

Yield-Aware Time-Efficient Testing and Self-fixing Design For TSV-Based 3D ICs

Jing Xie, Yu Wang*, Yuan Xie
 Pennsylvania State University,
 University Park, PA, USA

*Tsinghua University, Beijing, China
 {jingxie, yuanxie}@cse.psu.edu

Abstract—Testing for three dimensional (3D) integrated circuits (ICs) based on through-silicon-via (TSV) is one of the major challenges for improving the system yield and reducing the overall cost. The lack of pads on most tiers and the mechanical vulnerability of tiers after wafer thinning make it difficult to perform 3D Known-Good-Die (KGD) test with the existing 2D IC probing methods. This paper presents a novel and time-efficient 3D testing flow. In this Known-Good-Stack (KGS) flow, a yield-aware TSV defect searching and replacing strategy is introduced. The Build-in-Self-Test (BIST) design with TSV redundancy scheme can help improve the system yield for today's imperfect TSV fabrication process. Our study shows that less than 6 redundant TSVs is enough to increase the TSV yield to 98% for a TSV cluster with a size under 16×16 with relatively low initial TSV yield. The average TSV cluster testing and self-fixing time is about 3-16 testing cycle depending on the initial TSV yield.¹

I. INTRODUCTION

Three-dimensional integration based on through-silicon-vias (TSVs) has been considered as one of the most promising technology to overcome 2D scaling limitation and push integrated circuit to go beyond *Moore's law* [1] [2]. The significant improvements in performance, bandwidth, and power benefits drive many semiconductor vendors and IC manufacturers to explore 3D solutions for their future generation product lines [3]. Cost is one of the most important consideration for an emerging technology to become main-stream. Consequently, the yield of TSV-based 3D integration is one of the major barriers for its commercial success, and therefore testing for 3D ICs plays a key role in improving the yield and reducing the cost of the 3D ICs.

Testing for TSV-based 3D ICs has many unique challenges that are different from traditional 2D IC testing, such as the lack of probe access for wafers, test access to modules in stacked wafers/dies, mechanical issues, thermal concerns, test economics, and new defects arising from unique processing steps (such as wafer thinning, alignment and bonding) [4]. In wafer-to-wafer stacking, several layers (or so-called *tiers*) of wafers are stacked together before testing, which could save testing cost but lead to severe yield problems. According to the 3D cost model by Dong [5], simply stacking all the tiers and testing them together in a wafer-to-wafer bonding method is more expensive than the die-to-wafer bonding and the die-to-die bonding as the number of tiers increases. The latter two methods require the Known-Good-Die (KGD) test before

every stacking step, and the Know-Good-Stack (KGS) test (i.e., test the whole stack of tiers to make sure there is no defects) after every stacking step [6]. However, in order to reduce the TSV diameter with reasonable aspect ratio, each layer(tier) of a 3D chip is thinned to tens of micros [3]. The traditional 2D probing-based testing may cause wafer cracking with new defects for such thin wafers. In addition, there are no pads and ESD circuitry for TSVs connecting two tiers in order to achieve high-speed inter-tie communication.

Many research works have been reported in 3D testing field. For example, Marinissen proposed a Known-Good-Stack(KGS) testing method [6], and discussed the possibility of using the contact-less method for 3D testing [7]. Lou proposed three methods for testing TSV side-wall leakage and void/pinhole defects [8]. Huang *et al.* presented a BIST scheme for the TSV post-bond testing [9]. Ven der Plas *et al.* presented interesting experiment result of TSV cluster's yield and found that the yield of TSVs is location-dependent within a TSV cluster [10]. Methods of applying TSV redundancy were also studied to obtain higher yield [11].

In this work, we propose a TSV testing flow that only uses the pads of the bottom tier as system testing input. This flow focuses on providing better testing time and higher yield. We present a yield-aware testing and TSV replacement design algorithm. The TSV-only testing time is much faster than that for logic testing. Meanwhile, TSV has much lower yield than normal CMOS devices under current technology [10], so that detecting the TSV defects first can help decrease the testing cost. Our proposed testing flow and algorithms provide a different understanding of testing time and yield improvement for 3D ICs. The proposed design methodology uses yield as a parameter in Design-For-Test. We study the testing time and testing overhead under different TSV cluster yield. The self-fixing method applies redundancy to multiple defects TSVs in one cluster and its overhead is small.

