J Electron Test (2017) 33:573–589
DOI 10.1007/s10836-017-5681-x

Three-Stage Optimization of Pre-Bond Diagnosis of TSV Defects

Bei Zhang[1,2] · **Vishwani D. Agrawal**[1]

Received: 19 March 2017 / Accepted: 15 August 2017 / Published online: 2 September 2017
© Springer Science+Business Media, LLC 2017

Abstract Pre-bond TSV testing and defect identification is important for yield assurance of 3D stacked devices. Building on a recently proposed pre-bond TSV probing procedure, this paper develops a three-stage optimization method named "SOS3" to greatly reduce TSV test time without losing the capability of identifying given number of faulty TSVs. The optimization stages are as follows. First, an integer linear programming (ILP) model generates a near-optimal set of test sessions for pre-bond defective TSV diagnosis. Second, an iterative greedy procedure sequences the application of those test sessions for quicker diagnosis. Third, a TSV defect identification algorithm terminates testing as quickly as possible, often before all sessions are applied. Extensive simulation experiments are done for various TSV networks and the results show that the SOS3 framework greatly speeds up the pre-bond TSV test.

Keywords 3D stacked integrated circuits · Pre-bond TSV probing · Test sessions · Through-silicon via (TSV) defects

Responsible Editor: K. K. Saluja

✉ Bei Zhang
bzz0004@auburn.edu

Vishwani D. Agrawal
agrawvd@auburn.edu

[1] Department of Electrical, Computer Engineering Auburn University, Auburn, AL 36849, USA

[2] Apple Inc., Cupertino, CA 95014 USA

1 Introduction

Through-silicon vias (TSVs) act as media for transporting power supply or signals among layers of a three-dimensional (3D) stacked integrated circuit (IC). TSV yield is typically high (> 99%), but when there are thousands of TSVs in a die, the probability of all TSVs being defect-free could be low. TSV testing, diagnosis and repair are very important as a single irreparable TSV can cause an entire stack to fail. TSVs can be tested before die bonding (pre-bond) or after die bonding (post-bond). Pre-bond TSV test targets defects arising in wafer manufacturing, such as a void within a TSV, a complete break in a TSV, a pinhole creating a leakage path between TSV and substrate, etc. Pre-bond TSV testing is important as it helps identify defective dies early in the process and avoid situations where one single bad die causes entire 3D stack to be discarded. It is also necessary in providing known good die (KGD) information for the die-to-die or die-to-wafer fabrication process. Even for wafer-on-wafer stacking, pre-bond TSV test helps in better wafer matching and thus improves the yield [20, 24, 25, 30, 34].

A post-bond TSV test targets defects that arise during the assembly process, and is also necessary and important. These defects may be caused by TSV misalignment, mechanical stress, thermal issues, etc. Recent work [2] identifies 11 kinds of TSV defects, of which six occur before bonding. The post-bond TSV test has been extensively studied [7, 11, 12, 19]. After bonding TSVs are basically treated as wires. Post-bond TSV testing can also be conducted by employing the developing IEEE P1838 standard [11–13, 16]. Post-bond TSV testing serves as the final defense

line to guarantee the quality of TSVs and hence is equally important. However, if any defective TSV can be pinpointed before bonding, then we may be able to avoid some post-bond TSV testing or die-internal circuitry testing by early elimination of unrepairable dies.

The pre-bond TSV testing is challenging mainly because before-bonding a TSV is single ended, i.e., one of its ends is not connected to any circuitry. For pre-bond TSV test, we can test on a still thick wafer. In that case, the TSVs are deeply buried in the wafer substrate without any test access. This requires special per-TSV test logic, such as, e.g., built-in self-test (BIST), to test TSVs with only single-sided access. Several BIST techniques have been proposed for buried TSVs [3], including a voltage division circuit to measure the leakage resistance of TSVs to detect pin-hole defects, a DRAM and ROM-like test to determine the RC time constant, and measurement of resistance of blind TSVs or open-sleeve TSVs. Ring Oscillators have been widely used to characterize the propagation delay of TSVs and thereby diagnose possible resistive open or leakage defects [5, 28]. All BIST approaches require dedicated circuit to be added for each individual TSV, and the area overhead is huge since there can be tens of thousands of TSVs on a chip [10, 26]. Moreover, the BIST circuits themselves suffer from process variation, which may render them totally useless. An alternative is to test thinned wafers where TSV tips are exposed. This requires special facilities to probe thinned wafers (about 50 μm thick) without damaging them. However, the relatively large pitch (40 μm) of current probing technology [22, 27] prohibits individual TSV probing with a realistic pitch of 10 μm [10, 17].

A pre-bond TSV probing method has been recently proposed [15] where a group of TSVs, defined as a *TSV network*, is simultaneously contacted by a single probe needle. The number T of TSVs within a network is typically less than 20 and depends on the relative diameter of the probe needle and the pitch of TSVs [10, 16, 17, 22, 27]. TSV parametric test can be conducted by adding an active driver in the probe needle and forming a charge sharing circuit between single (or multiple) TSV(s) and the probe needle. This probing method offers robustness to process variation, requires less hardware overhead, provides accurate measurement of TSV resistance, and has many more benefits as explained in its original proposal [14–16].

For the TSV network probing procedure [15], a heuristic method [14] generates a series of test sessions that can uniquely locate a given number of faulty TSVs within a network. This heuristic method has been shown to reduce the test time compared to that of testing each TSV individually. However, the improvement is far from being optimal.

In this work, based on a number of publications and a doctoral dissertation [29], we address test session generation and application of pre-bond TSV probing, with focus on minimizing the identification time of faulty TSVs. This work has four parts:

1) An integer linear programming (ILP) model is devised for diagnostic test session generation for pre-bond TSV defects [33]. Compared to previous heuristic method [14], this ILP model always produces fewer sessions and a much reduced total test time.
2) A fast TSV identification algorithm is developed to identify defective TSVs based on given test sessions [31].
3) An iterative greedy procedure for session sorting [32] terminates pre-bond TSV test quicker when the number of faulty TSVs within a TSV network is small (< 2).
4) Finally, we combine the three preceding items into a three-stage optimization simulator (“SOS3”) for test time minimization. Thus, the three stages are: ILP-based session generation [33], iterative greedy procedure for session sorting [32], and fast TSV identification algorithm for early test termination [31]. Each stage provides input to the next stage. Experimental results demonstrate that SOS3 guarantees the expected time for TSV identification to be much lower than the total time of applying all sessions.

This paper is organized as follows. Section 2 summarizes the previous knowledge on pre-bond TSV probing and the motivations for this work. Section 3 defines the terminologies used in this paper. An ILP model for generating near-optimal set of test sessions is proposed in Section 4. Section 5 proposes an efficient TSV fault identification algorithm, which terminates as soon as the identification goal is achieved. A probabilistic analysis for a random number of faulty TSVs within a TSV network is described in Section 6. This serves as the motivation for the work presented in Section 7 that gives an ILP-based iterative greedy procedure to sort test sessions for further test time reduction. The framework named three-stage test time optimization simulator (“SOS3”) is presented in Section 8. Experiments are conducted in Section 9. Section 10 outlines some limitations of this work suggesting future research directions. Finally, Section 11 concludes the paper.

2 Preliminaries and Motivations

Since most pre-bond TSV defects are resistive in nature [2], pinpointing resistive defects is important. Considering the relative sizes of typical test probes and TSVs, an earlier proposal [15] uses a large probe needle with an active driver to make simultaneous contact with multiple TSVs at a time. Figure 1 shows a probe needle connected to two TSVs on the backside of a thinned die. When conducting die bonding in 3D IC, micro-bumps or landing pads are typically applied

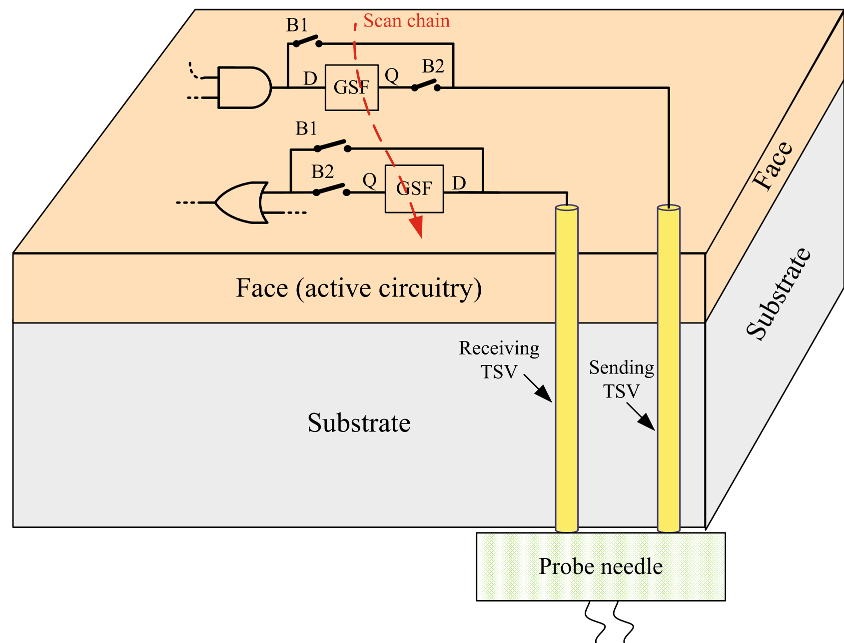

Fig. 1 Illustration of pre-bond TSV probing

to the TSV tip for bonding purpose. The proposed approach is about testing the TSV before any micro-bump or landing pad is attached to it. In Fig. 1, gated scan flip-flops (GSF) are inserted between system logic (shown as AND and OR gates) and TSVs in accordance with the developing IEEE P1838 standard [12, 16]. Here, the left TSV is a receiving TSV that would receive a signal from the other die and drive the on-chip logic while the right TSV is a sending TSV that is driven by the on-chip logic and sends a signal to the other die. In the normal mode, all GSFs are bypassed by opening B2 switches and closing B1 switches using a common "bypass" signal. Now the receiving TSV in Fig. 1 feeds signal from the other die (probe in this case) to the input of the OR gate, and the sending TSV is driven by the output of the AND gate.

