuristic, a fault-free TSV is occupied in the next orthogonal hypotenuse. Thus, we need to find another fault-free TSV for this occupied TSV in the next iteration. From a higher perspective, the number of occupied TSVs is accumulated in successive hypotenuses iteratively. This process continues until the redundant TSVs are reached. Obviously, the repair scheme fails as long as the number of faulty TSVs embodied by any right triangle is larger than the

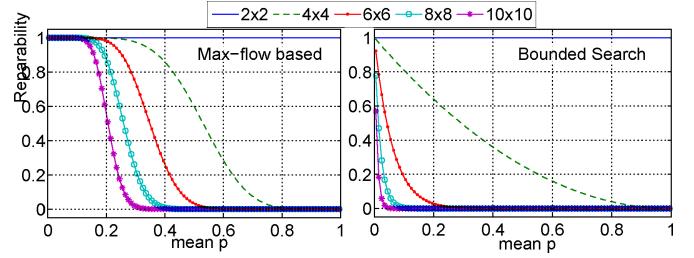


Fig. 12. Reparability approximation trends with respect to average defect probability and cluster size (from 2×2 to 10×10).

min-cut (number of edges) long with the hypotenuse. Suppose that we need n right triangles $T_n = \{t_1, t_2, \dots, t_n\}$ ($\text{Size}(t_i) < \text{Size}(t_{i+1})$) to embody the fault cluster. The random variable x_i denotes the number of faulty TSVs inside the i th right triangle that contains $\text{Num}(t_i) = \frac{i \times (i+1)}{2}$ TSVs in total. The reparability can be approximated in the following manner:

$$\prod_{i=1}^n P(x_i \leq \text{MinCut}(t_i)) = \prod_{i=1}^n P(x_i \leq 2i). \quad (4)$$

For any right triangle t_i , the reparable probability is

$$P(x_i \leq 2i) = \sum_{x_i=0}^{2i} \left[\binom{\text{Num}(t_i)}{x_i} p^{x_i} (1 - p)^{\text{Num}(t_i) - x_i} \right] \quad (5)$$

where p is the average defective probability.

Based on the above equations, we know that the key factor is the distance between TSVs. According to (1), the smaller distance between TSVs indicates that the defect probability p becomes higher. Although the exact defect probability is related to the clustering effect and defect intensity, the TSVs near the cluster center would have a very high probability to fail. We can approximately assume that the average defect probability within this fault cluster approaches 1 (i.e., $p \rightarrow 1$), which means that the fault has a more likelihood to occur. On the contrary, if this distance is larger, p approaches to a fixed failure rate, which indicates a random defect.

Now, let us consider this trend according to the approximated reparability from the above equations (see Fig. 12). As the average defect probability increases, the reparability drops dramatically. At the same time, it becomes more difficult to repair those fault clusters with larger sizes. Comparing the optimal solutions derived by the maximum flow algorithm, the bounded search heuristic is more vulnerable to the clustering effect. In order to enhance the reparability, it is urgent to reduce the average defect probability and reduce the size of fault cluster based on the above analysis.

C. Topology Mapping

In order to enhance the reparability, the basic strategy is either increasing the distance between TSVs or shrinking the size of the faulty TSV cluster. However, the distance of TSVs is normally determined according to the design specification and can hardly be changed afterwards. Similarly, the size of the faulty TSV cluster is also beyond our control. Fortunately, we can achieve both the above strategies by topology mapping. A topology mapping is a process that maps the TSVs from the

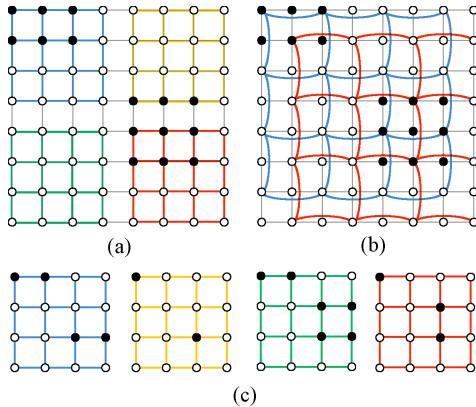


Fig. 13. Examples of the topology mapping and the change of faulty map in TSV grid. (a) Physical mapping. (b) Topological mapping. (c) Faulty maps after topological mapping.