II. THE PROPOSED METHODOLOGY AND MECHANISM

Known-Good-Die(KGD) test is commonly used in System-in-Package (SIP) testing flow. Chips with different functionality, for example, processor, DRAM, RF chips, can be packaged together using wire bounding or other package technologies. The difference of SIP testing and TSV-based 3D IC testing is that each die in a SIP has separate pads. These dies could be probed and functional test can be performed independently before packaging. It is not easy to perform KGD test for TSV-based 3D stacking, because typically only one 3D stacking

¹This work is supported in part by SRC grants, NSFC 61028006, NSF 0643902, 0903432, and 1017277.

tier carries all the system pads and the input and output pins of other tiers are exploded without ESD protection before stacking. On the other hand, the TSV defect rate could be much higher than that of wire bonding in normal SIP flow. In this section, we describe our proposed methodology and mechanism.

A. Time-Efficient Testing Flow

TSV-based 3D integration has three stacking methods: face-to-face, face-to-back and back-to-back. Our proposed methodology targets at the die-stacking method with face-to-back stacking for all tiers, except for the last tie, which uses face-to-face stacking, as illustrated in Figure 1. The pads on the bottom layer is connect to package from backside through TSV, so that the face to face metal resource can be reserved for inter-tier connection.

Our testing flow can be divided into two stages.

Stage I: Test the TSVs only and apply redundancy to fix defected TSVs.

Stage II: Test the internal circuit function.

The testing begins with tier 0 and follows these two testing stages. Stage I is performed first and Stage II is applied only after passing stage I. These two stages contribute to the test time separately. After finishing the test on tier 0, tier 1 is tested in the same way, and so on. In our targeted stacking structure (as shown in Figure 1), the last tier (Tier N) has no TSVs and uses top metal layer for inter-tie connection. Therefore, Tier N can skip the stage I. Usually, the normal CMOS process has pretty high yield, while the TSV yield still need to be improved [10]. Testing time is a key factor of testing cost [12]. Finding those TSV defects earlier can optimize the total testing time and reduce testing cost. The tier stacking method in Figure 1 shows that the BIST logics on each of the tiers to perform TSV self-test, whose structures will be discussed in next section.

The detail testing flow in Figure 2 is a modified Known-Good-Stack (KGS) testing strategy to increase yield. The logic testing method for 3D ICs has been discussed in prior work [13]. Our flow can use any existing logic testing methods for circuit testing. Our design focus on searching defect TSVs and applying redundancy. Such flow can increase the yield from two aspects: (1) no tiers will be stacked on defected chips; (2) using the testing result to config the TSV replacement circuit.

B. Circuit Level Consideration of Design for Testing

There are two TSV placement methods: the arbitrary pattern and the TSV cluster. The arbitrary placement provides more design flexibility, but recent research shows that the arbitrary placement is more vulnerable to TSV mechanical stress, which can result in TSV reliability issues [14] and affect the nearby CMOS device mobility [15]. As a result, we focus on TSV clustering design in this work.

The testing circuit configuration of an $N \times N$ TSV cluster is illustrated in Figure 3. A MUX-based test chain connects all the TSVs to the test block. During testing, the input signal

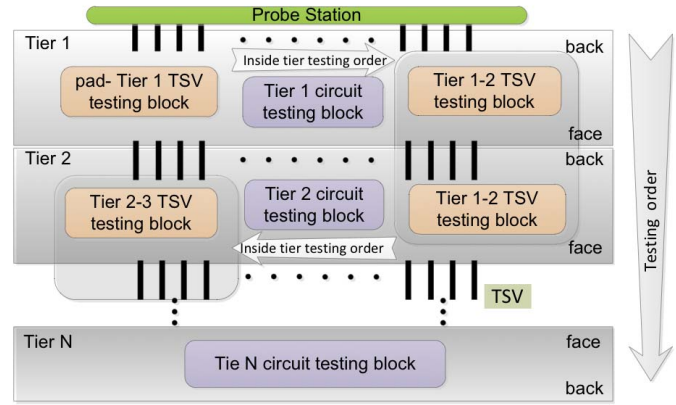


Fig. 1: Die stacking scheme and test order.

