A novel in-field TSV repair method for latent faults

Ni Tianming¹, Chang Hao², Qi Haochen³, and Huang Zhengfeng^{3a)}

¹ College of Electrical Engineering, Anhui Polytechnic University,
Wuhu 241000, China

² Department of Computer Science and Technology, Anhui University of Finance and
Economics, Bengbu 233030, China

³ School of Electronic Science and Applied Physics, Hefei University of Technology,
Hefei 230009, China

a) huangzhengfeng@139.com

Abstract: The aging effect due to the electromigration (EM) may result in faulty Through-silicon vias (TSVs) and affect the three-dimensional integrated circuits (3D ICs) lifetime. To design a flexible and efficient structure, in this paper, we enhance the region-based design for latent TSV faults, which can adjust the size of the TSV block and TSV redundancy. Experimental results demonstrate that the design can achieve 11.27% and 20.79% reduction of additional delay overhead as compared with router-based design and ring-based scheme, respectively. More importantly, when the target year is 3 years, the design can achieve above 99% reparability for all sizes of TSV blocks. After a given 5-year lifetime, the reparability of the proposed region-based design can still achieve 99% reparability when the TSV size is lower than $16 * 16$, which is the best choice by considering RTSV area and delay overhead.

Keywords: 3D ICs, TSV repair, latent faults

Classification: Integrated circuits

References

- [1] S. Wang, *et al.*: "Recovery-aware proactive TSV repair for electromigration lifetime enhancement in 3-D ICs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **26** (2018) 531 (DOI: [10.1109/TVLSI.2017.2775586](https://doi.org/10.1109/TVLSI.2017.2775586)).
- [2] L. Jiang, *et al.*: "On effective and efficient in-field TSV repair for stacked 3D ICs," IEEE Design Automation Conference (DAC) (2013) 1 (DOI: [10.1145/2463209.2488824](https://doi.org/10.1145/2463209.2488824)).
- [3] Y.-H. Chen, *et al.*: "Architectural evaluations on TSV redundancy for reliability enhancement," IEEE Design, Automation Test in Europe Conference Exhibition (DATE) (2017) 566 (DOI: [10.23919/DATE.2017.7927051](https://doi.org/10.23919/DATE.2017.7927051)).
- [4] L. Jiang, *et al.*: "On effective through-silicon via repair for 3-D-stacked ICs," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **32** (2013) 559 (DOI: [10.1109/TCAD.2012.2228742](https://doi.org/10.1109/TCAD.2012.2228742)).
- [5] W.-H. Lo, *et al.*: "Architecture of ring-based redundant TSV for clustered faults," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **24** (2016) 3437 (DOI: [10.1109/TVLSI.2016.2558514](https://doi.org/10.1109/TVLSI.2016.2558514)).
- [6] T. Ni, *et al.*: "A region-based through-silicon via repair method for clustered faults," IEICE Trans. Electron. **E100-C** (2017) 1108 (DOI: [10.1587/transele](https://doi.org/10.1587/transele)).

- E100.C.1108).
- [7] T. Ni, *et al.*: “Vernier ring based pre-bond through silicon vias test in 3D ICs,” IEICE Electron. Express **14** (2017) 20170590 (DOI: [10.1587/elex.14.20170590](https://doi.org/10.1587/elex.14.20170590)).
  - [8] U. Kang, *et al.*: “8 Gb 3-D DDR3 DRAM using through-silicon-via technology,” IEEE J. Solid-State Circuits **45** (2010) 111 (DOI: [10.1109/JSSC.2009.2034408](https://doi.org/10.1109/JSSC.2009.2034408)).
  - [9] A.-C. Hsieh, *et al.*: “TSV redundancy: Architecture and design issues in 3-D IC,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **20** (2012) 711 (DOI: [10.1109/TVLSI.2011.2107924](https://doi.org/10.1109/TVLSI.2011.2107924)).
  - [10] I. Loi, *et al.*: “A low-overhead fault tolerance scheme for TSV-based 3-D network on chip links,” IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2008) (2008) 598 (DOI: [10.1109/ICCAD.2008.4681638](https://doi.org/10.1109/ICCAD.2008.4681638)).