physical layout to a logical TSV grid. Thus, the neighbors in this logical grid (topological neighbors) are not necessarily the neighbors in the physical layout. To demonstrate this concept, we first show a straightforward mapping that the topological neighbors are the exact physical neighbors. In Fig. 13(a), an 8×8 TSV bundle is split into four 4×4 TSV grids. Although the proposed TSV redundancy architecture is inherently able to tolerate TSV clustering fault, applying the above topology disposes the TSV redundancy architecture to the risk of unsuccessful repair. For example, two faulty TSV clusters [black dots in Fig. 13(a)] are located into two TSV grids and make them irreparable. To overcome this vulnerability, the physical neighboring TSVs are distributed apart in derived grids or even into different grids. Fig. 13(b) renders a map such that the topological neighbors are 2-hop away (with one gapped TSV in between) in physical layout. The derived four 4×4 TSV grids are shown in Fig. 13(c) and are able to repair all the faulty TSVs. By employing this topology mapping, the clustered faulty TSVs are distributed, indicating a larger distance between TSVs and smaller cluster size and hence the reparability is improved.

The topology mapping would increase the delay of signal rerouting; however, the affect is limited. As we discussed in Section IV-B, the extra delay is roughly equal to the distance between to-be-repaired TSV and the repairing candidate TSV, if the switches and signal entries are centralized. Mapping TSVs in 2-hop distance away in the original grid into a logical TSV grid would double this extra delay. Since the state-of-art TSV process technology can achieve $< 10 \mu\text{m}$ pitch, doubling this delay after topology mapping would not affect the delay too much. Besides, the topology mapping could also be used upon the proposed timing aware length bounded search scheme, which can improve the yield while keeping the signals under the timing constraint.

VI. EXPERIMENTAL RESULTS

A. Experimental Setup

Defect rate and the number of TSVs are two key parameters for yield calculation that also determine the effectiveness of a repair scheme. The TSV number associates with a specific design, while the defect rate is related to a specific TSV process

TABLE I
TSV RELATED EXPERIMENTAL SETUP

Work	Defect Rate	TSV Pitch	Yield w/o Spare	TSV Number
IBM'05 [36]	13.9E-6	$0.4 \mu\text{m}$	95% 88%	1k–10k
IMEC'06 [35]	40.0E-6	$10 \mu\text{m}$	67%	10k
HRI'07 [33]	9.75E-6	—	68%	100k
HRI'09 [34]	7.95E-7	—	$\geq 90\%$	100k
SAMSUNG'09 [3]	0.63%	—	15%	300

technology (e.g., the pitch and aspect-ratio of TSV). Various settings of these parameters presented in previous works are summarized in Table I. A commonly acceptable failure rate is between 10^{-5} and 10^{-4} except for two extremes (i.e., Samsung and HRI'09). In this paper, we vary both these two parameters in a reasonable range.

We set up three types of sample chips (small/medium/large) based on TSV count, that is, 1k, 16k, and 128k. We group TSVs as bundles (16×16 for small chip and 32×32 for medium and large chips) and place them randomly on the chip. We vary the TSV failure rate in a range from 10^{-5} to 10^{-4} . In the experiments, we use three repair schemes as baseline solutions for comparison. The crossbar scheme [17] with a 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. 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 ratios 0.5 and 0.25, respectively. The 2HD indicates that the topological neighbors have two hops distance in a physical layout. After topology mapping with 2-hop distance, TSV grids 8×8 contain four 4×4 subgrids. Similarly, a 16×16 grid has eight 4×4 subgrids after a 4-hop distance topology mapping. Hence, numbers of redundancy TSVs are 32 and 128, respectively. For the sake of simplicity, the distances 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 (denoted as $-T(4)$ to $-T(1)$).

We conduct two sets of experiments. In the first set, the repairing efficiencies of the proposed maximum-flow-based method with and without topology mapping are compared with prior techniques. And the proposed length bounded search algorithm is evaluated under diverse length (timing) constraints in the second set. To study repair efficiency on clustered TSV faults, we adopt the mentioned compound Poisson distribution combined with spatial correlation in the both sets. The clustering effect parameter (denoted as *Alpha*) is varied from 0 to 3. Particularly, when *Alpha* is equal to 0, it degrades to a Poisson distribution.

B. Results and Analysis

The experimental results with $\text{Alpha} < 1$ are not shown in the paper, as all the repair schemes can guarantee 100% yield. The main reason is that the yield of a single TSV is

