An Optimized Diagnostic Procedure for Pre-Bond TSV Defects

Bei Zhang and *Vishwani D. Agrawal*
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA
bzz0004@auburn.edu, vagrawal@eng.auburn.edu

***Abstract*— Based on a recent pre-bond TSV probing technique, this paper proposes an iterative greedy procedure to sort the order of test sessions for reducing pre-bond TSV test time. We then combine that session sorting procedure with two existing methods to develop a 3-Step test time Optimization Simulator named "SOS3". SOS3 consists of an ILP (integer linear programming) model for session generation, the greedy procedure for session sorting, and a TSV identification algorithm for early test termination. Experiments are done for various TSV networks and two observations are made. First, session sorting plays an important role within SOS3 as it helps to further reduce pre-bond test time expectation and thus reduces pre-bond TSV test cost. Second, SOS3 as a framework greatly speeds up the pre-bond TSV test.**

I. Introduction

Pre-bond TSV testing identifies defective dies early in the process, which provide the known good die (KGD) information for die-to-die, die-to-wafer or even wafer-on-wafer fabrication process [14], [17]. Several built-in self-test (BIST) techniques for pre-bond TSV testing have been proposed [3], [4], [8], [13]. However, the 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 TSVs on a chip [12]. Moreover, the BIST circuits themselves suffer from process variation, which may render them totally useless.

A pre-bond TSV probing method has been proposed [7] where multiple TSVs are shorted together by one probe needle, forming a TSV network. A parametric test is conducted by adding an active driver in the probe needle and forming a charge sharing circuit between one (or multiple) TSV(s) and the probe needle. This probing method offers robustness to process variation, requires less hardware overhead, and is likely to be utilized in practice. Based on that work [7], those authors further proposed [6] a heuristic method to generate a series of test sessions that can uniquely identify up to a certain number of faulty TSVs within a network. This heuristic method reduced test time compared to sequential testing of each TSV. Recent work [16] further proposes an ILP (integer linear programming) model for session generation and demonstrates that the total number of sessions obtained by the ILP model is less than that generated by [6]. It has been observed [15] that though sometimes all sessions need to be tested to identify faulty TSVs, most of the time the TSV identification process can be terminated by testing only a fraction of all sessions.

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

In real silicon, TSV yield is expected to be more than 99% [2]. We calculated the probability of different number of failing TSVs within a TSV network considering different TSV defect distributions. The results suggest that the probability of ϕ faults within a TSV network decreases dramatically as ϕ increases. This observation motivates us to emphasize the application order of test sessions so that pre-bond TSV test can be terminated as soon as possible. We make two contributions in this paper. First, we propose an iterative greedy procedure for session sorting. Second, we combine the iterative greedy procedure with two existing methods and form a 3-Step test time Optimization Simulator ("SOS3"). SOS3 consists of three steps, namely, ILP-based session generation [16], iterative greedy procedure for session sorting, and fast TSV identification algorithm for early test termination [15]. Each step provides inputs to the next step. In the experimental section, we calculate the test time expectation for various TSV networks. The results demonstrate that session sorting procedure plays an important role in SOS3 as it helps further reduce test time expectation by as large as 31.8%. We also observe that with SOS3 the expectation of TSV identification time is much less than the total time of testing all sessions. Thus, fast pre-bond TSV testing offered by SOS3 is expected to greatly reduce pre-bond test cost of 3D stacked ICs.

II. Preliminaries

A. TSV probing technique

To pinpoint TSV resistive defects before die bonding, recent work[7] proposes the use of a large probe needle with active driver to contact many TSVs at a time. The TSVs simultaneously contacted by the probe needle form a TSV network. The number of TSVs within a network is typically less than 20 based on the relative diameter and pitch of probe needles and TSVs [9], [10]. TSV resistance test can be conducted by adding an active driver in the probe needle and forming a charge sharing circuit between one (or multiple) TSV(s) and the probe needle. The goal of TSV probing is to identify up to a certain number of faulty TSVs (let's say m) within a TSV network so that the on-chip TSV redundant architecture can replace all identified faulty

TSVs with redundant ones. If the number of faulty TSVs within network exceeds m, then not all faulty TSVs can be replaced due to the limited redundancy and the chip should be discarded.