1 Introduction

The advent of three-dimensional integrated circuit (3D IC) technology has opened up the potential of highly improved circuit designs. Through silicon vias (TSVs) enable the vertical integration of separate dies to form a single 3D chip. The TSV-based 3D stacking technology promises better performances, including smaller footprint, higher bandwidth, lower power and higher interconnect density [1, 2, 3, 4, 5, 6, 7]. However, the reliability of TSVs may become a serious problem due to electromigration (EM) [1, 2, 3], which causes TSVs to have voids during field operation. These voids could increase the delay and resistance of TSVs when the chip is in use, leading to reduced service life of 3D ICs. Therefore, repair solutions are needed in order to exploit the potential of 3D ICs.

One promising method to tolerant TSV failure is to add redundant TSVs in the design for built-in-self-repair (BISR). Several TSV redundancy schemes have been proposed [4, 5, 6, 8, 9, 10]. Kang *et al.* [8] used a signal-switching scheme to improve the yield of 3D DRAM. Hsieh *et al.* [9] proposed a signal-shifting strategy to form a TSV chain with one redundant TSV. Mitra *et al.* [10] proposed a crossbar scheme for 3D NoC links. These methods all use one-dimension shifting of signals, which limits usage of redundant TSVs. Jiang *et al.* [4] presented a router-based TSV redundancy framework. Redundant TSVs are added to one row and column of the TSV grid, and each TSV in the block is placed with a router, which can reroute the signals to neighboring or distant TSVs. However, this design requires many redundant TSVs placed in a row and column of the TSV grid, thus the area overhead cannot be ignored. Lo *et al.* [5] proposed a ring-based TSV redundancy architecture. This technique needs less redundant TSVs compared with the scheme in [4], however the reparability is relatively low. On the other hand, both the schemes in [4] and [5] have long repairing path, causing significant additional delay overhead along with corresponding repairing paths, and it may cause inevitable timing violation because of the long repairing path.

The previous schemes have not considered the latent faults, whose impact can be significant during filed operation [1, 2, 3]. In this paper, reparability means the probability that a chips time-to-failure is longer than the target lifetime. We will

present a novel design and study what the best architecture is for reliability when the size of the TSV grid is different. First, we will present a novel and flexible region-based solution for latent TSV faults repairing. Next, we will evaluate the region-based design and other redundant architectures for different sizes of TSV blocks in terms of reparability and delay overhead.

2 Preliminaries

2.1 Metric for reliability

Mean-time-to-failure (MTTF) is a common measurement for chip lifetime. MTTF represents an average time of which a TSV can reasonably be expected to perform in the field based on specific testing. Several researches are presented for TSV reliability and ways for calculating MTTF. The increased resistance of a TSV due to EM can be shown as follows:

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

R_0 is the initial resistance of TSV, A is the slope of resistance degradation on a logarithmic scale, and t_0 is the time when the void becomes larger than TSV section. For the resistance increase, we follow [3] to use Gaussian distribution to obtain parameters, such as R_0 , t_0 and A . In summary, given R_0 , t_0 and A , we calculate t when the TSV is failed and then average MTTF of the TSV for different parameters.

Although MTTF is a good metric for the lifetime, it is hard to determine the reliability using MTTF simply. When a device is in use, what we care about is not how long the MTTF can be. Instead, we care about the probability of the device that can survive longer than a target lifetime. To realize this concept, we use probability to present reliability. Given a target lifetime, *life*, a metric of reliability, $repair_{life}$, is defined as:

$$repair_{life} = P(MTTF > life) \quad (2)$$

2.2 Additional delay overhead

As mentioned in [5], additional delay introduced during signal rerouting is non-negligible and it may lead to timing violation, which needs to be taken into consideration. To reroute a critical signal, a distant redundant TSV (RTSV) may be used. The additional delay overhead caused by the switch between each TSV and its signal entry can be negligible, and the wire length of rerouting the signal mainly determines the additional delay [4, 5, 6]. In this case, the additional delay overhead depends on the distances of the original signal TSV and RTSV away from the respect signal entries, which roughly equals the distance between the two TSVs. To note that when a signal is on a timing critical path and the repairing distance is remote, it may violate the timing constraint, causing timing violations at the same time. Once a TSV is faulty, then all the signals on the repairing paths have to be shifted to the neighboring TSVs. Shifting process continues until the last signals are shifted to RTSVs, and each shifting process will insert extra delay overhead as mentioned in [4, 5, 6]. The total additional delay overhead should be achieved by

Fig. 1. Overview of region-based TSV redundancy design.

summing all the extra delays during signal shifting. Obviously, it is wise to shorten the repairing length to guarantee relatively high performance. In the following design, for sake of simplicity, the delay overhead can be represented by the length of repairing paths. In addition, for general purpose, we use the topological distance instead of physical wire length to represent the repairing length, and we assume the distance of adjacent TSVs in a grid is 1.