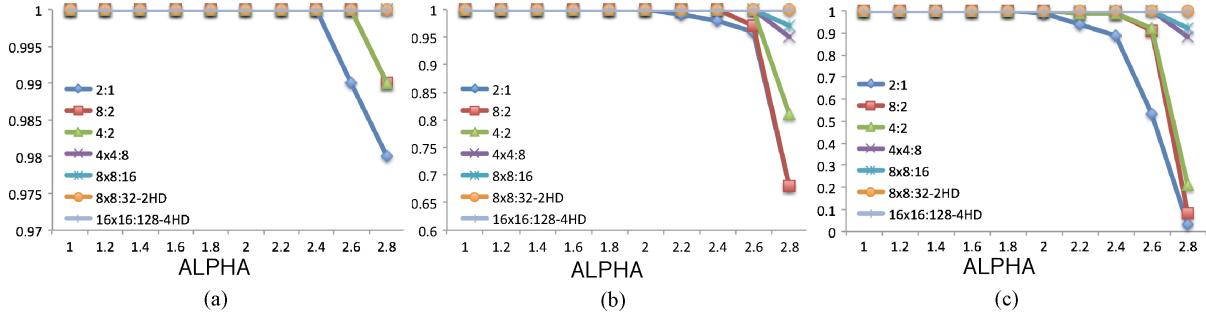


Fig. 14. Repair schemes varying alpha from 1 to 3. (a) 1K TSV, failure rate = 5E-5. (b) 16K TSV, failure rate = 5E-5. (c) 128K TSV, failure rate = 5E-5.

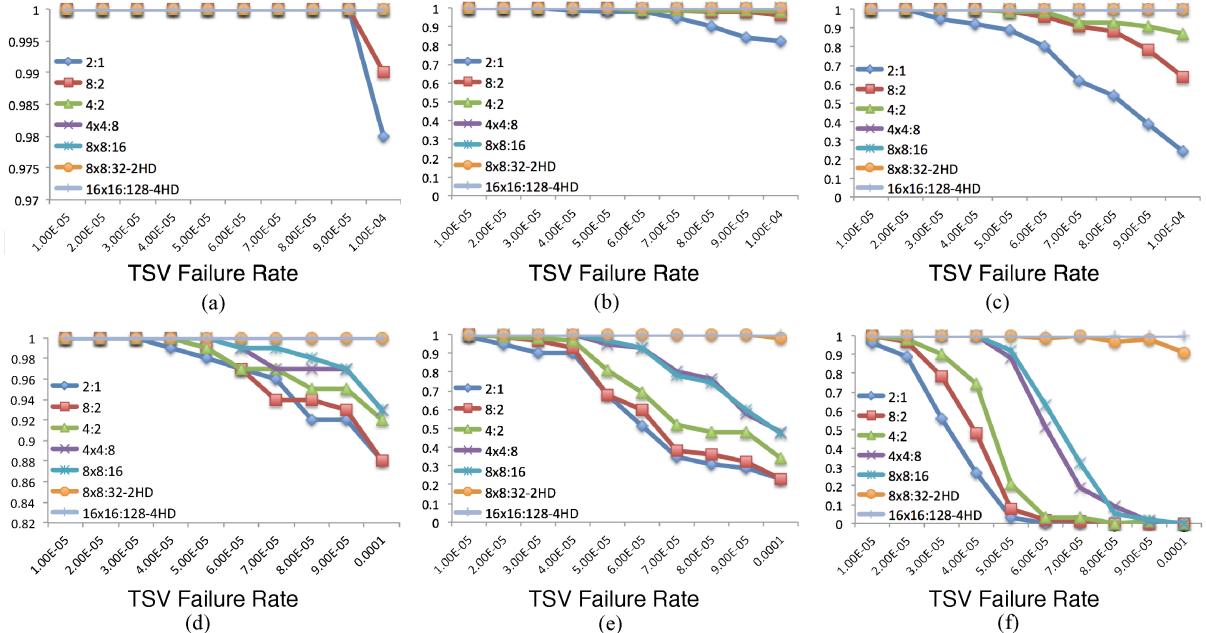


Fig. 15. Repair schemes varying TSV failure rate with fixed alpha. (a) 1K TSV, $\text{ALPHA} = 2$. (b) 16K TSV, $\text{ALPHA} = 2$. (c) 128K TSV, $\text{ALPHA} = 2$. (d) 1K TSV, $\text{ALPHA} = 2.8$. (e) 16K TSV, $\text{ALPHA} = 2.8$. (f) 128K TSV, $\text{ALPHA} = 2.8$.

quite high in the state-of-art process technique. However, the TSVs yield in real product and some prototype test chips is not that optimistic due to many reasons, one of which is the clustering faults.