Figure 2 shows a circuit model of the probing test setup [15, 16] for a 4-TSV network example. TSV i is represented by its resistance R_i and capacitance C_i. R_c is the contact resistance between TSV and the probe. R_p is the probe needle resistance. A gated scan flip-flop (GSF) is inserted between TSV i and the system logic. All GSFs can be loaded up or read out through a boundary scan mechanism. In the normal mode, all GSFs are made transparent. In the pre-bond TSV test mode, all GSFs drive respective TSVs. In Fig. 2, TSVs 1 and 2 are of the receiving type and TSVs 3 and 4 are of the sending type. A GSF in scan mode drives a receiving TSV during pre-bond TSV probing when both B1 and B2 switches are closed. A GSF drives a sending TSV when B1 is opened and B2 is closed. Note that there will be no drive conflict as all TSVs irrespective of whether they are sending or receiving will be driven by the GSF during pre-bond TSV testing. The configuration of switches differs for sending and receiving TSVs.

Power, ground, clock and special test signals controlling B1 and B2 switches are supplied to on-chip circuitry by making contact to dedicated TSVs with large probe pads already attached to them. Those signals are pre-requisite to any further TSV testing, and they will be supplied by TSVs with large probe pads contacted by special probe needles in the probe card. Please refer to the original proposal [15] for more implementation details.

Pre-bond TSV resistance measurement begins by scanning "1" in all GSFs. Capacitance C_{charge} and all TSVs are then discharged through the probe needle. By configuring the switches of a GSF, a charge sharing circuit is constructed between that GSF and C_{charge} through a TSV (either sending or receiving type). The charging rate of C_{charge} is compared to a calibrated curve of a good TSV to determine the resistance of the TSV under test.

A parallel TSV test is conducted by configuring multiple GSFs at a time. Now, C_{charge} is charged faster and the measurement terminates quicker. However, the number of TSVs tested in parallel cannot exceed a constant "r" due to a minimum measurement resolution constraint [15, 16]. This resolution of measurement refers to the minimum change in TSV resistance that can be detected by the technique and it is adversely affected by the number of TSVs tested in parallel. We call the TSVs tested in parallel within a TSV network a *test session*. Based on this probing technique, any faulty TSV within a session will cause the session test to fail but we cannot tell which TSV(s) is (are) faulty. On the other hand, a passing parallel test implies that all TSVs within

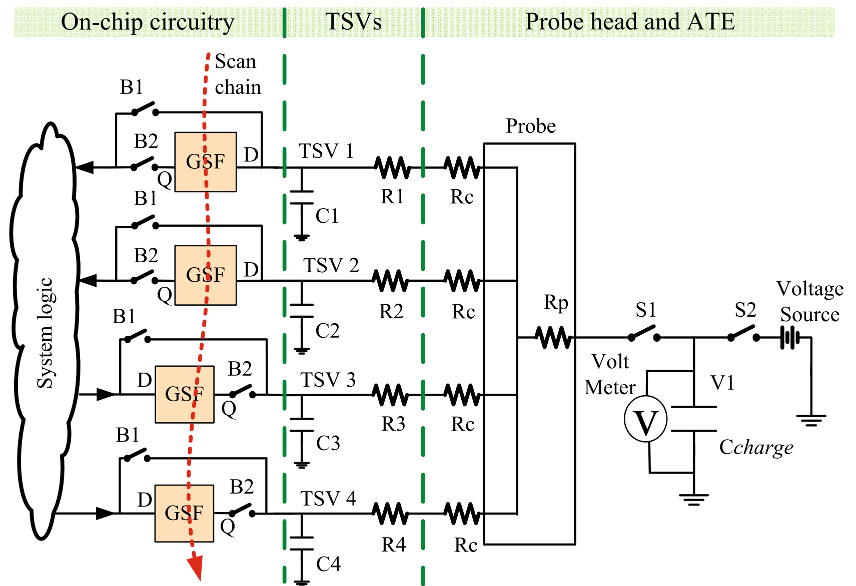

Fig. 2 Circuit model for pre-bond TSV probing

the session are fault-free. Because testing is based on calibration of RC constant of TSV, all TSV defects that cause abnormal resistance or capacitance changes can be detected.

The probe needle remains in contact with a TSV network throughout its test, and the charging and discharging process continues until either all TSVs within the network are identified as fault-free or a certain number of TSVs with resistive defects are pinpointed within the network. All TSV networks are tested in two groups. Networks in a group are tested simultaneously by a probe head containing a large number of needles, each making contact with a single network. Once all contacted networks are tested, the probe head is lifted and repositioned to test the remaining group [15, 16].

SPICE simulation of a TSV probe setup was done using the PTM (predictive technology model [18]) 45nm technology [14–16, 21], and the capacitance charging time as a function of *session size* q, i.e., the number of TSVs tested in parallel, is recorded in Table 1. Note that this number q cannot exceed the measurement resolution r. Also, *test time of a session* in this work only refers to the time to charge the capacitor C_{charge}, same as in [14]. It is related to the session size as shown in Column 2 of Table 1.

The goal of TSV probing is to identify up to a certain number, m, of faulty TSVs in a T TSV network under test when m is the number of redundant TSVs available to repair that TSV network. If the number of identified faulty TSVs exceeds m, then not all faulty TSVs in the network can be repaired and the chip would be discarded. Otherwise, the on-chip redundant TSVs are sufficient to replace all identified faulty ones. This goal of TSV probing can be achieved by testing one TSV at a time with highest resolution. However such high resolution may be unnecessary. Besides, a single-TSV session takes longer test time as shown in Table 1. Moreover, instead of identifying all faulty TSVs, only up to m faulty TSVs need to be pinpointed in a network. Significant test time saving is possible if we test TSVs in parallel without losing the capability of identifying up to m faulty TSVs, and also guarantee that the size of each test session does not exceed the resolution constraint r.

Reference [14] proposes a heuristic to generate test sessions. However, the results are far from being optimal due to the greedy nature of the heuristic. For example, to pinpoint 1 faulty TSV in a 6-TSV network with resolution constraint of $r = 4$, the heuristic based sessions [14] are {1,2,3,4}, {1,5,6}, {2,5}, {3,6}, {4}, where TSVs are numbered 1 through 6. A careful examination shows the last session {4} is unnecessary as the first four sessions uniquely identify any single faulty TSV. For example, if TSV1 is defective then the first two sessions would fail the test, and the third and fourth sessions would pass the test. The passing of third and fourth sessions means TSV2, TSV3, TSV5, and TSV6 are good TSVs. The failing of the second session means

Table 1 Capacitor charging time of parallel TSV test [21]

Number of TSVs tested in parallel (q)	Charging time $t(q)$ in μs
1	8.0
2	5.3
3	4.2
4	3.8

there is at least one defective TSV among TSV1, TSV5 and TSV6. Thus we reach the conclusion of TSV1 being defective by only testing the first four sessions. After removing {4}, the remaining sessions are still non-optimal as an optimal result is {1,2,3}, {1,4,5}, {2,4,6}, {3,5,6}, which further reduces test time by 9.7%, according to Table 1. The above example motivates us to find a way to generate an optimal set of test sessions with minimum test time.

In the above example if TSV 1 is faulty, then all four sessions, {1,2,3}, {1,4,5}, {2,4,6}, {3,5,6}, need to be applied to identify it. But if TSV 6 is faulty, then only the first three sessions need to be applied to pinpoint it. This observation further motivates us to develop an algorithm that can terminate the test as soon as our goal of identification is reached. This concept of abort-on-failure is similar to the concept in [8] which falls in the context of SoC test.

In a matured manufacturing process, TSV yield is expected to be more than 99% [1, 6]. We calculated the probability of ϕ TSVs being defective within a TSV network. The results suggest that this probability decreases dramatically as ϕ increases, by considering different TSV defect distributions. This observation allows us to choose the application order for sessions so that pre-bond TSV test can be terminated as early as possible.

3 Terminology

We define the following terms:

1) *TSV network.* A TSV network consists of all TSVs that are simultaneously connected to a single probe needle during test.
2) *Test session.* During pre-bond test of a TSV network, a test session refers to charging a subset of TSVs (in parallel) within the network under test.
3) *Session size.* Session size q is the number of TSVs in the subset within the TSV network that is being charged in the test session.
4) *Resolution constraint.* Resolution constraint r is an upper bound on the session size.
5) *Maximum number of faulty TSVs to be identified within a network.* This number m equals the number of redundant TSVs available to repair the TSV network being tested.
6) *Test time of a session.* Test time of a session refers to the charging time of C_{charge}. It is related to the session size (refer to Table 1).
7) *Fault map.* Fault map ρ represents positions of all defective TSVs within the network.
8) *Worst fault map.* Worst fault map for a given TSV network refers to a fault map that takes most sessions to identify.

4 ILP Model for Test Session Generation with Specified Identification Capability

We propose an ILP model (named ILP-1) to find a near-optimal set of test sessions. The problem is formulated as follows:

Problem 1 Given the test time $t(q)$ for test session size q, $1 \leq q \leq r$, and the maximum number m of faulty TSVs within a T-TSV network, determine a series of test sessions (none exceeding size r) so that up to m faulty TSVs can be uniquely identified and the total test time is minimized.

A sufficient condition to solve Problem 1 is as follows.

Condition 1 If each TSV (TSV_i) is put in $m+1$ test sessions, $S_1, S_2, \cdots, S_{m+1}$, and an intersection of any pair of these sessions contain only TSV_i, i.e., $S_j \bigcap S_k = TSV_i$ for $j \neq k \in [1, m+1]$, then up to m faulty TSVs within the network can be uniquely identified.

We refer to those $m+1$ sessions satisfying condition 1 as $m+1$ unique test sessions for TSV_i. A unique test session for TSV_i is defined as a session whose intersection with any other session containing TSV_i consists only of TSV_i. We prove the sufficiency of Condition 1 by first stating the following theorem.