3 Proposed region-based architecture

The previous schemes [4, 5] spend a lot of hardware resources, and the significant additional delay overhead caused by remote repairing paths may lead to timing violations. In order to reduce hardware usage and shorten repairing path length, yet still maintain high faulty TSV reparability for latent faults repairing, we present a novel region-based TSV redundancy architecture.

3.1 Overall region-based TSV redundancy structure

The overall structure is shown in Fig. 1, where there are $M * N$ TSVs in a TSV grid. In our design, the TSV grid is equally divided into four regions known as region I, region II, region III and region IV. Note the red cross section in Fig. 1, which includes two boundaries of each region, and redundant TSVs (RTSVs) can be placed at the center of the grid or any other places in the red cross section. In addition, each signal TSV is allocated with a switch, which links signal to the signal TSV or reroute this signal to neighboring fault-free TSV.

When more redundant TSVs are used, a higher reparability can be achieved. The higher number of redundant TSVs also means more area overhead. Choosing as small of a number of RTSVs as possible while maintain a high reparability is very important. Fig. 2 shows how to calculate the number of redundant TSVs for different sizes of TSV blocks.

As illustrated in Fig. 2, four centers of the TSV grid are designated as RTSVs, and the other RTSVs left can be inserted in any other places in the center cross section. Then, let *RTSV margin* be the number of signal TSVs between two RTSVs. When *RTSV margin* equals 0, the TSVs in the red cross section are all RTSVs. When *RTSV margin* equals to 1, there is one signal TSV between two RTSVs on each side. A lower *RTSV margin* means that there are more RTSVs in the structure and hence more TSV area cost. In the following sections, unless special explained,

Fig. 2. Illustration of the enhanced region-based design.

Fig. 3. Example of region-based TSV redundancy design.

we will detail the case that four RTSVs are placed at the center of a TSV grid with the size 8×8 as an illustration.

The switch design and corresponding rerouting directions depend on the placement of RTSVs. In Fig. 3, RTSVs are placed on the lower right corner of region I. Thus, we constrain signals to reroute from two directions (from west to east or from north to south) in region I. Similarly, in region II, the rerouting directions are from east to west and from north to south; we reroute the signal in region III from west to east or from south to north and the signals in region IV will be rerouted from east to west or from south to north.

Note that the signals also can be rerouted across regions. As shown in Fig. 3, TSV1 and TSV2 are placed in region I, TSV3 and TSV4 are included in region III. For example, TSV1 is faulty (marked with a red X), then the signal connecting TSV1 needs to be shifted, and there are two choices for shifting: one is shifted to TSV2 in the same region by *switch b*; another is rerouted to TSV4 in region III by *switch e*. At the same time, the signals can be rerouted from one RTSV to another RTSV bi-directionally. As Fig. 3 explains, the signal can be shifted from RTSV5 passing through *switch c* and *switch d* to RTSV6. Contrarily, signal also can be rerouted from RTSV6 to RTSV5 with the aid of *switch d* and *switch c*.

Next, we detail the interconnections of switches. As Fig. 4 illustrates, except for the TSVs along the same row or column with RTSVs and the ones at outmost

Fig. 4. Detailed switch design of the TSVs inside each region.

Fig. 5. Detailed switch design of the TSVs at the boundary of each region.

boundary of the TSV grid, each signal TSV in the four regions is connected to three 3-to-1 multiplexers (abbreviated as MUXs) whose inputs are from its own signal and other two neighboring signals. For example, *switch e* is located in region I, since the rerouting direction is either from west to east or from north to south, then the inputs of *switch e* are its own signal, signal from left *switch b*, and one signal from upper *switch a*. Meanwhile, the signals passing through *switch e* can be rerouted to either right *switch c* or lower *switch d*.

