

On Effective and Efficient In-Field TSV Repair for Stacked 3D ICs*

Li Jiang[†], Fangming Ye[‡], Qiang Xu[†], Krishnendu Chakrabarty[‡], and Bill Eklow[§]

[†]Department of CS&E, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

[‡]Department of ECE, Duke University, Durham, NC

[§]Cisco Systems, San Jose, CA

ABSTRACT

Three-dimensional (3D) integration based on through-silicon-vias (TSVs) is rapidly gaining traction for industry adoption. However, manufacturing processes for TSVs have been shown to introduce new failure mechanisms. In particular, thermo-mechanical stress and electromigration introduce reliability threats for TSVs, e.g., voids and interfacial cracks, which can lead to hard-to-predict timing errors on critical paths with TSVs, thereby resulting in accelerated chip failure in the field. Burn-in for screening latent defects during manufacturing is expensive and its effectiveness for new TSV defect types has yet to be thoroughly characterized. We describe a reconfigurable in-field repair solution that is able to effectively tolerate latent TSV defects through the judicious use of spares. The proposed solution includes a reconfigurable repair architecture that enables spare TSV sharing between TSV grids, and the corresponding in-field repair algorithms. The effectiveness and efficiency of our proposed solution is evaluated using 3D benchmark designs.

1. INTRODUCTION

Three-dimensional integrated circuits (3D ICs) based on through-silicon vias (TSVs) have emerged as one of the most promising solutions to overcome interconnect bottleneck in CMOS scaling [1]. Comparing to planar ICs, 3D ICs offer many advantages, such as smaller footprint, heterogeneous integration capability, shorter interconnects, and higher memory bandwidth. However, TSV fabrication involves several disruptive manufacturing technologies, which leads to new types of defects [2]. These defects are often latent and difficult to screen during manufacturing test, but their impact can be significant during field operation, leading to reduced service life of 3D ICs [3, 4]. Burn-in for screening latent defects during manufacturing is expensive and its effectiveness for new TSV defect types has yet to be thoroughly characterized. Therefore, repair solutions are needed in order to exploit the potential of 3D ICs and facilitate commercialization.

During TSV fabrication, the temperature is first increased for copper electroplating and then brought down to the ambient temperature. Owing to the large difference in coefficients-of-thermal-expansion (CTE) of the copper TSVs and that of the silicon [5], however, tensile stress inevitably appears on the silicon [6]. Such thermal-mechanical stress is likely to cause TSV interfacial cracks (see Fig. 1) that is usually undetectable during manufacturing test [7]. The forces induced by residual stress in the 3D structure cause the crack to grow dur-

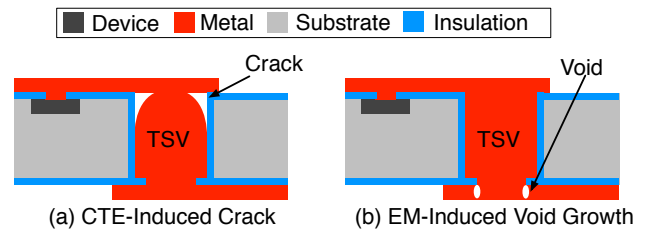


Figure 1: Illustration of some TSV latent defects

ing field operation, thereby increasing the delay of critical paths with TSVs (if any) and eventually forming an open defect [3]. Moreover, a number of recent works examined the classical electromigration (EM) failure mechanisms in 3D ICs, showing that TSVs are prone to EM-induced voiding effects [7–9] (see Fig. 1). Similar to TSV interfacial cracks caused by thermal-mechanical stress, EM-induced voids increase TSV resistance, causing path delay faults and eventually TSV open defects. Note that TSV-induced stress also reduces the reliability of nearby transistors and metal wires, and various analytical models and reliability-driven physical design techniques have been presented in the literature to mitigate this problem [2]. However, we limit the scope of this paper to the repair of TSV latent defects only.

One promising method to tolerate TSV failures is to add spare TSVs in the design for built-in self-repair (BISR). Various TSV redundancy allocation techniques and their corresponding repair algorithms have been proposed in the literature [10–20]. While effective for repairing TSV manufacturing defects occurred at $t = 0$, these solutions are not readily applicable for in-field repair of TSV latent defects that manifest themselves at $t > 0$. This is because the repair solution obtained with the *deterministic* repair algorithms used in these techniques may not satisfy the timing requirement of the circuit due to circuit aging, thereby rendering the repair solution less effective. To tackle the above problem, this paper presents a novel in-field TSV repair solution for stacked 3D ICs. The contributions of this work include the following:

- To the best of our knowledge, we present the first in-field TSV repair framework for 3D IC lifetime reliability enhancement.
- We propose an efficient TSV repair algorithm that is able to significantly improve the mean-time-to-failure (MTTF) of TSV grids through the judicious use of spares, as demonstrated by our experimental results.
- We enhance the TSV redundancy architecture in [18] by allowing redundancy sharing across neighboring TSV grids.

The remainder of this paper is organized as follows. Section 2 presents related works and further motivates this paper. In Section 3 and Section 4, we detail the proposed in-field TSV repair framework and the corresponding repair algorithm, respectively. Experimental results on 3D benchmark designs are next presented in Section 5. Finally, Section 6 concludes this paper.

