Test-TSV Estimation During 3D-IC Partitioning

Shreepad Panth[1], Kambiz Samadi[2], and Sung Kyu Lim[1]
[1]Dept. of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332
[2]Qualcomm Research, San Diego, CA 92121

***Abstract*—Three dimensional integrated circuits (3D-ICs) are emerging as a viable solution to the interconnect scaling problem. During early design space exploration, a large number of possible partitioning solutions are evaluated w.r.t. performance, area, through-silicon-via (TSV) count, etc. During this evaluation process, the number of test-TSVs need to be added to the total TSV count, to prevent unexpected area overhead later on in the design flow. While a fixed test-TSV count may provide sufficient guardbanding, in this paper we show that it often overestimates the actual number of test-TSVs required. Currently, the only way to determine the pareto-optimial test-TSV count is to sweep the test-TSV constraint, and repeatedly apply 3D test architecture optimization algorithms. This process is time consuming, and is too slow to be used in automated partitioning. In this paper, we present a quick and accurate estimation of the pareto-optimal number of test-TSVs required for a given partition. This can be used as an input to the partitioner to quickly estimate the total number of TSVs used for a given partition, reducing over-design.**

I. Introduction

Today's integrated circuits are interconnect limited, as interconnects get slower at smaller technology nodes. Three dimensional integrated circuits (3D-ICs) are emerging as a viable solution to this problem. Devices are placed in three dimensions, and the vertical interconnections are achieved using through-silicon vias (TSVs). This reduces the longest and average interconnect length, and it has been shown [1] that TSV-based 3D-ICs can achieve lower wirelength and longest path delay when compared with their 2D counterparts.

TSV-based 3D-ICs are manufactured by fabricating each die separately, and then stacking them one on top of the other. A 3D-IC can be tested either before the dies are stacked (pre-bond test), or after stacking (post-bond test) [2]. Pre-bond test access is provided by adding large probe-pads for probe needle touchdown [3], or using probe cards that can probe TSV microbumps directly [4]. Post-bond test access is provided by the package pins for the entire chip, and test-TSVs for dies not directly connected to the package substrate [5].

During early design space exploration, a large number of possible partitioning solutions are evaluated w.r.t. power, performance, area, TSV count, etc. The TSV count includes the number of signal TSVs, as well as estimates of TSVs for power delivery, clock, thermal, and test. The number of test-TSVs depend on the test architecture, and includes TSVs required for control, as well as those required to pump data. If test-TSVs are not accounted for during partition evaluation, downstream design steps may have insufficient area to add these TSVs. One such example is shown in Figure 1, where floorplanning was carried out considering only signal TSV count. Insufficient area remains to add other TSVs such as clock, power and test. The only solution is to expand die area, which increases cost, and reduces yield.

The number of test-TSVs required can be budgeted as a large fixed number, but this introduces the possibility of a very high guardband, which could degrade solution quality. A better approach is to accurately determine the exact number of test-TSVs required for a given partition. Existing work only focusses on determining the test time given a fixed test-pin and test-TSV constraint, so we would need to sweep the test-TSV constraint, and repeatedly apply these algorithms to find the pareto-optimial test-TSV count. While this process works if the partition is fixed, it is too slow to be used during early design space exploration. In this paper, we derive a fast and accurate estimate of the pareto-optimal number of test-TSVs required for a given 3D partition. This estimate can be fed into automated 3D-IC partitioning tools, to most accurately estimate the total TSV count of a given partition.

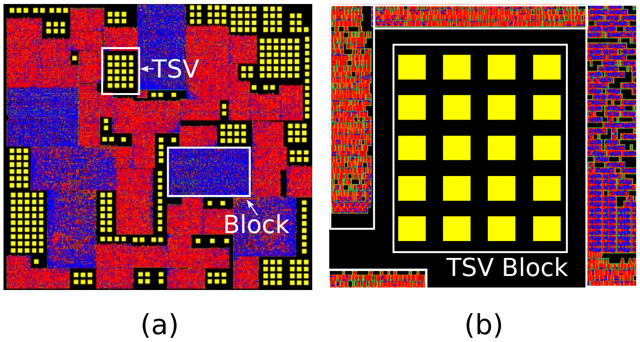

Fig. 1. (a) GDSII screen shot of a single die of a block-level 3D-IC (b) Zoom in shot of the boxed TSV block in (a)

II. Preliminaries

A. Prior Work

Challenges facing 3D test were enumerated in [6], and the first pre-bond test architecture was presented in [3]. This architecture is similar to IEEE 1500, and a pre-bond testable architecture based on extensions to IEEE 1500 was formalized in [2], [5].

Algorithms to construct 3D scan chains were presented in [7], but this architecture is not pre-bond testable. Pre-bond testable clock trees were presented in [8]. There has also been been prior work on designing wrappers for 3D-ICs, assuming different test access mechanism (TAM) width for pre-bond and post-bond test [9].

Test architecture optimization for 3D-ICs was presented in [10]. The authors formulated an ILP problem that performs test scheduling for a 3D-IC given a fixed test-pin and test-TSV constraint. While this algorithm can be repeatedly applied to determine the pareto-optimal test-TSV count for a given partition, it is too slow to be used to evaluate millions of possible solutions. To the best of our knowledge, there is no prior work on quickly estimating the number of pareto-optimal test-TSVs.