B. Test session generation

Any TSV can be tested individually using the previous technique [7]. However, testing each TSV sequentially results in unnecessarily long test time. To reduce test time, multiple TSVs within the same network may be tested in parallel [6]. With carefully designed parallel tests up to m faulty TSVs can be uniquely identified. The TSVs tested in parallel form a test session. The number of TSVs within a session can range anywhere from 1 to r where r represents the maximum number of TSVs a session can hold. Based on the previous technique [6], [7], we draw three conclusions. First, the test time of a session is inversely proportional to the number of TSVs within the session. Second, any faulty TSV within a session will cause the session test to fail but we cannot decide which TSV is faulty. Third, a good session test implies that all TSVs within the session are fault-free.

By forming a series of sessions that can uniquely identify up to m faulty TSVs within a TSV network [6], [16], it is possible to reduce the test time compared to that of sequentially testing each TSV. Reference [6] leveraged a heuristic method to generate such set of sessions, though with results that are far from being optimal. Recent work [16] has proposed an ILP (integer linear programming) model for test session generation. The ILP model reduces the total number of sessions and total test time compared to the heuristic [6]. That ILP model [16] is referred to as "ILP model 1" to differentiate it from the ILP model proposed in the present work.

C. A fast TSV identification algorithm

We define a fault map ρ as positions of all defective TSVs within a TSV network. $|\rho|$ indicates the number of faulty TSVs within ρ. The test sessions [6], [**?**] guarantee that all fault maps with $|\rho| \leq m$ can be uniquely identified. However, a careful observation [15] shows that for identifying most of the fault maps with $|\rho| \leq m$, only a small portion of the total sessions need to be tested. The identification test can stop as soon as one of the following conditions occurs: 1) all TSVs have been identified as either good or bad, or 2) the number of identified faulty TSV reaches $m+1$ because then the chip will fail due to lack of redundant resources and further identification is useless. Based on these termination conditions, a fast TSV identification algorithm has been developed [15]. With that TSV identification algorithm the average test time is much less than the total test time for various TSV networks.

D. Terminology

We define the following terminology:

1) *TSV network.* It consists of all TSVs that are simultaneously contacted by a single probe needle.
2) *Test session.* During pre-bond TSV probing, TSVs tested in parallel within the same TSV network form a test session.
3) *Session size.* Session size is the number of TSVs within a session.
4) *Test time of a session.* Test time of a session is inversely proportional to the session size [6], [7], [16].
5) *Resolution constraint.* Resolution constraint r indicates that the session size should never exceed r [6], [16].
6) *Maximum number of faulty TSVs to be identified within a network.* This number m equals to the number of redundant TSVs in the TSV network being tested.
7) *Fault map.* Fault map ρ represents positions of all defective TSVs within a network.

III. PROBABILISTIC ANALYSIS OF NUMBER OF FAULTY TSVS WITHIN A TSV NETWORK

TSV defect distributions can be broadly classified as two types, namely independent defect distribution [18] and clustered defect distribution [11], [18]. For independent TSV defect distribution, the failing probabilities of TSVs are independent of each other. The probability of ϕ faulty TSVs within a T-TSV network can be calculated as:

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

where p is average TSV failing probability.

To model defect clustering we consider a scenario where the presence of a defective TSV increases the probability of more defects in a close neighborhood [11], [18]. Such a phenomenon can be modeled [18] by assuming that, 1) a defect cluster center [11], [18] only consists of one single defective TSV, and 2) the failing rate of TSV_i is inversely proportional to the distance from the existing cluster center. Thus,

$$p(TSV_i) = p \cdot (1 + (\frac{1}{d_{ic}})^{\alpha}) \tag{2}$$

where $p(TSV_i)$ is the failing probability of TSV_i, d_{ic} is the distance between TSV_i and the cluster center, and α is a clustering coefficient. A larger value of α implies less clustering. As $\alpha \rightarrow \infty$, the defect distribution becomes independent defect distribution.

The clustered model need to take the TSV location information into account. Since the number of TSVs within a network is typically less than 20, we consider each TSV network as a 5×5 matrix. The value 5 is chosen based on the ratio of the pitch of current probe needle and the pitch of realistic TSVs [9], [10]. We randomly put T TSVs on the integral coordinates of the matrix to obtain the location information for each TSV. After that, we employ equation 2 to analyze the probability for the number of defective TSVs (ϕ) within a network. Following [18], we assume that each

TABLE I: Probability of the number ϕ of failing TSVs within a 15-TSV network.