*This work was supported in part by a research grant from Cisco Systems. The work of F. Ye and K. Chakrabarty was also supported in part by the National Science Foundation (NSF) under grant no. CCF-1017391, and by the Semiconductor Research Corporation (SRC) under contract no. 2118.001.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC '13, May 29 - June 07 2013, Austin, TX, USA

Copyright 2013 ACM 978-1-4503-2071-9/13/05 ...\$15.00.

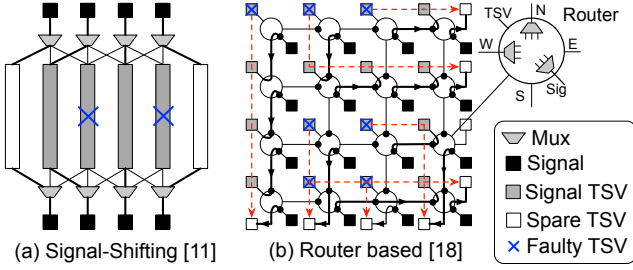


Figure 2: Existing TSV Repair Solutions

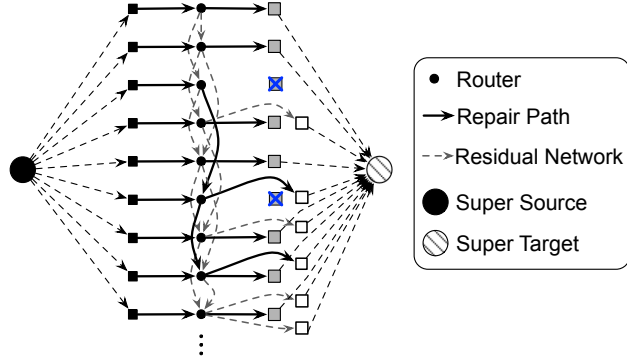


Figure 3: Maximum Flow Based Repair Algorithm.

2. PRELIMINARIES AND MOTIVATION

In this section, we first review related prior work on TSV repair and defect tolerance to increase manufacturing yield. Following that, we discuss the need for explicitly targeting TSV latent defects in the development of a repair solution that can be used in the field.

2.1 Prior Work

Various TSV repair solutions have been proposed in the literature for manufacturing yield enhancement [10–18, 20]. The repair capability of these solutions vary according to their different redundancy architectures and the corresponding repair algorithms. In [10–13], one or more redundant TSVs are added for a group of TSVs and a defective TSV is swapped with a fault-free one via signal shifting (see Fig. 2(a)). In [14], spare TSV rows are added to a TSV array for repair to reduce the storage requirement of reconfiguration data. The above methods assume uniformly distributed TSV faults and use neighboring TSVs to replace faulty ones, if any. In practice, however, if one TSV is defective during fabrication, it is more likely that its neighboring TSVs are also defective due to clustering [21]. This issue has been considered in [15–17], wherein more practical TSV grouping strategies were proposed to tolerate clustered TSV faults.

Recently, a low-cost and flexible TSV redundancy architecture was proposed in [18, 20], which enables defective TSVs to be replaced with distant spares to tolerate clustered faults and it is shown to have higher repair capability than previous methods. Considering the fact that neighboring TSVs usually suffer from similar thermal and mechanical stress, we leverage this TSV redundancy architecture in our work to enable repair of clustered latent faults within a TSV grid. We next briefly discuss it in the following.

As shown in Fig. 2(b), the proposed architecture links TSVs with switches and wires, leading to a TSV grid. Redundant TSVs are put at two borders of the grid for repair. If one signal is disconnected due to a TSV fault, the switches linking two ends of the faulty TSV reroute the signal through a neighboring fault-free TSV. Since the fault-free TSV is “borrowed” by the previously-rerouted signal, the signal originally linked to it needs to be rerouted as well. This procedure continues until a spare TSV at the boundary is used.

Consequently, the TSV repair problem can be formulated as a problem of finding *edge-disjoint repair paths* for faulty TSVs. Con-

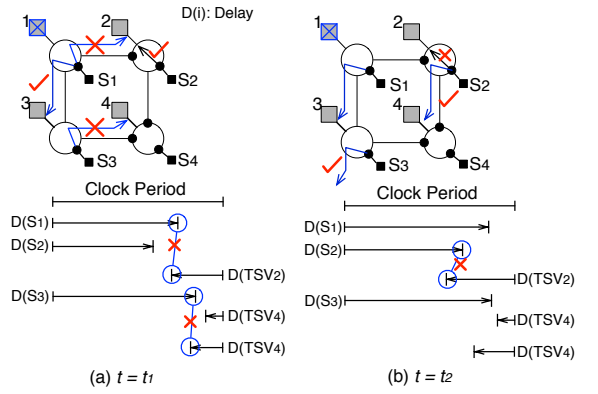


Figure 4: An example to motivate the need for careful in-field repair.

sider the TSV grid as a directed graph, wherein signals, TSVs and routers are represented as vertices, while the directed edges are used to link them. In order to route each signal to a fault-free TSV without any routing conflict, they first assign each edge in the directed graph with a unit capacity “1” to construct a directed flow network. Then, a super source vertex is added to the flow network, pointing to all the vertices that represent signals; while at the other side, all the vertices denoting fault free TSVs are pointing to a super target vertex merging all the fault-free TSVs into a target node (see Fig. 3). The original TSVs repair problem can then be solved using the maximum flow method and the TSV grid is repairable if and only if maximum flow value is equal to the number of signals.