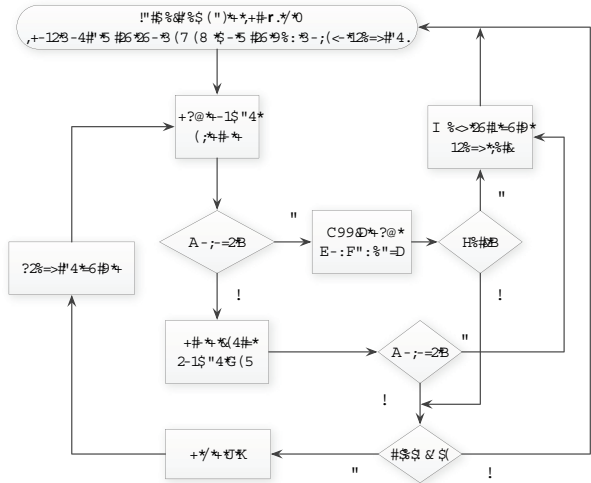


Fig. 2: Timing efficient stacking and test flow. N is the number of 3D stacking layers.

is generated with a 50% duty cycle pulse. When all the TSVs in a testing chain are functioning, the test block receives a flipping signal. In testing mode, the select signals $\langle s_0 : s_n \rangle$ determine which part of the TSV chain is under test. The *in* testing signal can be generated on chip with changeable frequency. Since this pulse input does not carry any data information, it can be fast and reach GHz, which is much faster than scan-in speed. Therefore, in test time discussion, one cycle or one testing cycle is not the pulse cycle time, but stands for the time to finish the test under one select signal $\langle s_0 : s_n \rangle$ configuration. In operation mode, the MUXs should not affect the data signals carried by TSVs. Hence, there are transmission gates inside the MUXs to disconnect them from TSVs, when necessary.

TSV delay under recent technology is less than 10 FO4. The testing signal can pass through each TSV in one pulse cycle. If one or more TSVs are affected by large variation or have defects, their RC delay may be larger than good TSVs. When such delay exceeds one pulse cycle, the *out* signal stops flipping, because the defect TSV output could not reach a threshold before the input signal changes. The electric model of a TSV is similar to a long metal wire [16]. Its equivalent

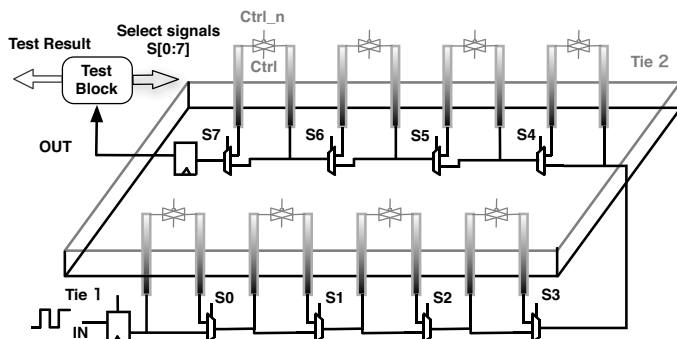


Fig. 3: TSV test chain. One cluster of TSVs are connected together by MUX and FF. The Test Block will generate the select signal of all the MUX and result of defect TSV locations.

circuit is shown in Figure 4, where each TSV has its own driving cell. The large side-wall capacitance makes it a heavy load for the driving cell. The signal transmitted through it experiences a relatively long delay.

We use TSV delay to distinguish defects. Such defects can be detected by changing the pulse frequency. The TSV delay also depends on its driving cell size, which should be small intentionally to sense the frequency change. Thus the carefully sized testing circuits do not have large area overhead comparing with the TSV area.

There are testing circuit on both tiers to test the TSV defects together. For example, The black tier (tier 1 in Figure 3) is closer to the bottom tier and has already been tested. In this tie, all the input signals are reliable. The grey tier (tier 2) is the upper tier. The TSV under test belongs to *tier 2* and all the signals in *tier 2* cannot be trusted yet. The *ctrl* signals of the transmission gate on tier 2 are set to be 1 as long as the upper layer is powered on. According to today's TSV yields, many TSV design rules require to group two TSVs together to transfer one signal. Accordingly, our testing circuit will testing two TSVs as a group. The testing result only shows whether a two-TSV group has defect. If these two TSVs need to transfer different signals, transmission gate will be turned off during chip operation.