Theorem 1 *Given that there are no more than m faulty TSVs within a network. If a good TSV, TSV_i, belongs to $m+1$ unique test sessions, then we can find at least one out of these $m+1$ sessions which consists of only good TSVs.*

We prove Theorem 1 by contraposition.

Proof of Theorem 1 Given there are up to m faulty TSVs within a network. Suppose each unique test session for TSV_i contains at least one faulty TSV. Because of the "unique" identity of these $m+1$ sessions, the faulty TSVs within each session should be different. We conclude that there will be at least $m+1$ faulty TSVs within the network, which is obviously a contradiction to the given condition that says there are at most m faulty TSVs. In other words, at least one unique session for TSV_i would contain only good TSVs. □

Next, we prove that Condition 1 is sufficient for solving Problem 1.

Proof of Sufficiency According to Theorem 1, Condition 1 guarantees that for any good TSV (say, TSV_i) there will be at least one unique session S_j consisting only of good TSVs.

The test result of S_j would then suggest that all TSVs within S_j are fault-free. Thus, we uniquely identify TSV_i as a good TSV. If all good TSVs are identified, then all defective TSVs are too.

Proof of Non-necessity The non-necessity of Condition 1 is proven by an example. Suppose, all TSVs are tested one at a time and so each is uniquely identified. In other words, each TSV is contained in only one session.

The model ILP-1 proposed in this section is based on Condition 1. We first summarize all general constraints for this model as follows:

1) **C1.** Each TSV should reside in at least $m + 1$ sessions.
2) **C2.** The size of a test session ranges anywhere from 0 (empty session) to r.
3) **C3.** We suppose any non-empty session is a unique session for any TSV within it.

An upper bound N_{up} on total number of sessions can be calculated as,

$$N_{up} = \left\lceil \frac{t(1)}{t(r)} \cdot T \right\rceil \tag{1}$$

where r is resolution constraint, $t(1)$ and $t(r)$ are test times for sessions with sizes 1 and r, respectively. If the total number of sessions N is larger than N_{up}, then even if all sessions have size r the total test time is still larger than that of sequential testing with a single TSV per session. This upper bound constrains the maximum number of sessions produced by the ILP model presented next.

A binary variable x_{ij} $(1 \leq i \leq T, 1 \leq j \leq N_{up})$ is defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } TSV_i \text{ is assigned to session } S_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

From **C1**, we have

$$\sum_{j=1}^{N_{up}} x_{ij} \geq m + 1 \tag{3}$$

We define an integer variable L_j, which denotes the size of session S_j,

$$L_j = \sum_{i=1}^{T} x_{ij} \tag{4}$$

From **C2**, we have

$$0 \leq L_j \leq r \tag{5}$$

From **C3**, if $\exists\ x_{ij} \cdot x_{ik} = 1$ for any i and any $j, k \in [1, N_{up}]$, $j \neq k$, then $\sum_{i=1}^{T} x_{ij} \cdot x_{ik} = 1$. **C3** also implies that the intersection of any two distinctly different sessions S_j and S_k should contain no more than a single TSV, thus,

$$\left| S_j \bigcap S_k \right| = \sum_{i=1}^{T} x_{ij} \cdot x_{ik} \leq 1 \tag{6}$$

Constraint (6) is obviously nonlinear. To linearize it, we further introduce a binary variable $z_{ijk} = x_{ij} \cdot x_{ik}$ and two more linear constraints:

$$x_{ij} + x_{ik} - z_{ijk} \leq 1 \tag{7}$$

$$x_{ij} + x_{ik} - 2 \cdot z_{ijk} \geq 0 \tag{8}$$

According to these constraints if $x_{ij} = 0$, then $z_{ijk} \leq \frac{x_{ik}}{2}$. Since both x_{ik} and z_{ijk} are binary variables, z_{ijk} must be 0. If $x_{ij} = 1$, $x_{ik} \leq z_{ijk} \leq 0.5 + 0.5x_{ik}$, then we conclude, $z_{ijk} = x_{ik}$. Thus, constraints (7) and (8) guarantee $z_{ijk} = x_{ij} \cdot x_{ik}$. With z_{ijk}, constraint (6) becomes

$$\sum_{i=1}^{T} z_{ijk} \leq 1 \tag{9}$$

The objective of the ILP model is to minimize the total test time of all sessions:

$$\text{Minimize} \sum_{j=1}^{N_{up}} t(L_j) \tag{10}$$

Both L_j and $t(L_j)$ are variables, we need to linearize the objective function so that any commercial ILP solver can be used. We introduce a new binary variable δ_{jq} $(1 \leq j \leq N_{up}$, $0 \leq q \leq r)$,

$$\delta_{jq} = \begin{cases} 1 & \text{if session } S_j \text{ contains } q \text{ TSVs} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

With Eq. 11, $t(L_j) = \sum_{q=0}^{r} \delta_{jq} \cdot t(q)$. In addition, two new constraints should be included in the model. First,

$$L_j = \sum_{q=0}^{r} q \cdot \delta_{jq} \tag{12}$$

indicates that a session can have 0 to r TSVs. Then,

$$\sum_{q=0}^{r} \delta_{jq} = 1 \tag{13}$$

indicates that the size of a session should be unique. For every session S_j, there will be exactly one value of q for

which $\delta_{jq} = 1$. Therefore, δ_{jq} determines the size of S_j. We can rewrite the objective function (10) as,

$$\text{Minimize} \sum_{j=1}^{N_{up}} \sum_{q=0}^{r} \delta_{jq} \cdot t(q) \quad (14)$$

The test time for any session size $t(q)$ is a constant obtained from SPICE simulation as shown in Table 1. Thus, the objective function is linearized.

The complete ILP model is summarized in Fig. 3. Both the number of variables and number of constraints (a measure of complexity of the model) of the ILP model are $O(N_{up}^2 T)$. For the example of Section 2, ILP-1 produces four test sessions with minimum test time. Note, however, that a globally optimal set of sessions is not guaranteed by ILP-1 since the model is based on Condition 1 which is a non-necessary condition for solving Problem 1.

The sessions generated by ILP-1 guarantee that all TSVs can be identified if the total number of faulty TSVs in a network does not exceed m. If there are more than m faulty TSVs within a network, then the sessions produced by ILP-1 can result in two possible situations. Either, not all TSVs are uniquely identified as either being good or faulty, or, the set of identified faulty TSVs is larger than m but contains all really faulty TSVs. In both situations, we conclude that there are more than m faulty TSVs within the network and the chip can be discarded. Therefore, the sessions provided by ILP-1 can always help make the right decision to either replace the identified bad TSVs or discard the chip as having too many faulty TSVs within a local silicon area.

Objective:

$$\text{Minimize} \sum_{j=1}^{N_{up}} \sum_{q=0}^{r} \delta_{jq} \cdot t(q)$$

Subject to:

1) $x_{ij} + x_{ik} - z_{ijk} \le 1; \quad \forall i \in [1, T], \forall j \ne k \in [1, N_{up}]$

2) $x_{ij} + x_{ik} - 2z_{ijk} \ge 0; \quad \forall i \in [1, T], \forall j \ne k \in [1, N_{up}]$

3) $\sum_{i=1}^{T} z_{ijk} \le 1; \ \forall j \ne k \in [1, N_{up}]$

4) $L_j = \sum_{i=1}^{T} x_{ij}; \ \forall j \in [1, N_{up}]$

5) $0 \le L_j \le r; \ \forall j \in [1, N_{up}]$

6) $\sum_{j=1}^{N_{up}} x_{ij} \ge m+1; \ \forall i \in [1, T]$

7) $L_j = \sum_{q=0}^{r} \delta_{jq} \cdot q; \ \forall j \in [1, N_{up}]$

8) $\sum_{q=0}^{r} \delta_{jq} = 1; \ \forall j \in [1, N_{up}]$

Fig. 3 ILP-1 for finding near-optimal test sessions with specified identification capability

5 A Fast TSV Identification Algorithm

The test sessions generated based on Condition 1 guarantee to handle any possible fault map. But for some fault maps, only part of the sessions need to be applied. Let us reconsider the example of Section 2. For $T = 6$, $m = 1$, $r = 4$, only 3 out of 4 test sessions were needed to identify the faulty TSV_6, which would further save 25% from test time. In this section, we now propose a faster TSV identification algorithm to further speed up the pre-bond TSV test in two aspects. First, during the identification process, any "currently unnecessary" session is skipped. Second, TSV test is terminated as soon as either all TSVs have been identified or the number of identified faulty TSVs exceeds m because the chip can be discarded due to lack of redundant TSVs and further test is deemed useless.