2.2 Motivation for In-Field Repair

Unlike TSV repair at $t = 0$ for yield enhancement, the objective of in-field repair for TSV latent faults at $t > 0$ is to increase the MTTF of 3D ICs. This problem is especially difficult due to circuit aging. Previous TSV repair solutions have focused on the replacement of defective TSVs with fault-free ones, i.e., the repair algorithms start from faulty TSVs and try to find repair paths to spares, without explicitly considering the impact of the repair solution on signal delays. Such repair methodology is generally applicable for detectable manufacturing defects such as opens and shorts when the distance between the failed TSV and its corresponding spare is not large.

However, both TSVs and other circuit elements wear out during field-operation. On one hand, it is likely that the “replacement-oriented” repair solution provided with existing methods violates signal timing requirements after shifting or rerouting, thereby leading to new “faulty TSVs”. On the other hand, a faulty TSV linking to a particular signal might be a good one if it links to another signal instead. This is because, a TSV fault occurring online is not necessarily a catastrophic open/short defect, but often a delay fault that cannot meet the timing requirement of critical paths going through it due to circuit degradation. Consider the example TSV grid shown in Fig. 4(a). Signal S_1 needs to be rerouted due to the latent defect that is manifested on its corresponding TSV. However, it may fail again if it is rerouted to use TSV_2 originally linked to S_2 , generating a “new” TSV fault even though this TSV is fault-free. Such fault propagation may eventually make the TSV grid irreparable, even though a more sustainable repair solution exists as shown in Fig. 4(b)).

Consequently, for in-field TSV repair, we should not focus only on faulty TSV replacement and simply find a repair path for each faulty TSV. Instead, we are to find the set of signal-TSV pairs that satisfy the timing requirement of every signal. Whether a signal and a particular TSV can be paired together is known only after we conduct online testing of those circuit paths going through the TSV, due to the difficulty to predict change of signal timing slacks with circuit aging. The above considerations have motivated the new in-field repair technique investigated in this paper.

3. IN-FIELD TSV REPAIR FRAMEWORK

In order to conduct in-field repair for TSV latent defects, we first need to be able to test and diagnose faulty TSVs in an online manner. To achieve these objectives, as in [22], we assume the existence of a processor core and non-volatile memory in the system for test and diagnosis purpose (see the conceptual architecture shown in Fig. 5). This assumption is reasonable because 3D logic-on-logic ICs or 3D logic-memory designs of the near future are likely to be large multi-processor system-on-a-chip (MPSoC) designs. Such designs provide the most compelling motivation for high-density 3D integration. To be specific, the non-volatile memory stores the test and diagnosis patterns for TSV faults, our in-field repair algorithm and the repair signature for each TSV grid, while a processor core is called upon for online test and repair, triggered periodically or by events.

3.1 Online Test and Diagnosis

As discussed earlier, TSVs suffer from interfacial cracks and EM-induced voids and such latent defects usually manifest themselves as hard-to-predict timing errors on critical paths with TSVs. From this perspective, TSV BIST techniques (e.g., [23]) are insufficient for in-field test and diagnosis because they target on faults occurred in individual TSV structure (and often consider TSV open/short only) instead of delay faults of circuit paths with TSVs. For example, as discussed in Section 2.2, using a fault-free TSV to replace a faulty one does not necessarily lead to a valid repair solution because of the unknown signal timing slack changes with circuit aging.

Consequently, it is important to online test those critical paths that go through TSVs. To be specific, for each TSV, we need to pick one or more long paths that go through it and store the corresponding path delay test patterns in non-volatile memory (in a compressed form to reduce the storage requirement, whenever possible).

Note that, we try to overcome the delay fault on a particular path with TSVs by signal rerouting using other TSVs. Even though this strategy mainly targets TSV degradation/failure, it can also be used to target for path delay faults caused by the degradation of other on-path circuit elements. That is, as long as the identified repair solution is confirmed to be valid with online testing, it is not necessary to root-cause the path delay fault to a particular circuit element.

3.2 Spare TSV Sharing and Reconfiguration

Due to the clustering effects of latent faults, unless the redundancy ratio is quite high, we may still run into the situation that some faulty TSV grids lack spare TSVs while the others have many redundant TSVs. We therefore propose to enhance the TSV redundancy architecture presented in [18] by allowing spare TSV sharing between TSV grids, as shown in Fig. 5.

Given the above TSV redundancy architecture for a 3D IC, the design flow of the proposed in-field repair solution is as follows. With online testing triggered periodically or by events, if a particular path with TSVs is found to be faulty, our TSV repair algorithm (detailed in Section 4) is called upon to obtain a possible repair solution. Afterwards, we rerun online testing to check whether this solution is acceptable. The above procedure iterates until a valid repair solution is achieved. The 3D IC is regarded as being irreparable if the circuit is not free of path delay faults after all the possible repair solutions have been considered.

4. PROPOSED REPAIR ALGORITHM

In this section, we first formulate the in-field repair problem and then present details of the proposed repair algorithm.