On the other hand, the detailed switch design of the TSVs along the same row or column with RTSVs are shown in Fig. 5. Take two switches in Fig. 5 (dashed rectangle), as an example. *Switch d*, *switch e*, *switch f*, *switch g* and *switch h* are located in region III, while *switch a*, *switch b* and *switch c* are placed in region I. Since the rerouting direction in region III is either from west to east or from south to north, then *switch d* consists of three 2-to-1 MUXs, whose inputs are from its own signal and the signal from lower *switch g* in the same region III. And the signals passing through *switch d* go to two directions: one is to right *switch e* in the same region III, another is to upper right *switch b* in region I. For *switch e*, the inputs are its own signal, signal from left *switch d*, signal from lower *switch h*, and signal from upper left *switch a*, thus three 4-to-1 MUXs are included in *switch e* design. Furthermore, the outputs of *switch e* go to right *switch f* in the same region and upper right *switch c* in region I.

4 Evaluation flow

Fig. 6 describes our evaluation flow. To evaluate an architecture, a target lifetime and TSV parameters are given as inputs. Then, we initialize *time* to zero and start the steps. If *time* exceeds the given target lifetime, the case passes. If not, *time* continues and TSV fault patterns are generated from the TSV parameters with

Gaussian distribution. In the step *Maximum flow path check*, we use the maximum flow algorithm to check if the faulty TSVs can be repaired [6]. If the repair fails before the given target lifetime, the case is reported fail. Otherwise, *time* is advanced and the iteration repeats until a pass or fail is reported. In total, 1000 cases are tested to obtain the *reparability*.

Fig. 6. The evaluation flow chart.

In the step *Maximum flow path check*, we check if the faulty TSVs can be repaired by redundant TSVs or not. The TSV grid is represented as a directed graph $G(V, E)$, where V is the set of TSVs and corresponding switches in this grid, and E represents the set of edges connecting two nodes with a unit capacity 1. Since the number of either faulty TSVs or RTSVs is greater than one, then the TSV grid becomes a multi-source multi-target network. By adding a super source node S that points to all faulty TSVs and merging all RTSVs into a super target node T , then the TSV grid is converted into a single-source single-target network. Each continuous repairing path can be considered as a unit flow starting from a faulty TSV and ending at a RTSV, then all faulty TSVs can be considered as repairable when the weight of the maximum flow is equal to the number of faulty TSVs. In practice, replacing a faulty TSV with a redundant one may cause additional delay overhead. In this paper, we assume that the timing overhead depends on the distances between the faulty TSV and RTSV. Furthermore, timing violation may occur if the repairing length is too large. As a result, another variable cost is used to model the length of repairing path, and the repairing path found with minimum cost is better. Thus, the TSV repairing problem can be transformed to the classical *minimum cost maximum flow* (referred to *MCMF*) problem.

Algorithm 1 shows the detailed algorithm of finding repairing paths for signal shifting. The inputs are the flow framework $G(V, E)$ and fault map of faulty TSVs $\{FTSV\}$; repairing edges denoted as RE is the output. The algorithm works as below. Firstly, we initialize the flow graph and store it in a queue. After TSV testing, we will get the TSV fault map and we assign a super source node pointing to all faulty TSVs, a super target node connecting RTSVs. For each edge in the flow graph, we initialize the flow to 0. Next, we will find possible augmenting path p

from source to target. By converting the delay overhead to path length, we can judge whether the timing violation occurs or not. When the number of faulty TSVs is equal to the maximum flow obtained, and the length of each augmenting path is no longer than the user-defined constraint allowed, then the TSV grid is repairable and the augmenting path we found represent the repairing path. If timing violation occurs, then the TSV grid is irreparable.

Algorithm 1: Proposed Repairing Algorithm

Input: $G = (V, E)$, Fault map of TSVs $\{FTSV\}$;

Output: Edges $RE \in \{AE\}$;

Initialize G , stored in the queue;

Given $\{FTSV\}$; source and target node connect to faulty TSVs and RTSVs on G ;

Initialize the flow f of each edge to 0;

Find possible augmenting path p from s to t residual network;

for each edge (u, v) in p **do**

if maximum flow = the number of faulty TSV **then**

        adjust each augmenting path p to achieve the minimum path length d ;

if $d < \text{maximum path length allowed}$ **then**

            The grid is reparable;

else

$RE = \emptyset$;

            The grid is Irreparable;

Return RE ;

---

5 Experimental results

First of all, we compare our region-based scheme with the previous router-based [4] and ring-based [5] designs by considering delay overhead. Then, by given a target lifetime, we compare the reparability of different redundancy architectures with different TSV block sizes. Assume that there are $8 * 8$ TSVs constituting the TSV grid. Router-based scheme uses 16 RTSVs placed at the right and bottom boundary of the grid. For the ring-based scheme, 8 RTSVs are located at the four corner and any other positions of the outermost ring of a grid. In our design, there are 8 RTSVs placed at the center of grid and any other locations in the red cross region.