The test block has three functions: (1) monitoring the *out* signal to check whether it is flipping; (2) generating the select signal to control the TSV cluster MUX. In this way, the tested TSVs group can be excluded from the chain; (3) providing defect information in the TSV cluster, which will be used locally to apply redundancy. The testing block senses if the *out* signal is flipping every test cycle, which is much longer than the pulse cycle. Therefore a long testing chain will not violate any timing constraint.

C. Yield Modeling and Defect Searching Algorithm

TSV redundancy can improve chip yield by replacing the failed TSVs with nearby redundant TSV resources. At the design stage, a BIST scheme can be implemented after the number and the location of the redundant TSVs are decided. A yield model is introduced in this design for estimating the number of extra TSVs required. After that, a breadth-first

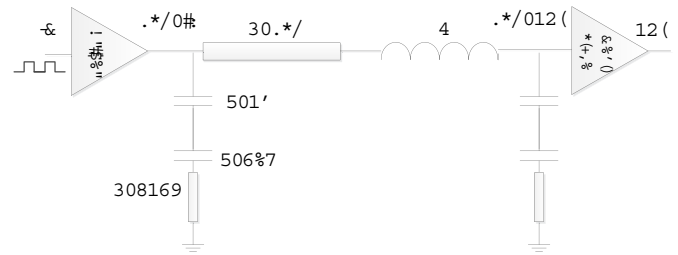


Fig. 4: TSV circuit model

binary search method is used to find the failed TSVs and apply redundancy.

The delay model for a TSV is discussed in the previous section. Process variation on its resistance and capacitance value causes uncertainty in signal transmission delay time. The delay for a cluster of TSVs fluctuates in a certain range. If the delay of a TSV is larger than a threshold, a switching signal cannot pass through it and this TSV is considered as failed.

The first step of this BIST design is to find out the number of redundant TSVs needed. The design input is the process variation model of a TSV, which is based on the prior work by Wu et al. [17]. Monte Carlo method is used to estimate the TSV defect rate. The modeling flow is shown in Algorithm 1. An instance of a TSV cluster is created, and the delay value for each TSV is stochastically determined using the process variation model. Two neighboring TSVs are grouped together, and if any of them has a delay longer than a threshold, this group is marked as failed. The threshold value can be decided based on the test pulse frequency and also the strength of the driver. The total number of failed TSV groups is counted for this instance. The number of extra TSV groups needed to fix it is calculated by dividing the number of failed TSV group with the average TSV yield.

The number of extra TSV group used for achieving a target cluster yield is determined by statistically analyzing the Monte Carlo simulation results. The simulation generates thousands of TSV cluster instances and records the number of TSV groups needed to be fixed. For example, if 3 redundant TSV groups are available, the instances with 3 or less defects can be fixed. The yield after applying the redundancy is calculated, where the fixed instances are now considered as a good one. The number of extra TSV group keeps increasing until reaching the target yield.

After determining the number of redundant TSV groups, the locations also need to be selected. The yield of a TSV is related to its location within the cluster. Prior work has shown that the TSVs at the edge of a cluster have lower yield comparing to the TSVs in the middle [10]. The yield at the four corners are even worse. Our model considers this case by increasing the fluctuation range of R and C values for those TSVs locating at the edge of the array. These TSVs have a higher possibility to fail. The yield map from our model agrees with the experiment results from [10].

Figure 5 shows an example of the location dependent TSV cluster yield, where the yield at the edge of the cluster is

Algorithm 1: Find Redundant TSV number

```

Input: TSV parameter variation model
Output: Number of extra TSV
for  $i = 1$  to  $instance\_number$  do
  Generate TSV delay map by TSV variation model;
  foreach TSV do
    if  $delay > threshold$  then
      mark as failed;
  Group two TSVs, and find the group pass/fail map;
  Calculate  $failed\_group\_number$ ;
   $extra\_group\_need[i] =$ 
   $failed\_group\_number/redundant\_yield$ ;
   $redundancy\_number = 0$ ;
  while  $yield\_after\_fix < target\_yield$  do
     $redundancy\_number ++$ ;
    foreach  $instant[i]$  do
      if  $extra\_group\_need[i] \leq extra\_group\_number$  then
         $instant[i] = good$ ;
      else
         $instant[i] = failed$ ;
    update  $yield\_after\_fix$ ;
  return ( $redundancy\_number$ );