Pseudo-code of the algorithm is shown in Fig. 4 where parameter t represents the test time of a session. The algorithm starts by initializing three empty lists named "Good", "Bad", and "F_C", respectively. The "Good" and "Bad" lists contain the identified good and faulty TSVs, respectively. The faulty candidate list "F_C" is used to contain any

```
Algorithm Fast_TSV_Identification (All_sessions, T, m, t)
1.  Good=[ ]; Bad=[ ]; F_C=[ ]; test_time=0; tested_sessions=0;
2.  foreach session in All_sessions
        // Skip any currently unnecessary test session
3.      if (all TSVs in session have been identified) or
           (there is at least one bad TSV in session)
4.          Continue;
5.      test_time+=t(session);  // Test time accumulation
6.      tested_sessions+=1;     // Test session accumulation
        // Handle a passing session
7.      if session is tested as being good
8.          Add all TSVs in session to Good;
9.          foreach FC_session in F_C
10.             Remove any good TSV from FC_session;
11.             if length(FC_session)==1
12.                 Add the TSV in FC_session to Bad;
13.                 Remove the entire FC_session from F_C;
        // Handle a failing session
14.     else if session is tested as being bad
15.         Remove any good TSV from session;
16.         if length(session)==1
17.             Add the TSV in session to Bad;
18.         else
19.             Append session to F_C;
        // Termination conditions
20.     if ((length(Good)+length(Bad))==T or (length(Bad)>=m+1)
21.         Break;
22. Return test_time, tested_sessions;
```

Fig. 4 A dynamically optimized TSV identification algorithm

failing session. The algorithm enumerates all given test sessions generated by either ILP-1 [33] or a heuristic method [14] and skips any "currently unnecessary" session where either all TSVs in the session have been identified so far or there is at least one identified bad TSV in the session. A "currently unnecessary" session does not provide any information on TSV identification. We define a *fault map* ρ as a set of positions of defective TSVs within a TSV network. Although a session may be "currently unnecessary" for identifying some fault maps of a TSV network, it could be essential for identifying other fault maps of the same TSV network. So, none of the "currently unnecessary" sessions can be deleted. If a session is not skipped, it will be applied. If tests in the session pass, then all TSVs covered by the session are added to "Good", which is then used to refine "F_C" by removing all identified good TSVs (see Line 10 of Fig. 4). Now, if any failing session in "F_C" contains only one TSV, then that TSV is identified as defective and added to "Bad." If a session fails the test, "Good" is again utilized to refine this failing session (line 15 of Fig. 4). If the session after refinement contains only one TSV, that TSV is added to "Bad." Otherwise, the refined failing session is appended to "F_C." The above procedure terminates as soon as any condition shown on line 20 in Fig. 4 is satisfied.

Table 2 shows the results of the proposed algorithm applied to various TSV networks. Column 1 shows parameters T (network size), m (redundant TSVs available to repair the network) and r (resolution). Column 2 gives the number of faulty TSVs (ϕ) within the network. Column 3 shows the total number of sessions and total test time (in 10^{-5}s) for exhaustive application of sessions optimized by ILP-1. For given ϕ, we enumerate all possible fault maps and obtain the test time and number of tested sessions using the algorithm of Fig. 4. Column 4 shows the average number of tested sessions and average test time for identifying all fault maps containing ϕ faulty TSVs. Column 5 shows the relative reduction in Column 4 over Column 3. Column 6 shows the maximum number of sessions tested and the corresponding test time for identifying a fault map. Column 7 shows the relative reduction in Column 6 over Column 3.

We make four observations from Table 2. First, the average number of tested sessions and average test time are much lower than the total number of sessions and total test time for any $\phi \leq m$ (reparable TSV network) or any $\phi > m$ (irreparable TSV network). For example, the average percentage reduction reaches 68.0% for parameters $T = 15$, $m = 4$, $r = 3$, and $\phi = 0$. On average, the proposed algorithm greatly speeds up the pre-bond TSV identification process. Note that the average percentage reduction in number of sessions is sometimes slightly different from the average percentage reduction in test time. This is because the test time of sessions can vary depending on the number of TSVs being tested in those sessions, as shown in Table 1. Second, as ϕ increases the average percentage reduction decreases. This is expected as pinpointing a larger number of faulty TSVs within a TSV network generally requires more sessions to be applied at the cost of more time. Third, in most cases even the maximum number of tested sessions is less than the total number of sessions. Fourth, as expected, the maximum number of applied sessions increases as ϕ increases for a given TSV network. From Column 7, reduction in the worst-case can be small for large ϕ, requiring all sessions to identify a fault map. This scenario occurs when fault map contains m or more faulty TSVs. The probability of such large number of faulty TSVs within a small localized silicon area may be negligible for a mature manufacturing process. Thus, the worst-case percentage test time reduction could be quite significant.

To sum up the preceding discussion, the proposed TSV identification algorithm has two main advantages. First, the average number of tested sessions and test time are guaranteed to be small factions of total sessions and test time. Second, even for the worst fault map, for which most sessions are needed, not all sessions may be used, mainly due to the non-necessity nature of Condition 1.

6 Probabilistic Analysis of Different Number of Faulty TSVs within a TSV Network

In this section, we analyze the effects of varying the number of faulty TSVs within a network. TSV defect distributions can be broadly classified as two types, namely independent defect distribution [35] and clustered defect distribution [23, 35]. For independent TSV defect distribution, the failing probabilities of TSVs are independent from each other and the probability of ϕ faulty TSVs within a T-TSV network can be expressed as:

$$P(\phi) = \binom{T}{\phi} p^{\phi}(1-p)^{T-\phi} \tag{15}$$

where p is average TSV failing probability.

Defects clustering models a scenario where the presence of a defective TSV increases the probability of more defects in close vicinity [23, 35]. Reference [35] assumes, 1) a defect cluster center [23, 35] consists of only a single defective TSV, and 2) the failure rate of TSV_i is inversely proportional to the distance from the defect cluster center. Thus,

$$p(TSV_i) = p \cdot (1 + (\frac{1}{d_{ic}})^{\alpha}) \tag{16}$$

where $p(TSV_i)$ is the failing probability of TSV_i, d_{ic} is the distance of TSV_i from the defect cluster center and α is a

Table 2 Exhaustive [33] and dynamically optimized (Fig. 4) application of TSV test sessions constructed by ILP-1

T, m, r	Number of faulty TSVs (ϕ)	Optimum exhaustive test [33] (# sessions, time$\times 10^{-5}$s)	Dynamically optimized test: Average test sessions (# used, time$\times 10^{-5}$s)	Average reduction (sessions, time)	Worst-case sessions (# used, time$\times 10^{-5}$s)	Worst-case reduction (sessions, time)
	0		(5.0, 2.10)	(37.5%, 37.5%)	(5, 2.10)	(37.5%, 37.5%)
8, 2, 3	1	(8, 3.36)	(5.3, 2.25)	(32.8%, 32.8%)	(6, 2.52)	(25.0%, 25.0%)
	2		(6.4, 2.71)	(19.1%, 19.1%)	(8, 3.36)	(0.0%, 0.0%)
	3		(7.5, 3.17)	(5.3%, 5.3%)	(8, 3.36)	(0.0%, 0.0%)
	0		(7.0, 2.94)	(56.2%, 56.2%)	(7, 2.94)	(56.2%, 56.2%)
	1		(7.5, 3.14)	(53.1%, 53.1%)	(9, 3.78)	(43.7%, 43.7%)
12, 3, 3	2	(16, 6.72)	(8.7, 3.65)	(45.5%, 45.5%)	(12, 5.04)	(25.0%, 25.0%)
	3		(10.3, 4.32)	(35.5%, 35.5%)	(14, 5.88)	(12.5%, 12.4%)
	4		(11.8, 4.97)	(25.9%, 25.9%)	(16, 6.72)	(0.0%, 0.0%)
	0		(8.0, 3.36)	(68.0%, 68.0%)	(8, 3.36)	(68.0%, 68.0%)
	1		(9.6, 4.03)	(61.6%, 61.6%)	(14, 5.88)	(44.0%, 44.0%)
15, 4, 3	2	(25, 10.50)	(11.1, 4.68)	(55.3%, 55.3%)	(17, 7.14)	(32.0%, 32.0%)
	3		(12.6, 5.33)	(49.2%, 49.2%)	(20, 8.40)	(20.0%, 20.0%)
	4		(14.3, 6.03)	(42.5%, 42.5%)	(23, 9.66)	(8.0%, 8.0%)
	5		(15.8, 6.66)	(36.5%, 36.5%)	(25, 10.50)	(0.0%, 0.0%)
	0		(9.0, 3.42)	(64.0%, 63.9%)	(9, 3.42)	(64.0%, 63.9%)
	1		(10.8, 4.10)	(56.8%, 56.7%)	(15, 5.69)	(40.0%, 39.9%)
20, 4, 4	2	(25, 9.50)	(12.3, 4.68)	(50.6%, 50.6%)	(18, 6.83)	(28.0%, 27.9%)
	3		(13.9, 5.31)	(44.0%, 44.0%)	(21, 7.97)	(16.0%, 15.9%)
	4		(15.1, 5.76)	(39.3%, 39.3%)	(24, 9.11)	(4.0%, 3.9%)
	5		(18.0, 6.85)	(27.8%, 27.8%)	(25, 9.49)	(0.0%, 0.0%)

clustering coefficient. Larger values of α imply less clustering. As $\alpha \to \infty$, the defect distribution tends to become an independent distribution.

The clustered model requires the TSV location information. Since the number of TSVs within a network is typically less than 20, we consider a TSV network as a 5×5 matrix of 25 possible locations separated by the TSV pitch (minimum separation between a TSV pair). The value 5 is chosen based on the ratio of the size of a probe needle and realistic TSV pitch [17, 22]. We randomly choose T locations of the matrix to construct a T TSV network. We then employ Eq. 16 to analyze the probability of ϕ TSVs being defective within a network. Because exactly modeling the TSV defect clustering effect across different TSV networks is difficult, following Zhao et al. [35] we assume that each TSV network has only one defect cluster center and defect clusters within different networks do not interfere with each other. This yield modeling approach tries to mimic the defect clustering effects of silicon while maintaining simplicity.

Table 3 shows probabilities for different values of ϕ for a 15-TSV network. We vary the TSV yield from 98% to 99.5% to accommodate different levels of maturity of the manufacturing processes. The clustering coefficient α is set as 1 and 2, similar to [35] and [9]. Note the values under clustered defect distribution are averaged over 100 Monte Carlo runs, with each run randomly placing 15 TSVs on a 5-by-5 matrix. By doing this, we try to simulate all possible TSV placements in real silicon.