5.1 Comparison of additional delay overhead

As mentioned in [4, 5], the extra delay after repairing depends on the distance between to-be-repaired TSV and the repairing TSV, which is roughly equivalent to the distance between two TSVs in a grid. Assume the logical TSV grid is symmetrical with equal side length, and each distance between two adjacent TSVs has the unit length 1. Then, the maximum shifting length in our design is always 1, which guarantees at most 1 delay overhead of each signal, and the overall extra delay can be roughly represented by the path length of each repairing path.

Fig. 7. Repairing lengths for 8 * 8 TSV grid.

In this section, the length of repairing paths with different types of RTSV designs will be compared. As mentioned in Section 2, for general purpose, we use the topological distance instead of physical wire length to represent the repairing length, and we assume the distance of adjacent TSVs in a grid is 1. The repairing path length comparison of different designs will be shown in Fig. 7. The column labeled 0–8 represents the number of faulty TSVs. For each case, the proposed region-based design has the shortest repairing length, indicating the least delay overhead of all these schemes. For a given number of TSVs (8 * 8), and the number of faulty TSVs ranges from 0 to 8, our design achieves 11.27% and 20.79% reduction of additional delay overhead as compared with router-based design [4] and ring-based scheme [5], respectively.

5.2 Comparison of reparability

Because various sizes of the TSV blocks are used for 3D ICs with different complexities, we like to understand how the best redundant TSV architecture for different sizes of the TSV blocks differs. Fig. 8 shows for a target lifetime, 3-year, the $repair_{3-year}$ of five architectures with different TSV block sizes. X-axis represents grid sizes of TSV block and Y-axis represents $repair_{3-year}$. *no* means no redundant TSV architecture, *shift* means the signal-shifting design [8], *switch* means the signal-switching design [9], *router* means router-based design [4], *ring* represents ring-based design [5], and *region* means our region-based design. When the target year is 3 years, both *router* and *region* achieve above 99% reparability for all sizes of TSV blocks, while *ring* achieves 99% reparability for sizes less than 52 * 52. When the reparability is required to be 95%, *switch* reaches the rate for the sizes from 4 * 4 to 20 * 20 while *shift* reaches the rate for the sizes from 4 * 4 to 8 * 8. However, in terms of hardware cost, *shift* and *switch* are much less than the *router* architecture.

The $repair_{5-year}$ of different architectures is shown in Fig. 9. We can obviously see that when given target time is longer, using large grid-size TSV blocks results in quickly dropping of reparability. We observe that both *shift* and *switch* cannot achieve 99% reparability for all sizes. When the size is lower than 8 * 8, the *region* architecture still achieves high reparability 99%, *ring* can achieve 99% reparability only when the size is lower than 4 * 4. *router* can still achieve 99% reparability when the size is lower than 16 * 16, because this architecture use much more

redundant TSVs, resulting in much hardware costs. By considering RTSV area and delay overhead, the proposed region-based design is the best choice.

Fig. 8. $repair_{3-year}$ of different architectures.

Fig. 9. $repair_{5-year}$ of different architectures.

6 Conclusion

In this paper, we propose a novel region-based TSV redundancy architecture with high yield and low addition delay overhead to repair latent faulty TSVs due to the EM effect. Simulation results show that for a given number of TSVs (8×8), our design achieves 11.27% and 20.79% reduction of additional delay overhead as compared with router-based design [4] and ring-based scheme [5], respectively. More importantly, when the target year is 3 years, the proposed region-based design can achieve above 99% reparability for all sizes of TSV blocks. After a given lifetime 5-year, the reparability of the proposed region-based design can still achieve 99% reparability when the TSV size is lower than 16×16 , which is the best choice by considering RTSV area and delay overhead.

Acknowledgments

This paper is supported by National Nature Science Foundation of China under grant No. (61574052, 61704001, 61874156), Anhui Provincial Natural Science Foundation under grant No. 1808085QF196, Anhui Polytechnic University Research Startup Foundation under grant No. 2018YQQ007, and Key projects of Natural Science Research of universities in Anhui Province under grant No. KJ2016A001.