```

lower than that at the center. Two possible redundant TSVs placement methods are shown at the right side of the figure. One is to place them in a uniformly spacing array, and the other is to put them at the edges. In the second case, the redundant TSVs are placed at the low yield part, so that the normal TSVs use the high yield portion. This method saves test time, since the normal TSV yield is higher than the average yield and fewer test cycles are required. The disadvantage of placing redundant TSVs at the edges is that the distance between a failed TSV and its backup is larger comparing to a uniformly distributed redundant TSV array. The routing network that transfer the signal to its backup TSV may have long wire delay.

After determining the amount of redundancy resources available, a BIST logic is designed to find and replace the failed TSV group with a good one. The first step is to find out the locations of those failed groups, if there is any. A breadth-first search algorithm is designed to perform this task, as shown in Algorithm 2. Its inputs are the TSV cluster size and the number of redundant TSVs available. Its outputs are the location and the total number of the failed TSVs. The search algorithm measures a portion of the full test chain in one test cycle. If the result indicates that it contains failure groups, the portion of the chain under test is separated into two chains from the middle and test again. A First-In-First-Out (FIFO) queue is used to store the start and end positions of the section that need to be tested. It receives the indexes of such section and put them at the tail of the queue. For a data request, it sends out the first item at the head of the queue and removes it from the FIFO.

The search begins by examining the full test chain. The variable *current* in Algorithm 2 indicates the section under test in current cycle. It is set to cover the full chain during the

Algorithm 2: Breadth First Binary Search of TSV defects

```

Input: The TSV cluster size N, extra TSV number
Output: Location of failed TSVs, number of failed TSV
foreach TSV clusters do
   $current = [test\_chain\_start\ test\_chain\_end]$ ;
  while  $in\_test == 1$  do
    if  $check\_tsv\_chain(current) == fail$  then
      if  $current\_start == current\_end$  then
        add  $current\_start$  to  $failed\_location$ ;
         $failed\_number ++$ ;
      else
         $first\_half = [current\_start$ 
         $(current\_start + current\_end)/2]$ ;
        push ( $first\_half$ ) => FIFO;
         $second\_half = [(current\_start + current\_end)/2$ 
         $current\_end]$ ;
        push ( $second\_half$ ) => FIFO;
    if  $fifo$  is empty then
       $In\_test = 0$ ;
    else
       $current = pop(fifo)$ ;
    if ( $failed\_block > extra\_tsv\_num$ ) at current search
    level then
       $in\_test = 0$ ;
      Set cannot fix;

```

first cycle. If the test result is good, no more action is required. If there is failed TSV groups in the chain, the indexes of the first half and the second half of the chain are pushed into the FIFO. In the subsequent cycles, the test block fetches one section from the FIFO and tests it. If this section failed, the test block repeats the previous separating and pushing operation again.

Algorithm 2 has a $d * \log(n)$ time complexity, where d is the number of redundancy TSV. n is the TSV cluster size. The search algorithm ends in two ways. One case is that it reaches a single TSV group and this group has defect. The location index of this failure point is recorded and the total failure number is increased by one. After all potential failure chain portion are examined, the FIFO is empty and the search finished. Another exit case is that the minimum number of failure points exceeds the number of redundancy resource available. The breadth-first search checks all possible failed section in one level then steps into the next level. If a section failed, it contains at least one failure TSV group. The minimum number of defects can be calculated by counting the number of failed section within a level. If there are more defects than the redundancy TSV groups, the cluster under test cannot be fixed and the test should be stopped. The size of the FIFO is determined by this ending condition. The maximum number of items in the FIFO is less than two times of the number of redundancy.