Fig. 14(a)–(c) shows the yield comparison by varying the clustering effect Alpha with a fixed TSV failure rate set to 5×10^{-5} . It can be observed from the figures that with the increasing clustering effect, all the previous repair schemes suffer notable yield loss. The yield drop becomes more severe as the TSV number increases. However, the proposed maximum-flow-based technique has much less vulnerability (at most 5% yield drop in a medium chip while 10% yield drop in a large chip). After the topology mapping, the proposed technique can achieve a near 100% yield. The vulnerability to clustering fault is different across these repair schemes. To be specific, signal shifting scheme 2 : 1 is worse than others because it fails whenever two adjacent TSVs have faults at the same time. The signal switching and cross-bar schemes behave better; however, their efficiency drops fast as the number of TSVs increases. That is because there are more chances to occur a triple TSV fault nearby.

Fig. 15(a)–(c) shows the yield comparison by varying the TSV failure rate with a fixing $\text{Alpha}=2$. With the ramping TSV failure rate, the yields of baseline repair schemes drop with different gradients while the maximum-flow-based scheme can still keep a very high yield. To be specific, the repair efficiency of a 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 the signal-switching scheme. The signal-switching scheme performs better for small and medium chips with up to 98% yield, but in the case of large chips, the yield drops nearly 10%. To observe closer into the clustering effect, we ramp up the Alpha to 2.8 indicating a high likelihood of catastrophic defects [see Fig. 15(d) and (e)]. As the TSV number increases, all the repair schemes are revolved into three groups. The baseline solutions are converged in the first group with significant yield drop. While the proposed maximum-flow based technique is much better to tolerate TSV clustered faults, it still suffers when TSV number is large enough for catastrophic defects to happen. After topology mapping, the repair efficiency of the proposed technique significantly increases. Compared to the

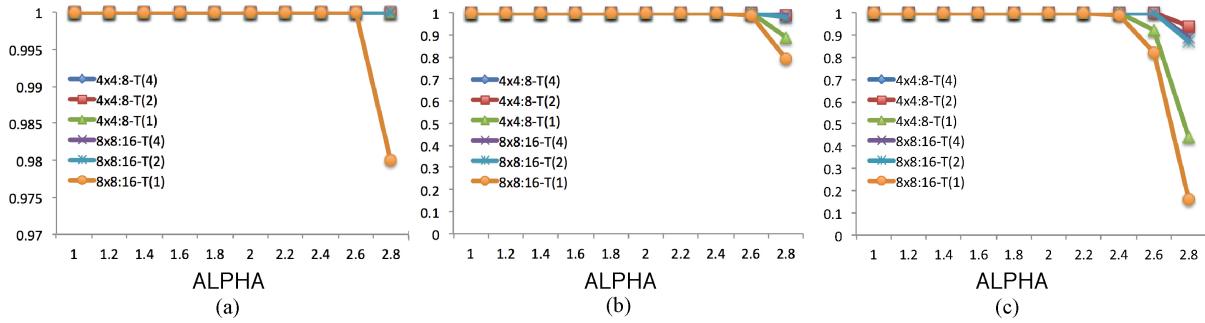


Fig. 16. Time constrained repair varying alpha from 1 to 3. (a) 1K TSV, failure rate = 5×10^{-5} . (b) 16K TSV, failure rate = 5×10^{-5} . (c) 128K TSV, failure rate = 5×10^{-5} .