4.1 Problem Formulation

For a 3D IC with TSV redundancy architecture as shown in Fig. 5, the in-field TSV repair problem is formulated as follows:

Given the set of signals $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ and the set of TSVs $\mathbf{T} = \{TSV_1, TSV_2, \dots, TSV_m\}$ ($n < m$), our goal is to link every signal in \mathbf{S} with a dedicated TSV in \mathbf{T} under the following conditions: (i)

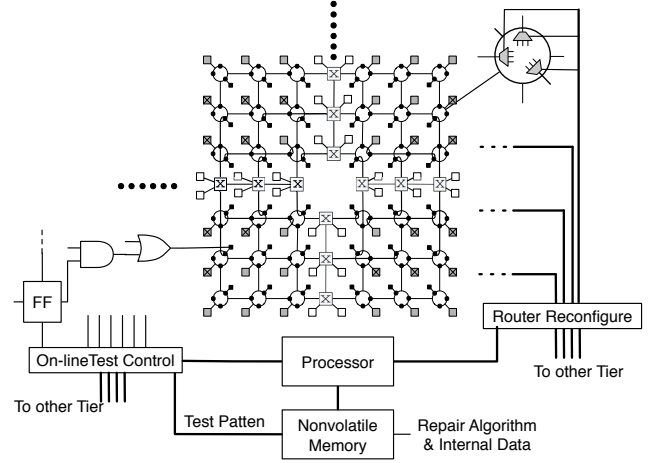


Figure 5: Illustration of the TSV redundancy architecture.

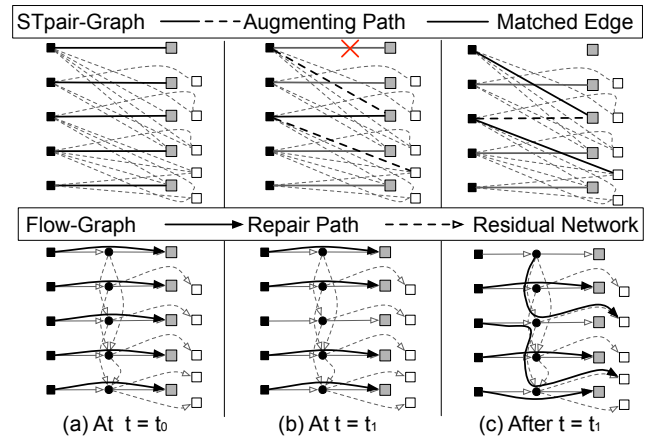


Figure 6: Illustration of the repair algorithm.

all signal-TSV pairs are routable with the given TSV redundancy architecture; (ii) it is confirmed with online testing with no timing violations.

As we need to invoke the online testing procedure whenever there is a possible repair solution, it is preferable to reduce the number of trials for valid repair.

4.2 In-field Repair Algorithm

In order to solve the above problem, we construct a bipartite graph to store all “possible” signal-TSV pairs, namely $STpair-graph$ in this paper. Fig. 6 (a) presents an example $STpair-graph$ at $t = 0$. In this graph, one side is the signal set \mathbf{S} while the other side is the TSV set \mathbf{T} , and an edge exists for a possible signal-TSV pair that has the following two properties: (i). there is at least one routing path from the signal to the TSV in the flow graph; (ii). there is no *confirmed* timing violation for this signal-TSV pair. $STpair-graph$ gets updated with online testing results, i.e., an edge is deleted if the corresponding signal-TSV pair fails path delay test, and a TSV and all its edges are removed when it has a catastrophic failure, e.g., a full open defect.

A valid repair solution is hence a *maximum matching* of the $STpair-graph$ whose matching number is equal to n (i.e., every signal is paired with a dedicated TSV) and all of the signal-TSV pairs are both routable and confirmed to have no timing violations with online testing. With continuous circuit aging, one can imagine that the number of edges in $STpair-graph$ keeps decreasing, and the 3D IC is irreparable when the matching number of a $STpair-graph$ is less than n .

We use Fig. 6 to illustrate one possible repair algorithm. Fig. 6(a) presents the matching used in the 3D IC at $t = 0$. Suppose online

testing is performed at $t = t_1$, and we found one signal-TSV pair fails its test. We remove this edge from *STpair-graph* and thus the current matching is not maximum any more. In order to find another maximum matching, we resort to Berge’s lemma [25], by iteratively finding the shortest *augmenting path*¹ from the unmatched signal to any free TSV. Such a method preserves the signal-TSV pairs in the earlier matching whenever possible and hence is more likely to be valid when compared to a solution based on a random maximum matching of the updated *STpair-graph*. In addition, routability checking is integrated into the above procedure for efficiency. That is, whenever we add an augmented path, we update the corresponding flow graph and check whether it can be routed in the residual network of the flow graph. To update the flow graph, we cancel the edges in the flow graph possessed by those signal-TSV pairs that are removed in the matching and move them to the residual graph (i.e., the sub-graph of the original one, composing edges with residual capacity) (Fig. 6(b)). Then, we can verify routability by finding edge-disjoint paths in the residual network for those signal-TSV pairs that are added into the new matching (Fig. 6(c)). If the matching solution is not routable, we find another augmenting path and iterate the above procedure. Otherwise, we invoke online testing to check whether this solution leads to any timing violation. If not, we have obtained a valid repair solution. Otherwise, we update *STpair-graph* by removing those edges whose corresponding signal-TSV pairs fail path delay tests, and repeat the above procedure on the updated *STpair-graph*.