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

TSV network has only one defect cluster and defect clusters within different networks do not interfere with each other.

Table I shows the probability of different values of ϕ for a 15-TSV network. We vary the TSV yield from 98.0% to 99.5% to accommodate different levels of maturity of the manufacturing processes. The clustering coefficient α is set as 1 and 2, similar as previously reported [5], [18]. The values under clustered defect distribution are averaged over 100 Monte Carlo runs, with each run randomly placing 15 TSVs on 5×5 matrix. By doing this, we try to simulate all possible TSV placements in real silicon. Three important observations are made from Table I. First, no matter what defect distribution is, the probability of $\phi = 0$ is the largest and even much larger than the sum of the rest situations 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 quite low. Third, as TSV yield decreases, the probability of $\phi = 0$ decreases. Motivated by these 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.

IV. An iterative greedy procedure for test session scheduling

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

$$E(\Gamma) = \sum_{\text{Any}\rho} \gamma(\rho) P(\rho) \tag{3}$$

where $\gamma(\rho)$ is the identification time to determine the fault map ρ using the TSV identification algorithm [15], 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. We formulate two problems to be solved.

Problem 1. 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 1, a straightforward method is to find all permutations of test sessions, and for each permutation calculate the test time expectation using equation 3. The permutation that yields minimum $E(\Gamma)$ would be the best choice. However, there can be $N!$ permutations and $2^T - 1$ fault maps. So the identification algorithm [15] must be run by $N! \cdot (2^T - 1)$ times, which is highly time-consuming even for a small network. Fortunately, we notice that the probability for the number ϕ of faulty TSVs is inversely proportional to ϕ. Specifically,

$$\sum_{|\rho|=i} P(\rho) \gg \sum_{|\rho|=j} P(\rho) \text{ for any } i < j \tag{4}$$

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 $P(\rho = \emptyset)$ (all TSVs being good in a network) 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, a second problem is formulated.

Problem 2. 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 2 can be solved by constructing an ILP model, referred to here as "ILP model 2". We introduce a variable P_j, $j \in [1, N]$, such that

$$P_j = \begin{cases} 1 & \text{if session } S_j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Then, the ILP model is described as follows:

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

Subject to constraints: TSV_i is tested at least once by the selected sessions, for $1 \geq i \geq T$.

Let L_j represent 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 [6]. The number of variables and constraints for ILP model 2 (a measure of the complexity of the problem) are $O(NT)$ and $O(T)$, respectively. In all our experiments, ILP model 2 is solved in 1 second or less. The M sessions covering all TSVs with minimal time will be sorted and tested before the rest of the sessions. This will reduce $\gamma(\rho = \emptyset)$ and thus reduce the test time expectation.