Three observations are summarized from Table 3. First, irrespective of the defect distribution, the probability of $\phi = 0$ is largest and is even much larger than the sum for

Table 3 Probability of different number ϕ of failing TSVs within a 15-TSV network

Defect distribution	TSV yield	Number of faulty TSVs, ϕ: 0	1	2	≥3
	99.5%	92.76%	6.99%	0.25%	0.00%
Independent	99.0%	86.01%	13.03%	0.92%	0.04%
	98.0%	73.86%	22.60%	3.23%	0.31%
Clustered	99.5%	92.76%	6.70%	0.35%	0.19%
α=1	99.0%	86.01%	12.07%	1.26%	0.66%
	98.0%	73.86%	19.55%	4.16%	2.43%
Clustered	99.5%	92.76%	6.78%	0.31%	0.15%
α=2	99.0%	86.01%	12.39%	1.13%	0.47%
	98.0%	73.86%	20.60%	3.81%	1.73%

the rest of the cases with $\phi > 0$. Second, the sum of probabilities for $\phi = 0$ and $\phi = 1$ is almost 1 in all situations, and the probability of $\phi \geq 3$ is very low. Third, as TSV yield decreases, probability of $\phi = 0$ decreases. Motivated by above observations, we propose to sort the order of test sessions to reduce the expectation of pre-bond TSV test time, as explained in the next section.

7 An Iterative Greedy Procedure for Test Session Scheduling

We express the expectation $E(\Gamma)$ of test time (Γ) of a TSV network as follows:

$$E(\Gamma) = \sum_{\forall \rho} \gamma(\rho) P(\rho) \tag{17}$$

where ρ is any specific fault map in the TSV network as defined in Section 3, $\gamma(\rho)$ is the identification time to determine ρ using the fast TSV identification algorithm of Section 5 [31], and $P(\rho)$ is the occurrence probability of ρ. Note that $\rho = \emptyset$ or $|\rho| = \phi = 0$ means that all TSVs within the network are fault-free. Going forward from Problem 1 of Section 4, we formulate the next two problems.

Problem 2 Given a series of N test sessions that can uniquely identify up to m faulty TSVs within a TSV network of T TSVs, find an optimal order to apply those sessions so that the expectation of pre-bond TSV test time is minimized for this TSV network.

To solve Problem 2, a straightforward method is to find all permutations of test sessions, and for each permutation calculate the test time expectation using Eq. 17. The permutation which yields minimum $E(\Gamma)$ would be the selected choice. However, there can be $N!$ permutations and $2^T - 1$ fault maps. So the identification algorithm [31] must be run $N! \cdot (2^T - 1)$ times, which is highly time-consuming even for a small network. Fortunately, we notice that the probability of different number of faulty TSVs is inversely proportional to ϕ. Specifically,

$$\sum_{|\rho|=i} P(\rho) \gg \sum_{|\rho|=j} P(\rho) \text{ for any } i < j \tag{18}$$

where $\sum_{|\rho|=i} P(\rho) = P(\phi = i)$ and $\sum_{|\rho|=j} P(\rho) = P(\phi = j)$.

Motivated by the fact that $P(\rho)$ is large for small $|\rho|$ and decreases dramatically as $|\rho|$ increases, if we can reduce $\gamma(\rho)$ for small $|\rho|$ the test time expectation should be greatly reduced. For example, the probability of $P(\rho = \emptyset)$, i.e., all TSVs in network are good, dominates. In case of $\rho = \emptyset$, all TSVs are identified as good TSVs as long as the already tested sessions covered all TSVs. Based on this observation, Problem 3 is formulated as follows.

Problem 3 Given N test sessions that can uniquely identify up to m faulty TSVs within a network of T TSVs, select M out of N sessions such that these M sessions cover each TSV at least once and the total test time of the selected M sessions is minimum.

Problem 3 can be solved by constructing an ILP model, named "ILP-2" to differentiate it from ILP-1 of Section 4. We introduce a variable P_j, $j \in [1, N]$, where

$$P_j = \begin{cases} 1 & \text{if session } S_j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Then, ILP-2 is described as follows:

$$\text{Objective: Minimize } \sum_{j=1}^{N} t(L_j) \cdot P_j$$

Subject to constraints: each $TSV_i, i \in [1, T]$, is tested at least once by the selected sessions.

L_j represents the size of session S_j, and $t(L_j)$ the test time of S_j, which is a constant for a given L_j. Considering the complexity of ILP-2, its numbers for variables and constraints are $O(NT)$ and $O(T)$, respectively. In all our experiments, ILP-2 was solved in 1 second or less. The sessions covering all TSVs with minimal test time are sorted according to Problem 3 and applied before the rest of the sessions. This would reduce $\gamma(\rho = \emptyset)$ and thus reduce the test time expectation.

As can be seen from Table 3, $P(\phi = 1)$ is not negligible. If we can further reduce $\gamma(\rho)$ with $|\rho| = 1$, the test time expectation should be further reduced. The N test sessions in Problems 2 and 3 can be produced by either the ILP-1 [33] of Section 4 or the heuristic method [14]. Sessions produced by both methods have characteristics such that if each TSV is covered (or tested) by at least two sessions, any single faulty TSV can be uniquely identified. To reduce $\gamma(\rho)$ with $|\rho| = 1$, we can retain the M sessions produced by ILP-2, and find M_1 additional sessions from the remaining $N - M$ sessions such that the $M + M_1$ sessions will cover each TSV at least twice and the test time of the M_1 sessions is minimum.

Next we explain how ILP-2 can be again utilized to find the M_1 sessions mentioned above. We first count the times each TSV is covered by first M sessions, and place the TSVs that have been covered only once in a list named "TSV_set". ILP-2 is now utilized to find M_1 sessions out

of the $N - M$ sessions such that each $TSV_i, i \in TSV_set$ is covered (or tested) at least once by these M_1 sessions and the total test time of these M_1 sessions is minimum. The resulting M_1 sessions will be sorted and applied directly after the M sessions. These $M + M_1$ sessions first guarantee $\gamma(\rho = \emptyset)$ is minimized and based on this premise further minimize $\gamma(\rho)$ with $|\rho| = 1$ (simply represented as $\gamma(|\rho| = 1)$). Similar procedure can be repeated for further reduction of $\gamma(|\rho| = 2)$, $\gamma(|\rho| = 3)$, $\cdots$, until $\gamma(|\rho| = m)$.

We summarize the overall procedure for session sorting in Fig. 5. ILP-2 is iteratively utilized in *Test_session_sorting* procedure with each execution trying to find a subset of sessions from "$Original_sessions$" so as to cover all TSVs within "TSV_set" at least once with minimum time. The greedy nature of our procedure is obvious since it puts higher priority on reducing $\gamma(\rho)$ with smaller $|\rho|$. The run time of the procedure is (almost) equal to the run time of ILP-2 multiplied by the number of times ILP-2 is executed, which is determined by "$Stop_index$" in Fig. 5. Note that "$Stop_index$" can be set to any value from 1 to $m + 1$. When "$Stop_index$" is 1, *Test_session_sorting* will only reduce $\gamma(\rho = \emptyset)$ by finding M sessions that cover all TSVs at least once with minimum time. When "$Stop_index$" is $m+1$, the procedure will first reduce $\gamma(\rho = \emptyset)$, then reduce $\gamma(|\rho| = 1)$, and then reduce $\gamma(|\rho| = 2)$, $\cdots$, all the way up to reducing $\gamma(|\rho| = m)$.

Procedure *Test_session_sorting*

Initialization:
Original_sessions = All the sessions;
TSV_set = All the TSVs within network;
Sorted_sessions = [ ];
Stop_index = any integer ∈ [1, m+1];
k=1;
Iterative Execution:
while (*k* <= *Stop_index*) **do**
{
Step1: Use ILP model 2 to find a subset of sessions from *Original_sessions* which cover each TSV within *TSV_set* at least once with minimum test time;
Step2: Append all sessions produced by Step1 to the end of *Sorted_sessions*;
Step3: Remove these sessions produced by Step1 from *Original_sessions*;
Step4: Based on *Sorted_sessions*, calculate the times each TSV is covered;
Step5: Set *TSV_set* ← TSVs which have been covered by only *k* times;
Step6: *k*++;
}
Return Final Results:
Append *Original_sessions* to the end of *Sorted_sessions*;
Return *Sorted_sessions*;

Fig. 5 Iterative session sorting through the use of ILP-2

8 A Three-Stage Test Time Optimization Simulator

In this section we propose a 3-Stage test time Optimization Simulator (SOS3). Stage 1 of SOS3 is ILP-1 of Section 4 for test session generation. Note that we choose ILP-1 instead of the heuristic method [14] because test sessions generated by both methods have exactly the same TSV identification capability, however, ILP-1 generates fewer sessions and is more test time-efficient as will be illustrated in the next section.

Stage 2 of SOS3 is the iterative greedy procedure for session sorting that implicitly uses ILP-2 of Section 7 (Fig. 5). This procedure accepts N sessions from Stage 1 as input and sorts the sessions to reduce test time expectation.

Stage 3 is a fast TSV identification algorithm. This algorithm takes the sorted list of sessions as input and finishes the identification process as soon as any termination condition becomes true. By integrating session sorting procedure and fast TSV identification algorithm in SOS3, the pre-bond TSV probing can be terminated as soon as possible with largely reduced test time expectation.

Figure 6 illustrates the overall flow of SOS3. The input to SOS3 contains three pieces of information: 1) TSV and TSV network information, 2) probing technology information, and 3) on-chip TSV redundancy information. The outputs of SOS3 are: 1) the sorted list of sessions, 2) identified TSVs, 3) test time expectation and 4) expectation of number of sessions to be applied.

9 Experimental Results

9.1 Test Time Comparison Between ILP-1 and Heuristic Method

In this subsection, we compare the total test time and total number of sessions for three methods: ILP-1, a heuristic method [14], and sequentially testing all TSVs one at a time. For comparison, similar to [14], we simply assume that all generated sessions are applied. The *relative test time ratio* in this section refers to the ratio of total test time to the total time of sequential TSV testing. Note again the *test time* in this work considers only the time to charge the capacitor and does not account for the time to physically move the probe [14]. Our charts only show cases where relative test time ratio is smaller than 100%. For the cases not shown, sequentially testing all TSVs one at a time was more time-efficient than the other two methods. The ILP-1 solution was obtained from a commercial solver named CPLEX from IBM [4]. In all situations, the run time of the solver was under 40 seconds.