III. EXPERIMENTAL RESULTS

In this section, we evaluate the test time and yield improvement of the purposed BIST design. The delay time of a TSV is simulated with HSPICE using the electrical model shown in Figure 4. The TSV resistance and capacitance fluctuation

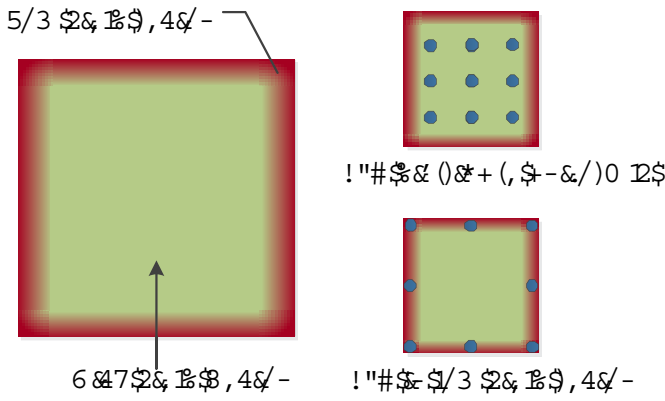


Fig. 5: TSV yield map

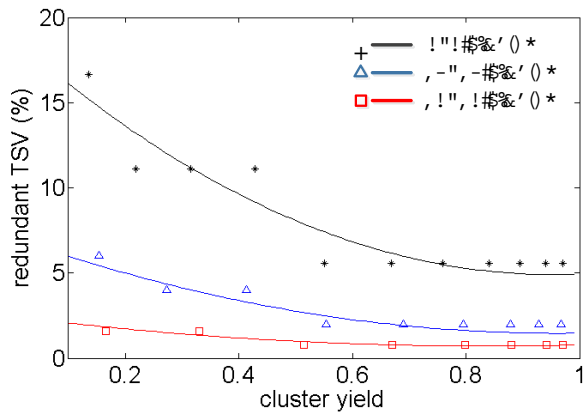


Fig. 6: Relationship of TSV cluster yield and number of redundant TSVs required to reach 98% TSV cluster yield

is described by normal distribution $N(\mu, \sigma^2)$. The TSVs at the edge of the cluster suffer larger process variation, so that their σ_{edge} is $\gamma (= 1.2)$ times bigger than the σ of normal TSVs. A delay time library was built, so that the BIST simulator can quickly find the delay for a specific R and C configuration.

Monte Carlo method is used to simulate the yield of a TSV cluster. 100,000 TSV cluster instances were created and their yield maps were generated with different delay thresholds. The breadth-first search algorithm was simulated at per-cycle level using a Matlab program for each instance, where the yield map at a specific delay threshold is used as its input. For each delay threshold, the yield before and after applying redundancy is statistically analyzed among all instances, and the average test time is calculated.

1) For a target yield, calculate the number of redundant TSV needed. The number of redundant TSV groups needs to be determined at the beginning of the BIST design. It is related to the yield of the TSV cluster, and also the size of the cluster. The Monte Carlo simulation results of three cluster sizes were plotted in Figure 6. These three sizes cover the reasonable cluster sizes in 3D IC design. The cluster yield in X axis is calculated as the percentage of instances that are defect-free in the Monte Carlo simulation. The target yield for this analysis is set to 0.98. The Y axis indicates the percentage of redundant TSVs achieve this target cluster yield. At higher

yield ($> 50\%$), the number of redundancy remains almost unchanged. This illustrates that only one or two defect sites appear for most of the instance. In this case, including a small amount of redundant TSVs can fix most of the failed clusters. At low yield, the number of redundant TSVs required increases quickly, which indicates the increase of the defects number. The amount of resources used in a small cluster, 6 by 6 for example, is larger than a big cluster. It is better to share the redundancy resource among more TSVs. When the cluster size increases larger than 10 by 10, the benefit of sharing is not very obvious, so that test time will become a more important consideration.

2) Performance Evaluation. The efficiency of this TSV redundancy design is evaluated by its testing time and the yield improvement. Three redundancy configurations are studied, including 1) one extra TSV in the center; 2) one extra TSV at the edge; and 3) three extra TSVs at the edge. The simulation results are shown in Figure 7.

The top row shows that the average test time for one cluster increases when its yield becomes lower. However, when the number of defects is larger than one, there is no enough redundancy resource to fix it and the search process stops after a few cycles of tests. That is the reason for the quick decrease at the low yield end. The maximum test time appears at the yield of 0.2-0.4. The test time also depends on the cluster size. A smaller array needs fewer cycles to locate the defect position. However, the difference is only 2-3 test cycles between a 6x6 and a 16x16 array. The time complexity of the search algorithm is $d * \log(n)$, and usually d is 1-2 in the high yield region. The effect of cluster size on testing time is logarithmic, so that the testing time variation is small.