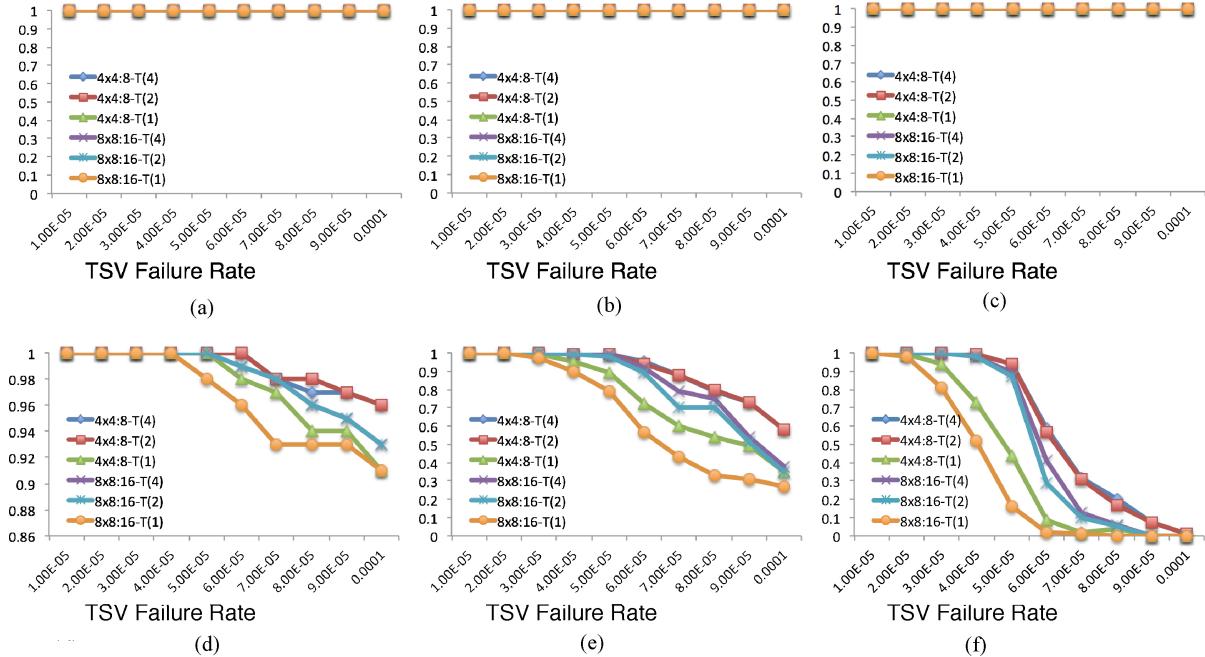


Fig. 17. Time constrained repair varying TSV failure rate with fixed alpha. (a) 1K TSV, $\text{ALPHA} = 2$. (b) 16K TSV, $\text{ALPHA} = 2$. (c) 128K TSV, $\text{ALPHA} = 2$. (d) 1K TSV, $\text{ALPHA} = 2.8$. (e) 16K TSV, $\text{ALPHA} = 2.8$. (f) 128K TSV, $\text{ALPHA} = 2.8$.

two topology mapping strategies, the one with 4-hop distance can keep the yield closer to 100%. Obviously, the 4-hop distance mapping makes the topology neighbor distant and tolerate faults with larger cluster size.

From the above results, we can observe that the redundancy ratio is not the only dominating factor for the final yield, e.g., the 8 : 2 scheme is better than the 2 : 1 scheme, even though its redundancy ratio is only half of the latter one. This is because the flexibility for repair has a significant impact on TSV repair efficiency. Let us consider a TSV bundle containing eight signal TSVs. As long as there are no more than two TSV failures, the cross-bar 8:2 scheme can successfully repair them. However, if two TSV failures occur in the same repair unit, the 2:1 signal shifting scheme would fail. When the TSV failure rate is not very high, the possibility of having more than two TSV failures in the bundle (and can be repaired with the 2:1 repair scheme) is lower than the possibility of having irreparable double TSV failures. Therefore, the 8:2 repair scheme results in a higher yield when compared to the 2:1 scheme. Similarly, the 8 × 8 : 16 TSV grid has a higher

yield than the 4 × 4 : 8 TSV grid when the cluster effect is strong. This is also expected since the 4 × 4 TSV grid is more vulnerable to the fault cluster whose area may cover the majority of the 4 × 4 TSV grid or be even larger than the TSV grid. Under this situation, it is by no means possible to repair such a fault cluster.

To evaluate the timing effect among the proposed bounded search heuristic, we reproduce the above comparison among different length constraints. Fig. 16(a)–(c) shows the yield comparison by varying the clustering effect Alpha with a fixed TSV failure rate equal to 5×10^{-5} . Both 4 × 4 and 8 × 8 TSV grids with 1-hop length constraint ($T(1)$) suffer from notable yield loss. This is expected because the repair path can only reach one hop of TSVs for repair and hence it is vulnerable to any TSV faults in close proximity.

Fig. 17(a)–(c) shows the yield comparison by varying the TSV failure rate with a fixing $\text{Alpha} = 2$. With the increase of TSV failure rate, only the proposed repair scheme with a 1-hop length constraint suffers the yield drops. After we raise the clustering effect to 2.8 [see Fig. 17(d) and (e)], the pro-

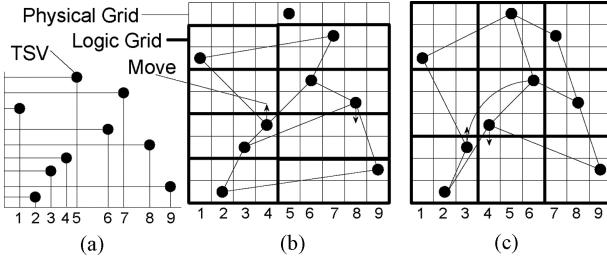


Fig. 18. Mapping irregular placed TSVs. (a) Sort by index. (b) Map to 4×2 grid. (c) Map to 3×3 grid.

posed repair scheme under all pre-defined length constraints suffers the yield drop in different gradients. For the bounded search algorithm, the 4×4 scheme is always better than the 8×8 , which is different from the maximum-flow-based method. That is because, the search bound limits the flexibility so that some fault maps, repairable in maximum-flow based method, become irreparable using bound search algorithm. An interesting observation is that increasing the search bound only improves the yield in a limited degree (see $T(4)$ and $T(2)$). As long as the more-than-1-hop repair path is allowed, the proposed timing aware repair scheme can guarantee a very high yield under our experimental settings. This observation also supports the advantage of topology mapping. Suppose the signal can reroute to four hops away in the original TSV grid, it can only reroute to two hops away after a 2-hop distance topology mapping. According to the above observation, as long as the timing constraint is larger than the 2-hop distance, it is worthwhile applying the topology mapping.