As can be seen from Table I, $Prob(\phi = 1)$ is significant. If we can further reduce $\gamma(\rho)$ with $|\rho| = 1$, the test time expectation should be reduced too. The N test sessions in Problems 1 and 2 can be produced by either the ILP model 1 [16] or the heuristic method [6]. Because 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 hold the M sessions produced by

```
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. 1: Pseudo-code for iterative test session sorting.

ILP model 2, and find M_1 sessions from the rest $N - M$ session such that these $M + M_1$ sessions will cover each TSV at lease twice and the test time of the M_1 sessions is the minimum. Next we explain how ILP model 2 can be again utilized to find these M_1 sessions. We first count the times each TSV is covered by the first M sessions, and put the TSVs which have been covered only once to a list named "TSV_set". ILP model 2 can be 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 the minimum. The produced M_1 sessions will be sorted and tested directly after M sessions. These $M + M_1$ sessions guarantee that $\gamma(\rho = \emptyset)$ is minimized and based on this premise we further minimize $\gamma(\rho)$ with $|\rho| = 1$ (simply represented as $\gamma(|\rho| = 1)$). Similar procedure can be repeated for reduction of $\gamma(|\rho| = 2)$, $\gamma(|\rho| = 3)$, $\cdots$, until $\gamma(|\rho| = m)$.

We summarize the overall procedure for session sorting in Figure 1. ILP model 2 is iteratively utilized in *Test_session_sorting* procedure with each execution tries to find a subset of sessions from "$Original\_sessions$" so as to cover all the 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 model 2 times how many times ILP model 2 is executed which is determined by "$Stop\_index$" in Figure 1. Note "$Stop\_index$" can be set as 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 which covered all TSVs at least once with minimum time. When "$Stop\_index$" is $m + 1$, the procedure will first reduce $\gamma(\rho = \emptyset)$, and then reduce $\gamma(|\rho| = 1)$, and then reduce $\gamma(|\rho| = 2)$, $\cdots$, all the way up to reducing $\gamma(|\rho| = m)$.

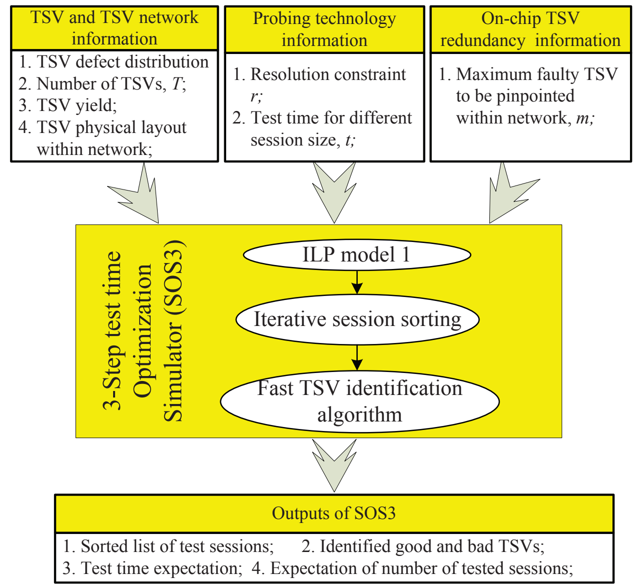

Fig. 2: Three-step test time optimization simulator.

V. A THREE-STEP TEST TIME OPTIMIZATION SIMULATOR

In this section we proposed a 3-Step test time Optimization Simulator (SOS3). The first step of SOS3 is ILP model 1 [16] for test session generation. Note that we choose ILP model 1 instead of the heuristic method [6] because test sessions generated by both methods have exactly the same TSV identification capability, however, the ILP model generates fewer sessions and is more time-efficient. The second step of SOS3 is the proposed iterative greedy procedure for session sorting. This procedure accepts the N sessions from step 1 as the inputs and sort the sessions to reduce test time expectation. The last step is the fast TSV identification algorithm [15]. This algorithm takes the sorted list of sessions as the inputs and finish the identification process as soon as any termination condition happens. 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 2 illustrates the overall diagram of SOS3. The inputs to SOS3 contain 3 pieces of information: 1) the TSV and TSV network information, 2) the probing technology information, and 3) the 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 tested sessions.

VI. EXPERIMENTAL RESULTS

We compare both the expectation of test time and number of tested sessions between two different simulators: SOS3 and SOS2 (2-Step test time Optimization Simulator). The only difference between SOS2 and SOS3 is that the iterative session sorting procedure is eliminated in the former. By comparing these two simulators, we try to illustrate the importance of session sorting for test time reduction. Note we did not compare the test time expectation of SOS3 to that of the heuristic method [6] due to the following two reasons. First, ILP model 1 returns a smaller number of sessions and less test time; Second, the session sorting procedure in combination with the identification algorithm helps reduce test time expectation further.

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

$$E(\Gamma) = \begin{cases} \sum\limits_{|\rho|<2} \gamma(\rho)P(\rho) + TT \sum\limits_{|\rho|\geq 2} P(\rho) & \text{if } m = 1 \\ \sum\limits_{|\rho|\leq 2} \gamma(\rho)P(\rho) + TT \sum\limits_{|\rho|\geq 3} P(\rho) & \text{if } m \geq 2 \end{cases} \quad (6)$$

where TT represents the total time of testing all the sessions produced by ILP model 1 [16].