The bottom row shows the improvement in cluster yield with different defect rates. The X axis is the initial yield before applying redundancy. The diagonal line is the reference, and the vertical distance between the data curve and the diagonal line is the improvement of yield by applying the redundancy scheme, as shown by the arrow at the left bottom plot. The improvement is large at the middle portion of the curve. This observation points out that a 3D connection process with low yield benefits more from this BIST design. The yield improvement depends on cluster size and the number of extra TSV groups. For a large cluster, the curve changes slightly among several configurations. On the other hand, the curve for a small cluster changes rapidly when the extra TSV group number increases.

Comparing the cases that only one extra TSV group is placed at different locations (left and middle columns), the shape of the average testing time curves are similar. The average testing time is around 1 cycle lower for the 16x16 cluster when the redundant TSV group is placed at the edge. The average yield is higher for TSVs in the center, so that the cluster has high possibility to be a good one and pass the test in one cycle.

Comparing the cases that one or three redundant TSV groups are applied and placed at the edge, the test time and yield curves show big difference (middle and right columns).

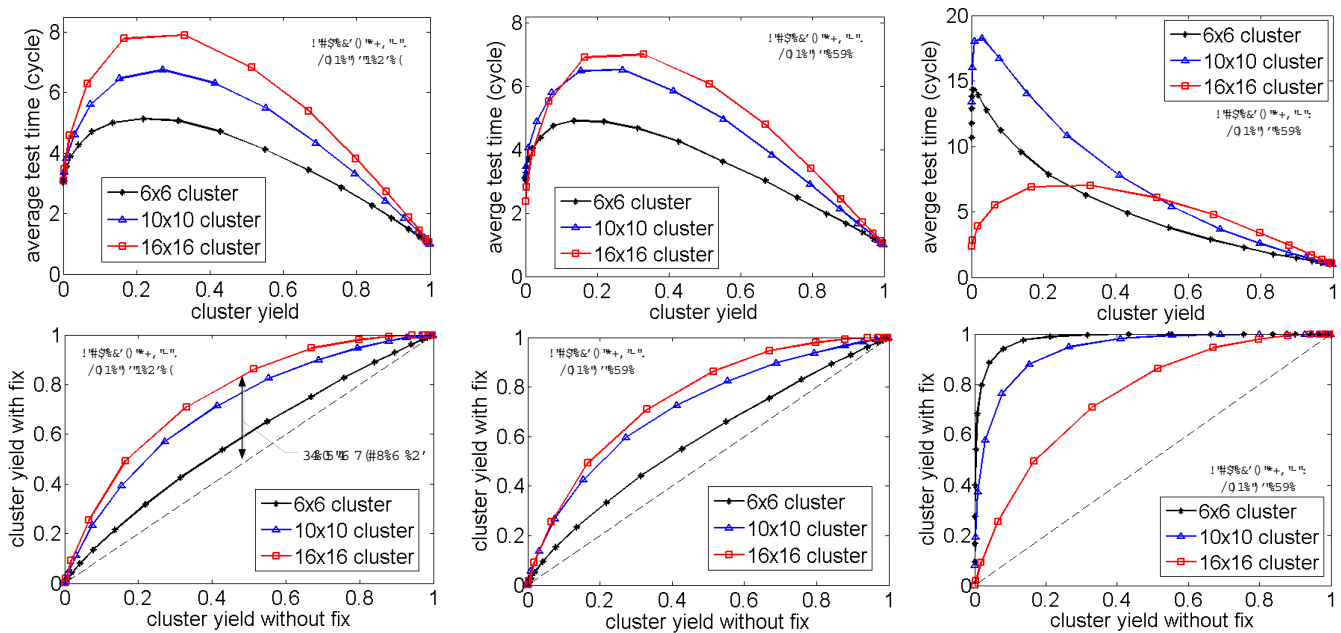


Fig. 7: Comparison of average test time (top row) and yield after applied redundancy (bottom row) for several schemes. The X axis represents the TSV cluster yield before applied redundancy. (A) One redundant TSV placed in the center of the cluster; (B) One redundant TSV placed at the edge of the cluster; (C) Three redundant TSVs placed at the edge. Three cluster size were used in experiment, which is (black) 6x6 TSV cluster, (blue) 10x10 array, and (red) 16x16 array.