While simple and effective, the above algorithm may invoke online testing many times due to the enumeration of matchings. Let us use M_i to denote the i_{th} maximum matching of the *STpair-graph* (containing the set of all signal-TSV pairs). The above repair algorithm iteratively finds a new maximum matching and performs online testing for it, until a matching (say, M_v) is shown to be valid. Hence, we need to perform v times of online testing. Generally speaking, however, there is usually a significant overlap of the signal-TSV pairs between M_i and M_{i+1} because we tend to preserve many existing valid signal-TSV pairs from the previous solution in each iteration. These preserved pairs are known to be fault-free with previous testing results, which do not need to be tested again.

Motivated by the above discussion, we propose a more efficient algorithm. Instead of checking one possible matching a time with online testing, we attempt to test “multiple matchings” simultaneously, whenever possible. For example, after testing M_0 , for a new maximum matching M_1 , we only need to perform online testing for those signal-TSV pairs in $M_1 \setminus M_0$, because the other signal-TSV pairs in $M_1 \cap M_0$ have been shown to be valid with previous testing results of M_0 . Without loss of generality, let us consider another maximum matching M_2 (if any), there must be some signal-TSV pairs in $M_2 \setminus M_1$ (otherwise M_2 is not a new matching). If some of these signal-TSV pairs have not been tested (i.e., they do not belong to M_0) and they are routable together with those signal-TSV pairs in $M_1 \setminus M_0$, they can be tested simultaneously in one iteration. Such a method reduces the number of online testing because, if a signal-TSV pair in $M_2 \setminus M_1$ is shown to be invalid, not only we do not need to test M_2 any more, but also the corresponding edge is removed from the *STpair-graph* and reduces the possibility to find other invalid matchings.

4.3 Impact of TSV Redundancy Sharing

With TSV redundancy sharing between neighboring TSV grids, there might be conflict between the repair requirements between them. We use the example shown in Fig. 7 to explain how we resolve this issue. In this example, TSV Grid A and TSV Grid B are sharing spare TSVs in Fig. 7(a)). In this case, for both grids, their *STpair-graphs* contain the shared TSVs. Hence, the two *STpair-graphs* are connected as shown in Fig. 7(b). We still obtain maximum match-

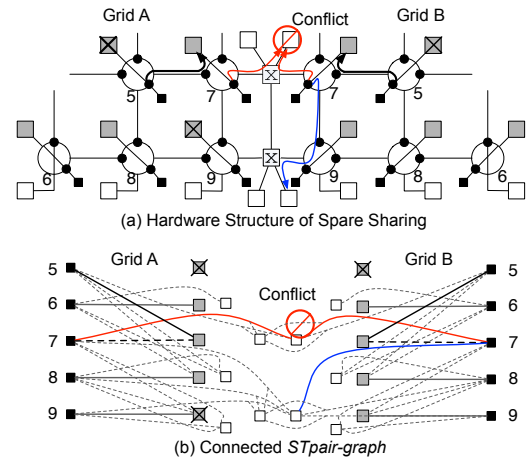


Figure 7: The impact of TSV redundancy sharing on repair algorithm.

ings for each grid by finding augmenting paths. When the two grids try to use the same spare TSV with their augmenting paths, a conflict arises (see red line in Fig. 7(b)). We then arbitrate which grid owns this spare TSV according to the fault maps of the two grids. In this example, Grid A has more faults than Grid B and hence this spare TSV will be assigned to Grid A. We remove this node from the *STpair-graph* of Grid B and it will look for a different matching for in-field repair.

5. EXPERIMENTAL RESULTS

5.1 Experimental Setup

To evaluate the effectiveness and efficiency of the proposed solution, we perform simulation studies and report results on MTTF and test times.

We use the maximum-flow based algorithm presented in [18] as the baseline solution for comparison. As [18] mainly deals with manufacturing defects and the original algorithm uses a static fault map as input, we make the following changes to generate two types of baseline in-field repair algorithms. The first type simply updates the fault map by marking the corresponding TSV to be “faulty” whenever online testing shows an invalid signal-TSV repair and utilizes the original algorithm to find a new repair solution (if possible), denoted as MF . For the second type, when a signal-TSV pair is shown to be invalid with online testing, it attempts to find another repair path for the faulty TSV instead of marking the TSV as “faulty”, denoted as MF' . The proposed algorithm based on maximum matching with routability verification is denoted as MV , while the proposed repair algorithm with test time reduction is named as MR . The above results are obtained based on the TSV redundancy architecture in [18]. We further present the results based on the proposed TSV redundancy architecture with spare sharing capability, denoted as MS . We compare the MTTF of the above solutions, and a particular 3D IC is deemed to fail when no repair solution can be found for a path delay fault due to aging effects.

The circuits used in our experiments are the performance-optimized data encryption standard (DES) circuit and the fast-Fourier transform (FFT) circuit from the IWLS 2005 OpenCore benchmarks. The DES circuit contains 26,000 gates and 2,000 flops, while the FFT circuit contains 229,000 gates and 20,000 flops. The DES circuit was partitioned into two-, three-, and four-die stacks using the Nangate open cell library and a placement engine for timing optimization. Given the operational frequency of the benchmark 3D ICs, we extract the timing slacks for paths with TSVs. Due to the lack of reliability models for stress-induced TSV interfacial cracks in the public literature, we form our model based on an EM reliability model for TSVs and vary its parameter to reflect the impact of TSV interfacial cracks [9, 17]. We also consider initial TSV failures due to manufac-

¹An augmenting path of a matching is defined as a path that starts and ends on free (unmatched) vertices, and alternates between edges in & not in the matching.