VII. DISCUSSION

A. TSV Grid Construction

TSVs are usually bundled together in a 3-D 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 a high aspect ratio (e.g., 1×32), we can first merge two columns into one column and the cluster becomes the 2×16 grid. Such a 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. 18(a)], which indicates their relative positions amongst TSVs. Next, we construct grids and

TABLE II
COST COMPARISON FOR 1K TSVs

1k TSVs	4:2	8:2	$4 \times 4:8$	$8 \times 8:16$	$16 \times 16:32$
#Spare TSVs	512	256	512	256	128
#Extra Muxes	1k	2k	3k	3k	3k
Area (μm^2)	52010	27220	53630	28030	15230

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. 18(b) and 3×3 in Fig. 18(c) in 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). Table II shows a simple comparison of cost assuming 1k TSVs. As calculated above, the size of each switch is approximately $0.9 \mu\text{m} \times 0.9 \mu\text{m}$, which is negligible compared to a TSV ($10 \mu\text{m} \times 10 \mu\text{m}$). As shown in Table II, the dominant factor of area is the TSV instead of the extra logic gates. More importantly, the TSV manufacturing cost is much larger than logic gates and the spare TSVs also suffer the yield problem. From this perspective, the proposed repair scheme requires much less redundant TSVs than existing solutions, rendering less hardware overhead, while achieving a higher yield at the same time.

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.