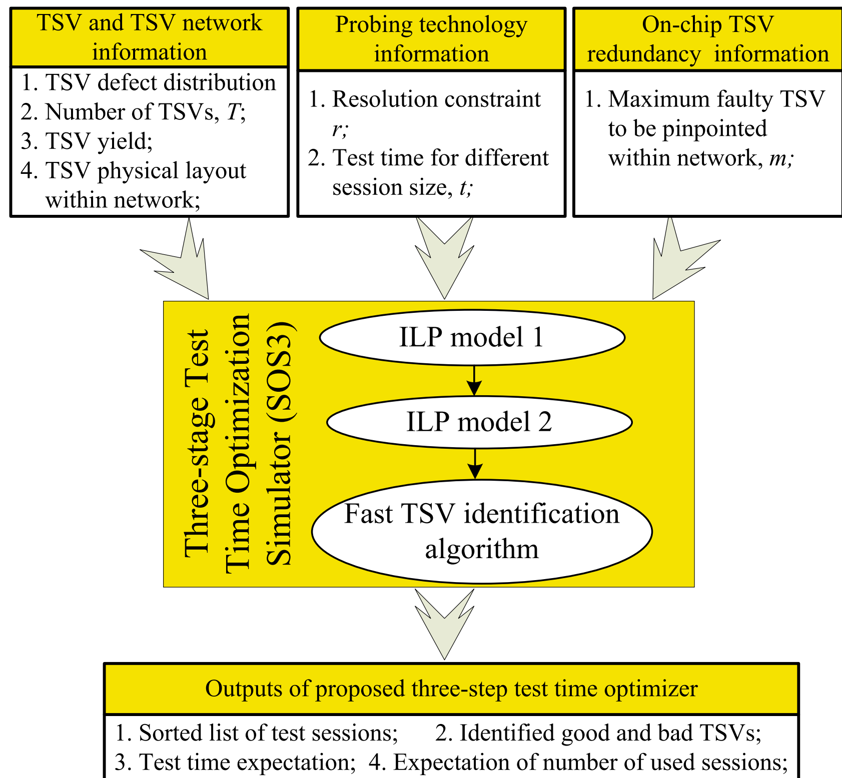

Fig. 6 Three-stage test time optimization simulator (SOS3)

Figure 7 shows the relative test time ratio for both ILP-1 and heuristic method considering different resolution constraint $r \in [2, 4]$ and different value of $m \in [1, 4]$ in a 20-TSV network. Note that the test time from ILP-1 is mostly lower and never higher than the result of heuristics. The number corresponding to each pair of bars represents the relative test time improvement of ILP-1 over heuristic.

In Fig. 7, for any given r, the relative test time ratio for both methods increases as m increases since pinpointing a larger number of defective TSVs requires more sessions and longer test time. For the same m, larger r offers smaller test time ratio because now a session can hold more TSVs with less test time. Moreover, the total number of sessions generally decreases for larger r. Also, note that ILP-1 always helps reduce test time further compared to the heuristic method, and the improvement generally increases as m increases. This demonstrates that for small m, the heuristic may behave well but as m increases, result of the heuristic method deviates farther away from the optimal result. The largest improvement of ILP-1 over heuristic method reaches 25.8% which happens when $T = 20$, $m = 3$ and $r = 4$.

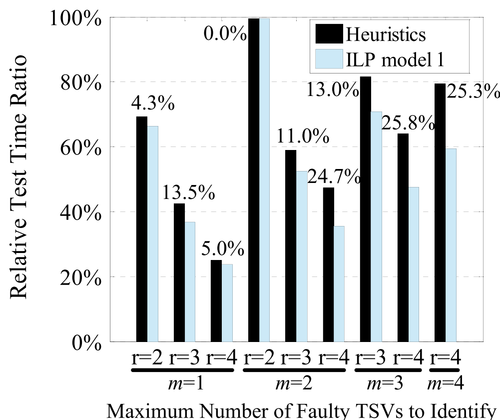

Fig. 7 Ratios of test time for a 20-TSV network obtained by ILP-1 (ILP model 1) and heuristics [14], with respect to test time for sequentially testing all TSVs one at a time

Figure 8 examines the impact of number T of TSVs within a network on relative test time ratio. We simulated four networks of different sizes with $m \in [1, 4]$ and r fixed at 3. We observe that ILP-1 reduces test time compared to heuristics regardless of the value of T. This relative improvement can be as large as 38.2% as shown on top of the pair of bars corresponding to $T = 11$, $m = 2$. It is also interesting to observe that the relative test time ratio of ILP-1 remains consistent across different values of T for a given m. While for the heuristics, the test time ratio varies for different network sizes. For example, when $m = 2$, the relative test time ratio of heuristics for an 11-TSV network is much larger than those for other networks. The unstable performance of the heuristic method is mainly due to its greedy nature of generating test sessions. These observations suggest that the proposed ILP-1 model is more robust across variations of TSV network parameters and thus could eliminate the need for redesign and optimization of each

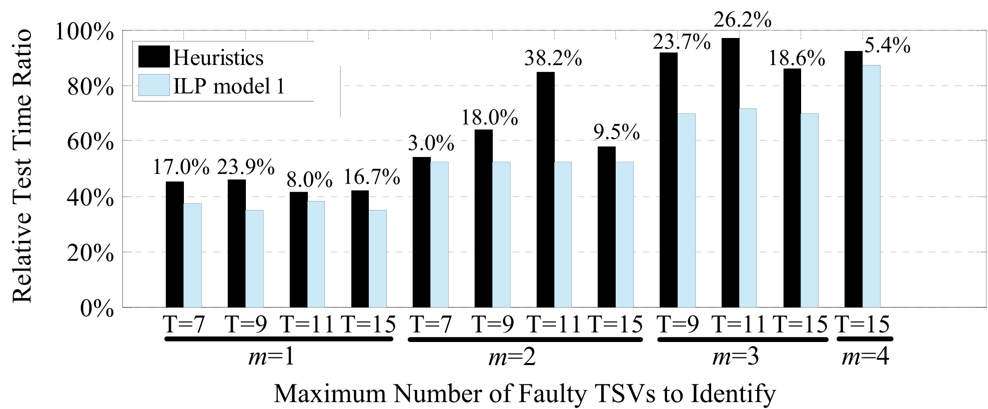

Fig. 8 Test time comparison between ILP-1 (ILP model 1) and heuristic [14] for resolution constraint $r = 3$

individual TSV network on chip as required in the heuristic method [14].

We show the total number of test sessions generated by the two methods for four different networks in Fig. 9 where r is fixed at 4 and $m \in [1, 4]$. The number on top of each pair of bars represents the relative reduction in the number of sessions for ILP-1 over heuristics. As expected, generally a smaller number of sessions is produced for smaller m. For a larger TSV network, it is possible to reduce test time with number of test sessions larger than the total number of TSVs (i.e., T). As can be seen, the ILP-1 sometimes produces the same number of sessions as the heuristic. However, our experimental results demonstrate that though the number of sessions produced by both methods may be the same, the sessions themselves are different. The ILP-1 sessions are guaranteed to be more time-efficient. For example, for $T = 8, m = 1, r = 4$ and $T = 8, m = 2, r = 4$, ILP-1 produces same number of sessions as the heuristic but with test time reduced by 5.8% and 12.5%, respectively. In most cases, ILP-1 helps reduce total number of sessions compared to the heuristic. For example, for $T = 16, m = 4, r = 4$, 4 out of 24 sessions can be further eliminated representing 16.7% relative reduction.

9.2 Impact of Session Sorting on Test Time and Test Time Evaluation of SOS3

We now compare the expectations of test time and the number of tested sessions for two simulators: SOS2 (2-Stage test time Optimization Simulator) and SOS3. The only difference between SOS2 and SOS3 is that the ILP-2 based iterative session sorting procedure is not used in the former. By comparing these two simulators, we illustrate the importance of session sorting in test time reduction. Note that we do not compare the test time expectation of SOS3 with that of the heuristic method [14], the reasons being, first, ILP-1 returns smaller number of sessions and less test time and, second, the ILP-2 based session sorting procedure in combination with the fault identification algorithm helps reduce test time expectation further.

The expectation of test time $E(\Gamma)$ is estimated as follows:

$$E(\Gamma) = \begin{cases} \sum_{|\rho|<2} \gamma(\rho)P(\rho) + TT \sum_{|\rho|\geq 2} P(\rho) & \text{if } m = 1 \\ \sum_{|\rho|\leq 2} \gamma(\rho)P(\rho) + TT \sum_{|\rho|\geq 3} P(\rho) & \text{if } m \geq 2 \end{cases} \quad (20)$$

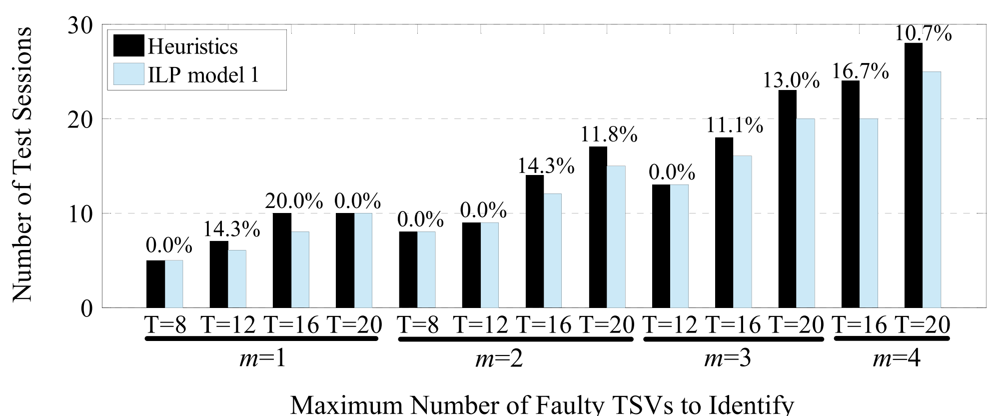

Fig. 9 Comparison of number of sessions between ILP-1 (ILP model 1) and heuristic [14] for resolution constraint $r = 4$

where TT represents the total time of testing all the sessions produced by ILP-1 [33]. Similarly, the expectation of number of tested sessions $E(S)$ is estimated as,

$$E(S) = \begin{cases} \sum_{|\rho|<2} \eta(\rho)P(\rho) + N \sum_{|\rho|\geq 2} P(\rho) & \text{if } m = 1 \\ \sum_{|\rho|\leq 2} \eta(\rho)P(\rho) + N \sum_{|\rho|\geq 3} P(\rho) & \text{if } m \geq 2 \end{cases} \quad (21)$$

where $\eta(\rho)$ represents the number of tested sessions for identification of fault map ρ using TSV identification algorithm [31]. N is the total number of sessions produced by ILP-1 [33].