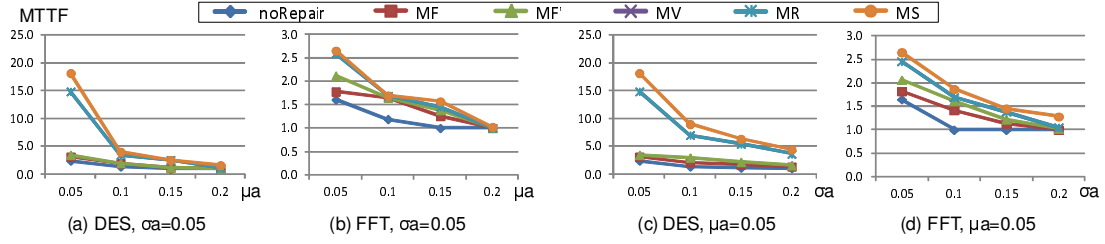


Figure 8: MTTF results in 4×4 TSV grid with varied aging coefficients and fixed Potential Crack or Void Defect Distribution ($0.1 \text{ k}\Omega\text{m}$, $0.1 \text{ k}\Omega\text{m}$).

turing defects. Aging effects are characterized by additional latent delay in TSVs, reflected as resistance increase in terms of time t , calculated as

$$R(t) - R_0 = A \ln\left(\frac{t}{t_0}\right) \quad (1)$$

where A is the slope of TSV degradation on a logarithmic scale, and t_0 is the time when the void becomes larger than their TSV section. Note that A and t_0 are affected by multiple parameters, such as the initial resistance R_0 of TSV, TSV barrier resistivity, TSV dimensions, and possibility of voids generated in TSVs. R_0 varies for different TSVs due to process variation and it is assumed to follow a Gaussian distribution. The parameter A indicates the aging rate, which is related to the workloads applied to the 3D IC which in turn determines the temperature and switching activities for TSVs. The dynamic changing of A due to workloads is aggregated in this paper and we use Gaussian distribution to obtain A .

5.2 Results and Analysis

Fig. 8(a)-(d) presents the normalized MTTF values, compared to the worst case without any redundancy for in-field repair. We have four configurations for aging coefficients with their mean values (μa) and variances (σa) varying between $0.05 \text{ k}\Omega/\log(\text{s})$ to $0.2 \text{ k}\Omega/\log(\text{s})$. This setting mimics the circuits under different stress. The distribution of the initial resistances R_0 of TSVs that represents the potential Crack or Void Defect Distribution are fixed with a mean value (μr) of $0.1 \text{ k}\Omega$ and variance (σr) of $0.1 \text{ k}\Omega$.

First of all, it can be observed that the two proposed repair algorithms with the TSV redundancy architecture in [18] lead to much higher MTTF values when compared to the two baseline solutions. For example, for DES design, MTTFs are 14.8 for both MV and MR with aging coefficient of $(0.05, 0.05)$, compared to 3.1 using MF and 3.5 using MF' . This is because we are able to search a much larger solution space by exploring all possible signal-TSV pairs while previous methods only target on repair path identification for faulty TSVs. The MTTF value of MF' is slightly better than that of MF because the latter solution regards the TSV from an invalid signal-TSV pair as “faulty”, rendering an even smaller solution space. It should be noted that MV and MR have the same MTTF as the solution space are the same for these two algorithms. By adding spare sharing capability with the proposed architecture, the MTTF is increased to 18.2 under the same aging rate.

Secondly, we observe significant MTTF reduction as aging coefficients increase (see Fig. 8(a)-(d)), due to the higher TSV failure probability with increasing aging rates. The differences are even larger for the proposed two repair algorithms (MV and MS), wherein the

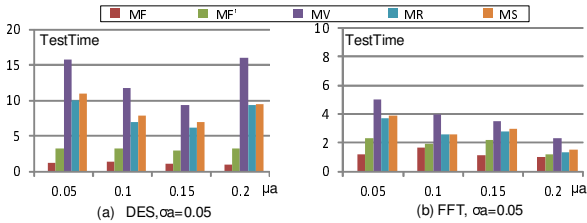


Figure 9: Test Time results in 4×4 TSV grid with varied aging coefficients and fixed Potential Crack or Void Defect Distribution ($0.1 \text{ k}\Omega\text{m}$, $0.1 \text{ k}\Omega\text{m}$).

impetus of downtrends is reduced as the aging coefficient increases. This is expected as the solution space shrinks quickly as aging effect becomes more severe, rendering less repair efficiency for all repair algorithms. This indicates that, even with a better TSV redundancy architecture (with spare TSV redundancy sharing), we cannot achieve high MTTF values when the circuit is under severe aging effects.

Thirdly, we compare the results of DES design in Fig. 8(a)(c) and FFT design in Fig. 8(b)(d). While we can see similar trends for the results of FFT design, but the MTTF differences between the five algorithms are not as significant as that of DES design. This is mainly because the timing slacks of paths with TSVs in FFT design is much tighter, thus leading to less MTTF values.