VIII. CONCLUSION

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

REFERENCES

- [1] V. Suntharalingam, R. Berger, J. A. Burns, C. K. Chen, C. L. Keast, J. M. Knecht, R. D. Lambert, K. L. Newcomb, D. M. O'Mara, D. D. Rathman, D. C. Shaver, A. M. Soares, C. N. Stevenson, B. M. Tyrrell, K. Warner, B. D. Wheeler, D.-R. W. Yost, and D. J. Young, "Megapixel CMOS image sensor fabricated in three-dimensional integrated circuit technology," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2005, pp. 356–357.
- [2] H. Yoshikawa, A. Kawasaki, Tomoaki, Iiduka, Y. Nishimura, K. Tanida, K. Akiyama, M. Sekiguchi, M. Matsuo, S. Fukuchi, and K. Takahashi, "Chip scale camera module (CSCM) using through-silicon-via (TSV)," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2009, pp. 476–477.
- [3] U. Kang, H.-J. Chung, S. Heo, D.-H. Park, H. Lee, J. H. Kim, S.-H. Ahn, S.-H. Cha, J. Ahn, D. Kwon, J.-W. Lee, H.-S. Joo, W.-S. Kim, D. H. Jang, N. S. Kim, J.-H. Choi, T.-G. Chung, J.-H. Yoo, J. S. Choi, C. Kim, and Y.-H. Jun, "8 Gb 3-D DDR3 DRAM using through-silicon-via technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, Jan. 2010.
- [4] T. Mitsuhashi, Y. Egawa, O. Kato, Y. Saeki, H. Kikuchi, S. Uchiyama, K. Shibata, J. Yamada, M. Ishino, H. Ikeda, N. Takahashi, Y. Kurita, M. Komuro, S. Matsui, and M. Kawano, "Development of 3D-packaging process technology for stacked memory chips," in *Proc. MRS*, vol. 970, no. 1, 2006.
- [5] M. Kawano, S. Uchiyama, Y. Egawa, N. Takahashi, Y. Kurita, K. Soejima, M. Komuro, S. Matsui, K. Shibata, J. Yamada, M. Ishino, H. Ikeda, Y. Saeki, O. Kato, H. Kikuchi, and T. Mitsuhashi, "A 3-D packaging technology for 4 Gbit stacked DRAM with 3 Gbps data transfer," in *Proc. IEEE IEDM*, Dec. 2006, pp. 1–4.
- [6] G. H. Loh, "3D-stacked memory architectures for multicore processors," in *Proc. Int. Symp. Comput. Architecture*, 2008, pp. 453–464.
- [7] D. H. Woo, 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 *Proc. IEEE Int. Symp. High Performance Comput. Architecture*, Jan. 2010, pp. 1–12.
- [8] T. Zhang, K. Wang, Y. Feng, X. Song, L. Duan, Y. Xie, X. Cheng, and Y.-L. Lin, "A customized design of DRAM controller for on-chip 3-D DRAM stacking," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2010, pp. 1–4.
- [9] X. Y. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3-D ICs)," in *Proc. IEEE Asia South Pacific Des. Autom. Conf.*, Jan. 2009, pp. 234–241.
- [10] G. Smith, L. Smith, S. Hosali, and S. Arkalgud, "Yield considerations in the choice of 3-D technology," in *Proc. Int. Symp. Semicond. Manuf.*, Oct. 2007, pp. 1–3.
- [11] Q. Xu, L. Jiang, H. Li, and B. Eklow, "Yield enhancement for 3D-stacked ICs: Recent advances and challenges," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2012, pp. 731–737.
- [12] H.-H. S. Lee, and K. Chakrabarty, "Test challenges for 3-D integrated circuits," *IEEE Des. Test Comput.*, vol. 26, no. 5, pp. 26–35, Sep.–Oct. 2009.
- [13] L. Jiang, R. Ye, and Q. Xu, "Yield enhancement for 3D-stacked memory by redundancy sharing across dies," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2010, pp. 230–234.
- [14] C. W. Chou, Y. J. Huang, and J. F. Li, "Yield-enhancement techniques for 3-D random access memories," in *Proc. Int. Symp. VLSI Des. Autom. Test*, 2010, pp. 104–107.
- [15] C. Ferri, S. Reda, and R. I. Bahar, "Strategies for improving the parametric yield and profits of 3-D ICs," in *Proc. Int. Conf. Comput.-Aided Des.*, 2007, pp. 220–226.
- [16] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, C. S. Patel, R. J. Polastrone, K. Sakuma, R. Sirdeshmukh, E. J. Sproglis, S. M. Sri-Jayantha, A. M. Stephens, A. W. Topol, C. K. Tsang, B. C. Webb, and S. L. Wright, "Three-dimensional silicon integration," *IBM J. Res. Dev.*, vol. 52, no. 6, pp. 553–569, 2008.
- [17] I. Loi, S. Mitra, T. H. Lee, S. Fujita, and L. Benini, "A low-overhead fault tolerance scheme for TSV-based 3-D network on chip links," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2008, pp. 598–602.
- [18] A.-C. Hsieh, T.-T. Hwang, M.-T. Chang, M.-H. Tsai, C.-M. Tseng, and H.-C. Li, "TSV redundancy: Architecture and design issues in 3-D IC," in *Proc. Des. Autom. Test Eur. Conf. Exhibit.*, 2010, pp. 166–171.
- [19] B. Kim, C. Sharbono, T. Ritzdorf, and D. Schmauch, "Factors affecting copper filling process within high aspect ratio deep vias for 3-D chip stacking," in *Proc. Electron. Compon. Technol. Conf.*, 2006, p. 6.
- [20] A. P. Karmarkar, X. Xu, and V. Moroz, "Performance and reliability analysis of 3D-integration structures employing through silicon via (TSV)," in *Proc. IEEE Int. Reliab. Phys. Symp.*, Apr. 2009, pp. 682–687.
- [21] E. J. Marinissen and Y. Zorian, "Testing 3-D chips containing through-silicon vias," in *Proc. IEEE Int. Test Conf.*, Nov. 2009, pp. 1–11.
- [22] A. W. Topol, D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Jeong, "Three-dimensional integrated circuits," *IBM J. Res. Develop.*, vol. 50, nos. 4–5, pp. 491–506, Jul.–Sep. 2006.
- [23] Y. Zhao, S. Khursheed, and B. M. Al-Hashimi, "Cost-effective TSV grouping for yield improvement of 3D-ICs," in *Proc. Asian Test Conf.*, 2011, pp. 201–206.
- [24] L. Jiang, Q. Xu and B. Eklow, "On effective TSV repair for 3D-stacked ICs," in *Proc. Des. Autom. Test Eur. Conf. Exhibit.*, 2012, pp. 6–11.
- [25] 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. Conf. Wafer Scale Integr.*, 1990, pp. 243–250.
- [26] L. Jiang, L. Huang, and Q. Xu, "Test architecture design and optimization for three-dimensional SoCs," in *Proc. Des. Autom. Test Eur. Conf. Exhibit.*, 2009, pp. 220–225.
- [27] A. V. Goldberg and S. Rao, "Beyond the flow decomposition barrier," *J. ACM*, vol. 45, no. 5, pp. 783–797, 1998.
- [28] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, and M. Skutella, "Length-bounded cuts and flows," *ACM Trans. Algorithms*, vol. 7, no. 1, pp. 4:1–4:27, 2010.
- [29] I. Koren and Z. Koren, "Defect tolerance in VLSI circuits: Techniques and yield analysis," *Proc. IEEE*, vol. 86, no. 9, pp. 1819–1838, Sep. 1998.
- [30] F. J. Meyer and D. K. Pradhan, "Modeling defect spatial distribution," *IEEE Trans. Comput.*, vol. 38, no. 4, pp. 538–546, Apr. 1989.
- [31] M. B. Tahoori, "Defects, yield, and design in sublithographic nanoelectronics," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2005, pp. 3–11.
- [32] J. Koren and Z. Koren, "Defect tolerance in VLSI circuits: Techniques and yield analysis," *Proc. IEEE*, vol. 86, no. 9, pp. 1819–1838, Sep. 1998.
- [33] N. Miyakawa, T. Maebashi, N. Nakamura, S. Nakayama, E. Hashimoto, and S. Toyoda, "New multilayer stacking technology and trial manufacture," in *Proc. 3-D Architectures Semicond. Integr. Packag.*, 2007.
- [34] N. Miyakawa, "A 3-D prototyping chip based on a wafer-level stacking technology," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2009, pp. 416–420.
- [35] B. Swinnen, W. Ruythooren, P. De Moor, L. Bogaerts, L. Carbonell, K. De Munck, B. Eyckens, S. Stoukatch, D. S. Tezcan, Z. Tokei, J. Vaes, J. van Aelst, and E. Beyne, "3-D integration by Cu–Cu thermo-compression bonding of extremely thinned bulk-Si die containing 10 μm pitch through-Si vias," in *Proc. IEDM*, 2006, pp. 1–4.
- [36] A. W. Topol, D. C. La Tulipe, L. Shi, S. M. Alam, D. J. Frank, S. E. Steen, J. Vichiconti, D. Posillico, M. Cobb, S. Medd, J. Patel, S. Goma, D. Dimilia, M. T. Robson, E. Duch, M. Farinelli, C. Wang, R. A. Conti, D. M. Canaperi, L. Deligianni, A. Kumar, K. T. Kwietniak, C. D'Emic, J. Ott, A. M. Young, K. W. Guarini, and M. Jeong, "Enabling SOI-based assembly technology for three-dimensional (3-D) integrated circuits (ICs)," in *Proc. IEEE IEDM*, Dec. 2005, pp. 352–355.