Let us examine equations (20) and (21). For TSV network with $m = 1$, we simply assume any fault map with $|\rho| \geq 2$ will cause the entire sessions to be tested. This is because sessions generated for $m = 1$ are not intended to identify more than one faulty TSV. According to Table 3, $P(\phi \geq 2) = \sum_{|\rho|\geq 2} P(\rho)$ is small. Such a low probability has negligible impact on expectation calculation. For TSV networks with $m \geq 2$, we assume that all sessions need to be applied to identify fault maps with $|\rho| \geq 3$. This is because it generally takes most of the sessions to identify large number of defective TSVs (like $|\rho| \geq 3$). Moreover, $P(\phi \geq 3) = \sum_{|\rho|\geq 3} P(\rho)$ is quite small according to Table 3. Such a low probability has little impact on the expected value.

Using Eqs. 20 and 21, we compare SOS2 and SOS3 for various values of T, m and r. The commercial ILP solver CPLEX [4] is again used in these experiments. For all simulations, SOS2 and SOS3 provide the outputs in less than one minute. The expectations of number of test sessions and test time for both SOS2 and SOS3 are shown in Tables 4 and 5, respectively. We provide an insightful evaluation of SOS3 by varying TSV yield from a low of 98.0% to a practically expected high of 99.5%. Due to space limitation, we have not included simulation result for 99.0% TSV yield. Nonetheless, those results always lie somewhere between those for 98.0% and 99.5% TSV yields. Note we only show the results under clustered defect distribution with $\alpha = 1$, since the results are very similar for the other two defect distributions in Table 3.

Table 4 shows expected number of sessions and Table 5 shows expected test time. The first columns in both tables give TSV network parameters T, m and r. The second columns show the total number of sessions and total test time of all sessions, respectively, produced by ILP-1 [33]. Columns 3 and 4 contain triplets of values, namely, SOS2 result, SOS3 result, and percent improvement of SOS3 result over SOS2 result. Note that each value is the averaged result from 100 Monte Carlo runs, where a run corresponds to a different random TSV placement over a 5×5 matrix. By doing this we try to consider all TSV placement possibilities.

We make several observations from Tables 4 and 5. First, both expectation of number of test sessions and expectation of test time are only small fractions of total number of sessions and total test time, respectively. This demonstrates that SOS2 and SOS3 greatly speed up the pre-bond TSV identification process and thus reduce pre-bond TSV test cost. From Table 4, the total number of sessions can reach as high as 25, but the expectation of number of tested sessions is no more than 8.7 for SOS3. For TSV network with $T = 20$, $m = 4$ and $r = 4$, the test time expectation of SOS3 result is only 31.3% of the total test time for 98.0% TSV yield.

Table 4 Expectation of number of tested sessions, defect clustering coefficient $\alpha = 1$, data shows (sessions for SOS2, sessions for SOS3, percentage reduction by SOS3)

Parameters	Total number	Expected number of tested sessions, $E(S)$	
T, m, r	of sessions, N	TSV yield = 99.5%	TSV yield = 98.0%
8, 1, 2	8	(5.0, 4.0, 19.3%)	(5.2, 4.3, 17.4%)
8, 2, 3	8	(5.0, 4.0, 19.4%)	(5.1, 4.2, 17.6%)
11, 1, 2	11	(8.0, 6.0, 24.2%)	(8.2, 6.4, 21.8%)
11, 2, 3	11	(6.0, 4.1, 31.6%)	(6.3, 4.6, 26.8%)
15, 2, 2	23	(10.0, 8.1, 19.3%)	(10.6, 8.7, 17.0%)
15, 3, 3	20	(7.1, 6.1, 13.5%)	(7.8, 6.9, 11.2%)
16, 3, 4	16	(6.1, 5.2, 15.3%)	(6.9, 6.1, 11.6%)
16, 4, 4	20	(5.2, 4.3, 17.9%)	(6.2, 5.4, 12.8%)
20, 3, 4	20	(9.1, 6.3, 31.1%)	(9.9, 7.5, 24.2%)
20, 4, 4	25	(9.2, 6.3, 31.3%)	(10.3, 7.8, 24.5%)

Table 5 Expectation of test time (μs), defect clustering coefficient $\alpha = 1$, data shows (test time for SOS2, test time for SOS3, reduction by SOS3)

Parameters	Test time of all	Expectation of test time, $E(\Gamma)$	
T, m, r	sessions, TT	TSV yield = 99.5%	TSV yield = 98.0%
8, 1, 2	42.4	(26.6, 21.5, 19.3%)	(27.6, 22.8, 17.4%)
8, 2, 3	33.6	(21.0, 16.9, 19.4%)	(21.5, 17.7, 17.6%)
11, 1, 2	58.3	(42.5, 32.1, 24.2%)	(43.6, 34.1, 21.8%)
11, 2, 3	46.2	(25.4, 17.3, 31.6%)	(26.4, 19.3, 26.8%)
15, 2, 2	121.9	(53.3, 43.0, 19.3%)	(56.1, 46.5, 17.0%)
15, 3, 3	84.0	(29.9, 25.8, 13.5%)	(32.7, 29.0, 11.2%)
16, 3, 4	60.8	(23.4, 19.8, 15.3%)	(26.3, 23.2, 11.6%)
16, 4, 4	76.0	(19.9, 16.3, 17.9%)	(23.8, 20.8, 12.8%)
20, 3, 4	76.0	(34.7, 23.9, 31.1%)	(37.9, 28.7, 24.2%)
20, 4, 4	95.0	(35.0, 24.0, 31.3%)	(39.3, 29.7, 24.5%)

Our second observation is that both expectation of test time and number of tested sessions increase for lower TSV yield. This is because the probability of having a larger number of defective TSVs within a network increases as TSV yield drops. Identifying a larger number of faulty TSVs typically takes more sessions and longer test time from both SOS2 and SOS3.

A third observation is that SOS3 further reduces the expectation values compared to SOS2. For example, for $T = 11$, $m = 2$ and $r = 3$, SOS3 reduces test time expectation by 31.6% compared to SOS2 for 99.5% TSV yield.

The fourth and final observation is that the improvement of SOS3 over SOS2 decreases as TSV yield decreases. This is because the ILP-2 based session sorting procedure within SOS3 puts higher priority on reducing the identification time for smaller value of $|\rho|$. To identify fault map with large $|\rho|$ (e.g, $|\rho| \geq 2$), SOS3 does not have much advantage over SOS2. As TSV yield decreases, $P(\phi \geq 2) = \sum_{|\rho| \geq 2} P(\rho)$ increases and the advantage of SOS3 also slightly decreases. However, even the smallest improvement of SOS3 over SOS2 is 11.2% in both tables. Nevertheless, this comparison between SOS2 and SOS3 illustrate the significance of session sorting in reducing test time expectation.

10 Limitations and Future Work

The TSV probing technique originally proposed in [14–16] has several potential advantages. However, one possible concern in practical utilization of this technique is that the planarity of TSV tips may not be consistent and could prevent a large probe needle from making good electrical contact with all TSVs of the network, simultaneously. Still, the great benefit offered by the probing method serves as a strong motivation for both wafer fab and test equipment manufacturers to work toward finding a solution.

There are other limitations. This TSV probing technique requires special probe needles to be added to the probe card to supply power and clock. This will add to the total cost. Also, the additional test logic including the B1 and B2 switches shown in Fig. 1 will occupy extra die area increasing the cost. On-going work is now trying to quantify the cost of the proposed method. However, analyzing the cost of the proposed TSV testing and diagnosis method alone does not present the full picture of 3D IC testing cost. Eventually, to analyze the economic factors of 3D IC testing, we must consider the entire 3D IC manufacturing and test flow. The 3D IC cost depends on many factors like the total number of layers in a 3D IC, the layer bonding scheme (die on die or wafer-on-wafer), the specific test flow (are all layers tested, or only some layers are selectively tested), etc.

The TSV probing approach [14–16] enables testing of many TSVs in parallel, and thus raises the concerns on the thermal issues. Compared to the on chip circuitry, TSV testing occupies very small amount of logic. The power of TSV testing is expected to be small. Further research is needed to quantify the power issues in parallel TSV testing.

After identification of faulty TSVs using SOS3, TSV redundancy repairing scheme can come into play to replace the faulty TSVs with redundant good ones. Due to the clustering nature of TSV defects, replacing a faulty TSV with a neighboring TSV may not be practical, since the probability of that TSV being defective can be relatively high. On the other hand, if the spare TSV is placed far away from the faulty TSV it would cause long routing associated with the repair. Research in this direction is needed to resolve the routing congestions during TSV replacement and repair.

11 Conclusion

The three-stage optimization method named SOS3 [29] reduces pre-bond TSV test and identification time. SOS3 includes two integer linear program (ILP) models. The first ILP model (ILP-1) generates near-optimal set of test sessions for pre-bond TSV probing. Extensive experiments demonstrate that in all cases considered ILP-1 reduced the pre-bond TSV identification time compared to that of a previous heuristic method. We define two termination conditions for TSV probing and show how these conditions help speed up faulty TSV identification in a network. Finally, a session sorting procedure is proposed to sequence test sessions in such a way that the pre-bond TSV test can terminate as soon as possible for a small number of faulty TSVs within a network. This session sorting procedure, although greedy in nature, is based on iterative execution of the second ILP model (ILP-2). Extensive experimental results for various TSV networks demonstrate the benefit of session sorting on reducing test time expectation. We show that SOS3 guarantees that the test time expectation is always a small fraction of the total time of applying all sessions. As a framework, SOS3 is expected to greatly reduce pre-bond TSV test cost in real applications.