Similarly the expectation of number of tested sessions $E(S)$ in this section is estimated as follows:

$$E(S) = \begin{cases} \sum\limits_{|\rho|<2} \eta(\rho)P(\rho) + N \sum\limits_{|\rho|\geq 2} P(\rho) & \text{if } m = 1 \\ \sum\limits_{|\rho|\leq 2} \eta(\rho)P(\rho) + N \sum\limits_{|\rho|\geq 3} P(\rho) & \text{if } m \geq 2 \end{cases} \quad (7)$$

where $\eta(\rho)$ represents the number of tested sessions for identification of fault map ρ using TSV identification algorithm [15]. N represents the total number of sessions produced by ILP model 1 [16].

Equations 6 and 7 are explained as follows. 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 $m = 1$ is not intended for identifying more than one faulty TSV. Moreover, $P(\phi \geq 2) = \sum\limits_{|\rho|\geq 2} P(\rho)$ is low. Such a low probability has negligible impact on expectation calculation. For TSV networks with $m \geq 2$, we assume the entire sessions need to be tested 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, $P(\phi \geq 3) = \sum\limits_{|\rho|\geq 3} P(\rho)$ is pretty low as referring to Table I. Such a low probability has little impact on expectation calculation.

Based on equations 6 and 7, we compare SOS2 and SOS3 for various values of T, m, r. A commercial ILP solver named CPLEX [1] is used in our experiments. For all simulations, SOS3 and SOS2 provide the outputs in less than one minute. The expectation of number of tested sessions and test time for both SOS2 and SOS3 are shown in Tables II and III, respectively. We provide an insightful evaluation of SOS3 by varying TSV yield from a low value of 98.0% to a practically expected value of 99.5%. Due to space limitation, we did not provide simulation results of 99.0% TSV yield.

TABLE II: Expectation of number of tested sessions, defect clustering coefficient $\alpha = 1$, data shows (sessions for SOS2, sessions for SOS3, reduction by SOS3).

Parameters T, m, r	Total number of sessions, N	Expected number of tested sessions, $E(S)$	
		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 III: Expectation of test time (10^{-7}s), defect clustering coefficient $\alpha = 1$, data shows (test time for SOS2, test time for SOS3, reduction by SOS3).

Parameters T, m, r	Test time of all sessions, TT	Expectation of test time, $E(\Gamma)$	
		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%)

Nonetheless, those simulation results always lie somewhere between the results 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 quite similar for the rest two defect distributions in Table I.

The first column of both tables show various TSV networks with different T, m, r. The second column of Table II and Table III show the total number of sessions and total test time of all sessions produced by ILP model 1 [16], respectively. Column 3 and 4 of both tables consist of elements with each element consisting of 3 values, i.e., (the expectation value of SOS2, the expectation value of SOS3, the relative percentage improvement of SOS3 over SOS2). Note that all values are the averaged results of 100 Monte Carlo runs, with each run corresponding to a different TSV placement in a 5-by-5 matrix. By doing this we try to simulate all possible TSV layouts in real silicon.

We make four observations from Table II and III. First, both the expectation of number of tested sessions and the expectation of test time is only a small portion 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. As can be seen from Table II, the total number of sessions can reach as high as 25, but the expectation of number of tested sessions is always no more than 8.7 for SOS3. For TSV network with $T = 20$, $m = 4$, $r = 4$, the test time expectation of SOS3 is only 31.3% of the total test time, considering 98.0% TSV yield. Second, both the expectation of test time and number of tested sessions increases for lower TSV yield. This is because the probability of having larger number of defective TSVs within a network increases as TSV yield decreases. Identifying larger number of faulty TSVs typically takes more sessions and longer test time for both SOS2 and SOS3. Third, SOS3 helps further reduce the expectation value compared to SOS2. For example, for $T = 11$, $m = 2$, $r = 3$, SOS3 further reduces test time expectation by 31.6% compared to SOS2 considering 99.5% TSV yield. Fourth, the improvement of SOS3 over SOS2 decreases as TSV yield decreases. This is because the session sorting procedure within SOS3 puts higher priority on reducing the identification time for smaller value of $|\rho|$. To identify fault map with large $|\rho|$ (like $|\rho| \geq 2$), 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. The large differences between SOS2 and SOS3 illustrate the significance of session sorting in reducing test time expectation.