Li Jiang (S'08) received the B.S. degree in computer science and technology from Shanghai Jiaotong University, Shanghai, China, in 2007. He is currently pursuing the Ph.D. degree with the Reliable Computer Laboratory, Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, Hong Kong.

His current research interests include very large scale integration testing, fault-tolerant computing, and yield and reliability enhancement techniques in 3-D integrated circuits.



Qiang Xu (M'06) received the B.E. and M.E. degrees in telecommunication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2005.

He has been with the Chinese University of Hong Kong, Shatin, Hong Kong, since 2005, where he is currently an Associate Professor of computer science and engineering and leads the CUHK Reliable Computing Laboratory. His current research interests include fault-tolerant computing and trusted computing. He has authored or co-authored more than 70 technical papers in these areas.

Computing Laboratory. His current research interests include fault-tolerant computing and trusted computing. He has authored or co-authored more than 70 technical papers in these areas.

Dr. Xu was a recipient of the Best Paper Award in the 2004 IEEE/ACM Design, Automation, and Test in Europe Conference (DATE). He has several other papers nominated for Best Paper Award at prestigious conferences such as ICCAD and DATE. He is currently an Associate Editor for the IEEE DESIGN AND TEST OF COMPUTERS.



Bill Eklow (F'11) is currently a Distinguished Engineer with Cisco Systems, Inc., San Jose, CA. His current research interests include component test, board and system test, and correlating board and system level failures to component defects.

Mr. Eklow is the Chair of the IEEE 1149.6 Working Group, and is an Active Member on the IEEE 1149.1 (Boundary-Scan), P1687 (I_JTAG), and P1838 (3-D Test Access) Working Groups. He is also the Sub-Group Leader for the ITRS 3-D Test Sub-Group. He is a Committee Member for several conferences, workshops, and symposia, including the International Test Conference, the Board Test Workshop, the 3-D Test Workshop, the Silicon Debug and Diagnosis Workshop, the VLSI Test Symposium, and the ATE Vision Conference. He is an Eta Kappa Nu Member and an IEEE Golden Core Member.