Note that the ILP-1 model is based on a sufficient but non-necessary condition for test session generation, which still leaves room for future explorations, such as, possibly finding a necessary and sufficient condition to generate a globally optimal set of sessions.

Acknowledgments This research was supported in part by the National Science Foundation Grant CCF-1116213.

References

1. Aoki M, Furuta F, Hozawa K, Hanaoka Y, Kikuchi H, Yanagisawa A, Mitsuhashi T, Takeda K (2013) Fabricating 3D integrated CMOS devices by using wafer stacking and via-last TSV technologies. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), pp 29.5.1–29.5.4
2. Chen H, Shih J, Li SW, Lin HC, Wang M, Peng C (2010) Electrical tests for three-dimensional ICs (3DICs) with TSVs. In: Proceedings of IEEE International Workshop on Testing Three-Dimensional Stacked ICs (3D-TEST), pp 1–6
3. Cho M, Liu C, Kim D, Lim S, Mukhopadhyay S (2010) Design method and test structure to characterize and repair TSV defect induced signal degradation in 3D system. In: Proceedings of International Conference on Computer-Aided Design (ICCAD), pp 694–697
4. CPLEX Optimizer. avialable from https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, accessed on July 27, 2017
5. Deutsch S, Chakrabarty K (2013) Non-invasive pre-bond TSV Test using ring oscillators and multiple voltage levels. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), pp 1065–1070
6. Hsieh A, Hwang T, Chang M, Tsai M (2010) TSV Redundancy: architecture and design issues in 3D IC. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), pp 166–171
7. Huang Y, Li J, Chen J, Kwai D, Chou Y, Wu C (2011) A built-in self-test scheme for the post-bond test of TSVs in 3D ICs. In: Proceedings of IEEE 29th VLSI Test Symposium (VTS), pp 20–25
8. Ingelsson U, Goel SK, Larsson E, Marinissen EJ (2005) Test scheduling for modular SOCs in an Abort-on-Fail environment. In: Proceedings of European Test Symposium (ETS), pp 201–206
9. Jiang L, Xu Q, Eklow B (2013) On effective through-silicon via repair for 3-D-Stacked Cs. IEEE Trans Comput Aided Des Integr Circuits Syst 32(2):559–571
10. Jung M, Mitra J, Pan D, Lim S (2011) TSV Stress-Aware Full-Chip Mechanical Reliability Analysis and Optimization for 3D IC. In: Proceedings of 48th Design Automation Conference (DAC), pp 188–193
11. Marinissen EJ, Chi CC, Verbree J, Konijnenburg M (2010) 3D DfT architecture for pre-bond and post-bond testing. In: Proceedings of IEEE International 3D Systems Integration Conference (3DIC), pp 1–8
12. Marinissen EJ, Verbree J, Konijnenburg M (2010) A structured and scalable test access architecture for TSV-based 3D stacked ICs. In: Proceedings of IEEE 28th VLSI Test Symposium (VTS), pp 269–274
13. Marinissen EJ, McLaurin T, Jiao H (2016) IEEE Std P1838: DfT Standard-Under-Development For 2.5D-, 3D-, and 5.5d-SICs. In: Proceedings of IEEE 21th European Test Symposium (ETS), pp 1–10
14. Noia B, Chakrabarty K (2011) Identification of defective TSVs in pre-bond testing of 3D ICs. In: Proceedings of 20th Asian Test Symposium (ATS), pp 187–194
15. Noia B, Chakrabarty K (2011) Pre-bond probing of TSVs in 3D stacked ICs. In: Proceedings of International Test Conference (ITC), pp 1–10
16. Noia B, Chakrabarty K (2014) Design-for-Test and test optimization techniques for TSV-based 3D stacked ICs. Springer, Berlin
17. Pan D, Lim S, Athikulwongse K, Jung M, Mitra J, Pak J, Pathak M, Yang J (2012) Design for manufacturability and reliability for TSV-Based 3D ICs. In: Proceedings of 17th Asia and South Pacific Design Automation Conference (ASP-DAC), pp 750–755
18. PTM 45nm Model. available from http://ptm.asu.edu/, accessed on July 27, 2017
19. Rajski J, Tyszer J (2013) Fault diagnosis of TSV-based interconnects in 3-D stacked designs. In: Proceedings of International Test Conference (ITC), pp 1–9
20. Reda S, Smith G, Smith L (2009) Maximizing the functional yield of wafer-to-wafer 3-D integration. IEEE Trans Very Large Scale Integr VLSI Syst 17(9):1357–1362
21. Roy SK, Chatterjee S, Giri C, Rahaman H (2013) Faulty TSVs identification and recovery in 3D stacked ICs during pre-bond testing. In: Proceedings of IEEE International 3D Systems Integration Conference (3DIC), pp 1–6
22. Smith K, Hanaway P, Jolley M, Gleason R, Strid E (2011) Evaluation of TSV and micro-bump probing for wide I/O testing. In: Proceedings of International Test Conference (ITC), pp 1–10
23. Stapper CH, Armstrong FM, Saji K (1983) Integrated circuit yield statistics. In: Proceedings of the IEEE, pp 453–470
24. Taouil M, Hamdioui S, Verbree J, Marinissen EJ (2010) On maximizing the compound yield for 3D wafer-to-wafer stacked ICs. In: Proceedings of International Test Conference (ITC), pp 1–10
25. Verbree J, Marinissen EJ, Roussel P, Velenis D (2010) On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking. In: Proceedings of 15th IEEE European Test Symposium (ETS), pp 36–41
26. Woo DH, Seong NH, Lewis DL, Lee H-HS (2010) An optimized 3D-stacked memory architecture by exploiting excessive, High-Density TSV bandwidth. In: Proceedings of IEEE 16th International Symposium on High Performance Computer Architecture (HPCA), pp 1–12
27. Yaglioglu O, Eldridge B (2012) Direct connection and testing of TSV and Microbump devices using nanopierce contactor for 3d-IC Integration. In: Proceedings of IEEE 30th VLSI Test Symposium (VTS), pp 96–101
28. You J-W, Huang S-Y, Kwai D-M, Chou Y-F, Wu C-W (2010) Performance characterization of TSV in 3D IC via sensitivity analysis. In: Proceedings 19th Asian Test Symposium (ATS), pp 389–394
29. Zhang B (2014) Pre-bond TSV test optimization and stacking yield improvement for 3D ICs. PhD thesis, Auburn University
30. Zhang B, Agrawal VD (2014) A novel wafer manipulation method for yield improvement and cost reduction of 3D wafer-on-wafer stacked ICs. J Electronic Testing: Theory Appl 30(6):57–75
31. Zhang B, Agrawal VD (2014) An optimal probing method of pre-bond TSV fault identification for 3D stacked ICs. In: Proceedings of IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), pp 1–3
32. Zhang B, Agrawal VD (2014) An optimized diagnostic procedure for pre-bond TSV defects. In: Proceedings of the 32nd IEEE International Conference on Computer Design (ICCD), pp 189–194
33. Zhang B, Agrawal VD (2015) Diagnostic tests for pre-bond TSV defects. In: Proceedings of the 28th International Conference on VLSI Design, pp 387–392
34. Zhang B, Li B, Agrawal VD (2013) Yield analysis of a novel wafer manipulation method in 3D stacking. In: Proceedings of IEEE International 3D Systems Integration Conference (3DIC), pp 1–8
35. Zhao Y, Khursheed S, Al-Hashimi BM (2011) Cost-effective TSV grouping for yield improvement of 3d-ICs. In: Proceedings of IEEE 20th Asian Test Symposium (ATS), pp 201–206

Bei Zhang obtained his PhD degree from the Electrical and Computer Engineering Department at Auburn University in 2014. During his PhD, he conducted research on electronic testing, fault diagnosis, and 3D stacked ICs testing. He received his BE and MS degrees with honors from the Electrical Engineering Department at Harbin Institute of Technology, China in 2008 and 2010, respectively. After graduation from Auburn University, he joined Apple Inc., Cupertino, California, USA, where he is currently working as a DFT (design for test) engineer.

Vishwani D. Agrawal is a Professor Emeritus in the Department of Electrical and Computer Engineering at Auburn University, Alabama, USA. Prior to retirement in 2016, he was the James J. Danaher Professor in the same department. He has over forty years of industry and university experience, working at Bell Labs, Murray Hill, NJ, USA; Rutgers University, New Brunswick, NJ, USA; TRW, Redondo Beach, CA, USA; IIT, Delhi, India; EG&G, Albuquerque, NM, USA; and ATI, Champaign, IL, USA. His areas of expertise include VLSI testing, low-power design, and microwave antennas. He obtained his BE degree from the Indian Institute of Technology Roorkee, Roorkee, India, in 1964; ME degree from the Indian Institute of Science, Bangalore, India, in 1966; and PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign, in 1971. He has published over 400 papers, has coauthored five books and holds thirteen United States patents. He is the Editor-in-Chief of the Journal of Electronic Testing: Theory and Applications, and a past Editor-in-Chief (1985-87) of the IEEE Design & Test of Computers magazine. He served on the Board of Governors (1989-90) of the IEEE Computer Society, and in 1994 chaired the Fellow Selection Committee of that Society. He has received nine Best Paper Awards, two Lifetime Achievement Awards, and the Distinguished Alumnus Award of the University of Illinois. Agrawal is a Fellow of the ACM, IEEE and IETE India. He has served on the Advisory Boards of the ECE Departments of the University of Illinois at Urbana-Champaign, New Jersey Institute of Technology, and the City College of the City University of New York. See his website: http://www.eng.auburn.edu/~vagrawal.