Fig. 9(a)-(b) describe the corresponding test time (in terms of the number of performed online testing) of the circuit with the five repair methods under various aging coefficients, corresponding to Fig. 8(a)-(b). The proposed algorithms requires more test time compared to the baseline algorithms due to the fact that more on-line tests are conducted to achieve more successful repair. While the MTTF values for MV and MR are the same, the test times of MR are much smaller. This is because, we try to perform online testing for multiple possible matchings at the same time. With such test time reduction scheme, the test times of MS only increase slightly although it has a larger solution space to explore with spare redundancy sharing.

Fig. 10 shows the MTTF values of different repair methods when we vary the initial resistances of TSVs, which demonstrate the impact of undetectable cracks/voids during fabrication on the service life of 3D ICs. Due to space limit, we only report the results of DES circuit. We fix the aging coefficient as $(0.05, 0.05)$, and vary the TSV initial resistances R_0 with its mean values from $0.1 \text{ k}\Omega$ to $0.4 \text{ k}\Omega$ and a fixed variance value in Fig. 10(a). While in Fig. 10(b), we also have four configurations for R_0 with the same variance value of $0.1 \text{ k}\Omega$ but different mean values ranging from $0.1 \text{ k}\Omega$ to $0.4 \text{ k}\Omega$. From this figure, we can observe that the TSV initial resistance has minor impact on the MTTF values, when compared to the aging coefficients changes shown in Fig. 8. This is also expected because TSV voids/cracks that have passed burn-in test, have to grow large enough to affect circuit timing, which is determined more by the aging rates instead of their initial values.

Fig. 11 shows the MTTF values of the proposed repair methods with a 8×8 grid size for the repair architecture. The trends are similar to that in Fig. 8, however, the MTTF values of MR and MS are much larger than that in Fig. 8. The main reason is that, in this experiment the signal rerouting delay is not considered and hence we have a much larger repair solution space to explore with a larger TSV bundle.

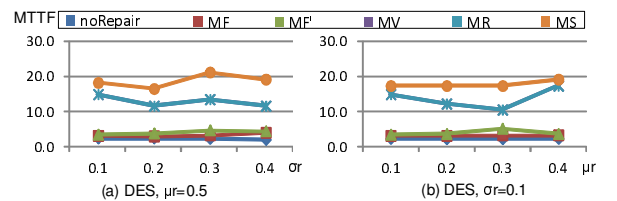


Figure 10: MTTF results for DES in 4×4 TSV grid with varied Potential Crack/void Defect Distribution and fixed aging coefficients ($0.05 \text{ k}\Omega/\log(\text{s})$, $0.05 \text{ k}\Omega/\log(\text{s})$).

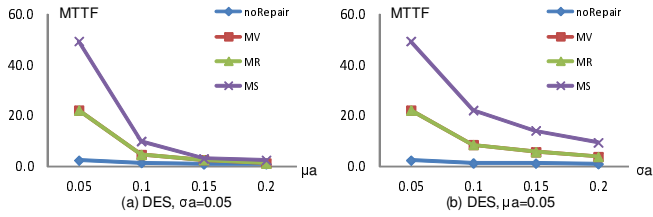


Figure 11: Experimental results in 8×8 TSV grid size repair architecture with varied aging coefficients and fixed Potential Crack/void Defect Distribution (0.1 kΩ, 0.1 kΩ).

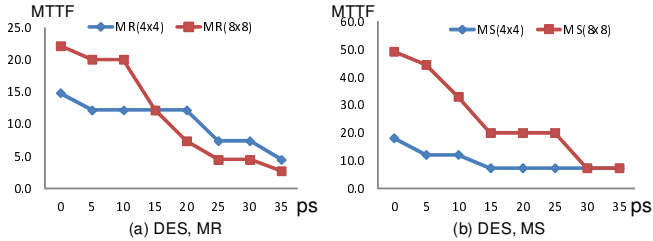


Figure 12: Experimental results with varied rerouting delay between two adjacent routers (ps) and fixed Aging Coefficients and Potential Crack/void Defect Distribution.

It should be noted that, the extra signal rerouting delay is already taken into consideration in the proposed repair architecture even though the previous simulation studies ignore it. As long as the rerouting delay of a signal-TSV pair exceeds the timing slack of the path containing this signal-TSV pair, the on-line test can detect a timing error and discard this pair from the solution space. Fig. 12 investigates the effect of this rerouting delays and shows the MTTF of the proposed repair methods for two different grid size in the repair architecture. For both methods, the architecture with 8×8 grid size performs better when the rerouting delay is small, because it has larger solution space for repair. As rerouting delay increases, the MTTF curves of the two architectures intersect at a point when it has become a bottleneck for signals to be able to reach many TSVs for repair. Beyond this point, the architecture with 4×4 grid size results in more successful repair with higher redundancy ratio. Compared to MR, the intersection point of the two architecture occurs later for MS because the shared redundant TSVs give each TSV grid more solution space to explore.

6. CONCLUSION

TSV-based 3D ICs have emerged as one of the most promising solutions to overcome interconnect bottleneck in CMOS scaling. The disruptive manufacturing process of TSVs, however, introduce new failure mechanisms such as stress-induced interfacial cracks and EM-induced voids. Such reliability threats reduce the service life of 3D ICs. In this paper, we have described a novel in-field repair solution that is able to effectively and efficiently tolerate latent TSV defects through the judicious use of spares. Experimental results on 3D benchmark circuits show that the proposed solution is able to significantly increase MTTF when compared to existing TSV repair techniques.