The test time for the 16x16 size cluster does not change much. However, the shape of the curves changes largely for small clusters. When the redundant TSV groups increases, it is possible to fix more defects. The search process keeps running even for very poor yield clusters and tries to fix it with the abundant resource. The yield curve shows great improvement (70%-80%) at the low initial yield portion for small clusters.

IV. CONCLUSION

In this work, we designed a TSV-first Known-Good-Stack test scheme for reducing 3D ICs test time and increasing yield. Our design is based on the TSV cluster structure. We presented a breadth-first binary searching algorithm for BIST design. This algorithm found the failed TSVs within a few cycles. The defect TSVs were replaced by redundant resources with small overhead. The BIST algorithm and circuit achieved high testing speed and can decrease the testing time to under 20 cycles. This methodology worked for a large range of defect rate and significantly increased the overall TSV cluster yield. When the TSV cluster yield is over 30%, less than 6 redundant TSVs can fix all the defects in most case.

REFERENCES

- [1] Y. Xie, G. Loh, B. Black, and K. Bernstein, "Design space exploration for 3d architecture," vol. 2, no. 2, pp. 65–103, 2006.
- [2] Y. Xie, G. Loh, and B. Black, "Processor design in three-dimensional die-stacking technologies," vol. 27, no. 3, pp. 31–48, 2007.
- [3] Tezzaron, "Tezzaron: The very best in 3D-IC." <http://www.tezzaron.com/TezzaronBest3DIC.html>, 2011.
- [4] J. Xie, J. Zhao, X. Dong, and Y. Xie, "Architectural benefits and design challenges for three-dimensional integrated circuits," in *APCCAS*, pp. 540–543, 2010.
- [5] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," in *ASP-DAC*, 2009.
- [6] E. Marinissen and Y. Zorian, "Testing 3D chips containing through-silicon vias," in *ITC*, pp. 1–11, 2009.
- [7] E. Marinissen, D. Y. Lee, J. Hayes, *et al.*, "Contactless testing: Possibility or pipe-dream?," in *DATE*, pp. 676–681, 2009.
- [8] Y. Lou, Z. Yan, F. Zhang, and P. Franzon, "Comparing Through-Silicon-Via (TSV) Void/Pinhole Defect Self-Test Methods," *Journal of Electronic Testing*, pp. 1–12, 2011.
- [9] Y.-J. Huang, J.-F. Li, J.-J. Chen, *et al.*, "A built-in self-test scheme for the post-bond test of TSVs in 3D ICs," in *VTS*, pp. 20–25, 2011.
- [10] G. Van der Plas, P. Limaye, I. Loi, *et al.*, "Design Issues and Considerations for Low-Cost 3-D TSV IC Technology," *JSSCC*, vol. 46, no. 1, pp. 293–307, 2011.
- [11] A.-C. Hsieh, T. Hwang, M.-T. Chang, *et al.*, "TSV redundancy: Architecture and design issues in 3D IC," in *DATE*, 2010.
- [12] M. Bushnell and V. Agrawal, *Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits*. Frontiers in electronic testing, Kluwer Academic, 2000.
- [13] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test-access mechanism optimization for core-based three-dimensional SOCs," in *ICCD*, pp. 212–218, 2008.
- [14] M. Jung, J. Mitra, D. Pan, and S. K. Lim, "TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC," in *DAC*, pp. 188–193, 2011.
- [15] H. Chaabouni, M. Rousseau, P. Leduc, *et al.*, "Investigation on TSV impact on 65nm CMOS devices and circuits," in *IEDM*, pp. 35.1.1–35.1.4, 2010.
- [16] L. Cadix, M. Rousseau, C. Fuchs, *et al.*, "Integration and frequency dependent electrical modeling of Through Silicon Vias (TSV) for high density 3DICs," in *IIITC*, pp. 1–3, 2010.
- [17] X. Wu, W. Zhao, C. Nimmagadda, D. Lisk, M. Nakamoto, S. Gu, R. Radojic, M. Nowak, and Y. Xie, "Electrical Characterization for Inter-tier Connections and Timing Analysis for 3D ICs," in *TVLSI*, 2011.