VII. Conclusion

In this work, 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 small number of faulty TSVs within a network. This session sorting procedure, although greedy in nature, is based on iterative execution of an ILP model. We also propose a 3-step test time optimization simulator (SOS3) to reduce pre-bond test time expectation. In SOS3, an existing ILP model is first used to generate a series of test sessions with certain identification capability. Then, session sorting is conducted based on the generated sessions. Lastly, an existing fast TSV identification algorithm is used for early test termination. Extensive experimental results for various TSV networks demonstrate the benefit of session sorting on reducing test time expectation. We also show how SOS3 guarantees that the test time expectation is always a small portion of the total time of testing all the sessions. As a framework, SOS3 is expected to greatly reduce pre-bond TSV test cost in real silicon.

References

[1] "CPLEX Optimizer, avialable from." http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, accessed on May 18, 2014.

[2] M. Aoki, F. Furuta, K. Hozawa, Y. Hanaoka, H. Kikuchi, A. Yanagisawa, T. Mitsuhashi, and K. Takeda, "Fabricating 3D Integrated CMOS Devices by using Wafer Stacking and Via-Last TSV Technologies," in *Proc. IEEE International Electron Devices Meeting (IEDM)*, 2013, pp. 29.5.1–29.5.4.

[3] P. Chen, C. Wu, and D. Kwai, "On-Chip Testing of Blind and Open-Sleeve TSVs for 3D IC Before Bonding," in *Proc. IEEE 28th VLSI Test Symposium*, 2010, pp. 263–268.

[4] M. Cho, C. Liu, D. H. Kim, S. K. Lim, and S. Mukhopadhyay, "Design Method and Test Structure to Characterize and Repair TSV Defect Induced Signal Degradation in 3D System," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 694–697.

[5] L. Jiang, Q. Xu, and B. Eklow, "On Effective Through-Silicon Via Repair for 3-D-Stacked ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 2, pp. 559–571, 2013.

[6] B. Noia and K. Chakrabarty, "Identification of Defective TSVs in Pre-Bond Testing of 3D ICs," in *Proc. IEEE 20th Asian Test Symposium (ATS)*, 2011, pp. 187–194.

[7] B. Noia and K. Chakrabarty, "Pre-Bond Probing of TSVs in 3D Stacked ICs," in *Proc. International Test Conference (ITC)*, 2011, pp. 1–10.

[8] B. Noia and K. Chakrabarty, *Design-for-Test and Test Optimization Techniques for TSV-based 3D Stacked ICs*. Springer, 2014.

[9] D. Z. Pan, S. K. Lim, K. Athikulwongse, M. Jung, J. Mitra, J. S. Pak, M. Pathak, and J. Yang, "Design for Manufacturability and Reliability for TSV-Based 3D ICs," in *Proc. 17th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2012, pp. 750–755.

[10] K. Smith, P. Hanaway, M. Jolley, R. Gleason, and E. Strid, "Evaluation of TSV and Micro-Bump Probing for Wide I/O Testing," in *Proc. International Test Conference (ITC)*, 2011, pp. 1–10.

[11] C. H. Stapper, F. M. Armstrong, and K. Saji, "Integrated Circuit Yield Statistics," in *Proceedings of the IEEE*, 1983, pp. 453–470.

[12] D. H. Woo, N. H. Seong, D. L. Lewis, and H.-H. S. Lee, "An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth," in *Proc. IEEE 16th International Symposium on High Performance Computer Architecture (HPCA)*, 2010, pp. 1–12.

[13] J. You, H. S., D. Kwai, Y. Chou, and C. Wu, "Performance Characterization of TSV in 3D IC via Sensitivity Analysis," in *Proc. 19th Asian Test Symposium (ATS)*, 2010, pp. 389–394.

[14] B. Zhang and V. D. Agrawal, "A Novel Wafer Manipulation Method for Yield Improvement and Cost Reduction of 3D Wafer-on-Wafer Stacked ICs," *Journal of Electronic Testing: Theory and Applications*, vol. 30, pp. 57–75, 2014.

[15] B. Zhang and V. D. Agrawal, "An Optimal Probing Method of Pre-Bond TSV Fault Identification for 3D Stacked ICs," in *Proc. IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, 2014. Personal communication.

[16] B. Zhang and V. D. Agrawal, "Diagnostic Tests for Pre-Bond TSV Defects," in *28th International Conf. VLSI Design*, Jan. 2015. Submitted.

[17] B. Zhang, B. Li, and V. D. Agrawal, "Yield Analysis of a Novel Wafer Manipulation Method in 3D Stacking," in *Proc. IEEE International 3D Systems Integration Conference (3DIC)*, 2013, pp. 1–8.

[18] Y. Zhao, S. Khursheed, and B. M. Al-Hashimi, "Cost-Effective TSV Grouping for Yield Improvement of 3D-ICs," in *Proc. 20th Asian Test Symposium (ATS)*, 2011, pp. 201–206.