7. REFERENCES

- [1] International Technology Roadmap for Semiconductors (ITRS'11), available at <http://www.itrs.net/>.
- [2] D. Pan, et al. Design for manufacturability and reliability for TSV-based 3D ICs. In *IEEE/ACM Asia and South Pacific Design Automation Conference*, pages 750–755, 2012.
- [3] A. Karmarkar, X. Xu, and V. Moroz. Performance and reliability analysis of 3D-integration structures employing through silicon via (TSV). In *IEEE International Reliability Physics Symposium*, pages 682–687, april 2009.

- [4] K. N. Tu. Reliability Challenges in 3D IC Packaging Technology. In *Microelectronics Reliability*, pages 517–523, March 2011.
- [5] T. Dao, D. Triyoso, M. Petras, and M. Canonico. Through silicon via stress characterization. In *Proc. IEEE International Conference on IC Design and Technology*, 2009.
- [6] C. Selvanayagam, J. Lau, X. Zhang, S. Seah, K. Vaidyanathan, and T. C. Chai. Nonlinear thermal stress/strain analysis of copper filled TSV and their flip-chip micro-bumps. In *Proc. IEEE Electronic Components and Technology Conference*, pages 1073–1081, 2008.
- [7] S. Ryu, K. Lu, X. Zhang, J. Im, P. Ho, and R. Huang. Impact of near-surface thermal stresses on interfacial reliability of through-silicon vias for 3-D interconnects. *IEEE Transactions on Device and Materials Reliability*, 11(1):35–43, 2011.
- [8] Y. Tan, C. Tan, X. Zhang, T. Chai, and D. Yu. Electromigration performance of through silicon via (TSV) – A modeling approach. *Microelectronics Reliability*, 50(9):1336–1340, 2010.
- [9] T. Frank, C. Chappaz, P. Leduc, L. Arnaud, F. Lorut, S. Moreau, A. Thuaire, R. El Farhane, and L. Anghel. Resistance increase due to electromigration induced depletion under TSV. In *Proc. IEEE International Reliability Physics Symposium*, pages 3F.4.1–3F.4.6, 2011.
- [10] A. Hsieh, T. Hwang, M. Chang, M. Tsai, C. Tseng, and H.-C. Li. TSV redundancy: Architecture and design issues in 3D IC. In *Proc. Design, Automation, and Test in Europe Conference Exhibition*, pages 166–171, march 2010.
- [11] U. Kang, H. Chung, S. Heo, D. Park, H. Lee, J. Kim, S. Ahn, S. Cha, J. Ahn, D. Kwon, et al. 8 GB 3-D DDR3 DRAM using through-silicon-via technology. *IEEE Journal of Solid-State Circuits*, 45(1):111–119, 2010.
- [12] I. Loi, S. Mitra, T. Lee, S. Fujita, and L. Benini. A low-overhead fault tolerance scheme for TSV-based 3D network on chip links. In *Proc. International Conference on Computer-Aided Design*, pages 598–602, nov. 2008.
- [13] M. Nicolaidis, V. Pasca and L. Anghel. Through-silicon-via built-in self-repair for aggressive 3D integration. In *IEEE International On-Line Testing Symposium (IOLTS)*, pages 91–96, 2012.
- [14] Y.-J. Huang and J.-F. Li. Built-In Self-Repair Scheme for the TSVs in 3-D ICs. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(10): 1600–1613, 2012.
- [15] Y. Zhao, S. Khursheed, and B. Al-Hashimi. Cost-Effective TSV Grouping for Yield Improvement of 3D-ICs. In *Proc. IEEE Asian Test Symposium*, 2011.
- [16] J. Xie, Y. Wang, and Y. Xie. Yield-aware time-efficient testing and self-fixing design for TSV-based 3D ICs. In *IEEE/ACM Asia and South Pacific Design Automation Conference*, pages 738–743, 2012.
- [17] F. Ye and K. Chakrabarty. TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation. In *Proc. IEEE/ACM Design Automation Conference*, pages 1024–1030, 2012.
- [18] L. Jiang, Q. Xu, and B. Eklow. On effective TSV repair for 3D-stacked ICs. In *IEEE/ACM Proc. Design, Automation, and Test in Europe*, pages 6–11, 2012.
- [19] L. Jiang, R. Ye, and Q. Xu. Yield enhancement for 3D-stacked memory by redundancy sharing across dies. In *Proc. International Conference on Computer-Aided Design*, pages 230–234, nov. 2010.
- [20] L. Jiang, Q. Xu, and B. Eklow. On effective TSV repair for 3D-stacked ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, to appear.
- [21] G. Van der Plas, P. Limaye, I. Loi, et al. Design issues and considerations for low-cost 3-D TSV IC technology. *IEEE Journal of Solid-State Circuits*, 46(1):293–307, 2011.
- [22] Y. Li, S. Makar, and S. Mitra. CASP: Concurrent autonomous chip self-test using stored test patterns. In *Proc. IEEE/ACM Design, Automation, and Test in Europe Conference and Exhibition*, pages 885–890, 2008.
- [23] Huang, Yu-Jen, et al. A built-in self-test scheme for the post-bond test of TSVs in 3D ICs. In *Proc. IEEE VLSI Test Symposium (VTS)*, pages 20–25, 2011.
- [24] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.
- [25] C. Berge. Two theorems in graph theory. In *National Academy of Sciences of the United States of America*, volume 43 of 9, pages 842–844, 1957.