B. Motivation

As mentioned in Section I, the total TSV count of a given partition needs to include accurate estimates of the test-TSV count. The chosen test architecture determines the number of control test-TSVs, while

This work is supported by SRC under the Integrated Circuit & Systems Sciences program (Task ID: 1836.075).

978-1-4673-6484-3/13/$31.00 ©2013 IEEE

the number of TSVs required to pump data are variable, and left up to the design engineer. Only the latter is of interest in this paper, as the former remains constant irrespective of partition. In the remainder of this paper, test-TSVs refer only to those TSVs used to carry test vectors and responses, and control test-TSVs can be treated as a separate, fixed constant.

If a fixed number of test-TSVs ($TSV_{t,f}$) are allocated during partitioning, there is the possibility of overestimating the real total TSV count of a partition. It has been shown [11] that pareto-optimality exists in the test-TSV count. If $TSV_{t,po}$ is the pareto-optimal number of test-TSVs, any TSVs allocated beyond this will not yield a reduction in test time. The actual number of test-TSVs used during scheduling is given by

$$TSV_t = \min(TSV_{t,f}, TSV_{t,po}) \qquad (1)$$

In area critical designs, when $TSV_{t,f}$ is small, it is usually the smaller of the two, so it serves as a reasonable estimate. However, if $TSV_{t,f}$ is large, and it was used as an estimate for TSV_t, several candidate solutions would be discarded for having too many TSVs. Therefore, an accurate estimate of $TSV_{t,po}$ is required, and it needs to be quickly computed to be incorporated into automatic partitioning.

We focus on block-level 3D-ICs in this work, as they will be the first 3D-ICs to appear [12]. Only post-bond test is considered, as the pre-bond test time is influenced by factors other than test-TSV count, such as probe pad count etc. The ILP-based test scheduling algorithm presented in [10] is used to compute test time. Since the test time estimate is meant to be used during design space exploration, top-level interconnect tests are ignored, and all blocks are assumed to be soft i.e., the number of scan chains are yet to be decided.

III. Die-level partitioning

We first study die-level partitioning, where different partitions have different orders in which the dies are stacked. While the solution space is small, and exhaustive search methods can easily be applied, we use insights gained in this section to explain block-level partitioning in Section IV.

A. Two-die stack

A two tier die-level stack is the simplest form of 3D-IC, and there are only two partitions possible. Furthermore, only two test scheduling options exist, serial or parallel test. In serial test, each die is tested one at a time, the bottom die with all the test-pins, and the top die with all the test-TSVs. In parallel test, the test-pins are divided between the bottom and the top die. We consider the three circuits shown in Figure 2. The first circuit is a homogeneous stack, and the next two are different die-level partitions of a heterogeneous stack. Each die is a circuit taken from the ITC'02 SOC benchmarks [13]. Since the solution space is small, we try all possible test scheduling options, and tabulate the pareto-optimal TSV count for both serial and parallel test in Table I. We assume 50 test-pins, and sweep the test-TSV count to obtain the minimum test time and $TSV_{t,po}$. The parallel schedule offers lower test time, and would be chosen by any test scheduling algorithm. For the homogeneous stack, an equal division of test-pins is optimal, which implies that $TSV_{t,po}$ is half of the number of test-pins, or 25. For the heterogeneous stack however, we observe that both partitioning options give the same minimum test time, but $TSV_{t,po}$ is different. As expected, the partition with the more complex die on top requires more test-TSVs to obtain minimum test time.

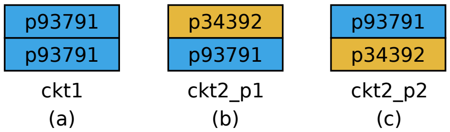

Fig. 2. Three different circuits considered for die-level partitioning of a two-die stack. (a) A homogeneous stack, (b & c) Two different partitions of a heterogeneous stack. A larger number implies the die is more complex.

TABLE I
The optimal test times (in cycles) achieved for a two-die circuit, along with the TSV usage at which this optimum time is reached.

Circuit	Serial Test		Parallel Test	
	T_{min}	$TSV_{t,po}$	T_{min}	$TSV_{t,po}$
ckt1	2,447,767	47	2,363,730	25
ckt2_p1	1,931,750	47	**1,899,170**	**19**
ckt2_p2	1,940,656	47	**1,899,170**	**31**

B. Multi-die stack

The approach taken in this section is to tabulate the test time for a given set of partitions under fixed test-pin and TSV constraints, and then use this information to identify what characteristics of the partition affect the test time. We consider the three and four die stacks shown in Figure 3. TSV constraints can be assigned in two ways. The first method is *uniform TSV constraints*, which allocates an equal TSV budget to all the dies. The second method is *tapering TSV constraints*, which allocates more TSVs for the lower dies (= closer to the package), and less TSVs for the upper dies. The test time is computed using ILP-based scheduling. We study the test time difference for both types of constraints, and tabulate them for three and four dies in Tables II and III, respectively.

It is clear from these tables that, as expected, the test time of a partition with the most complex dies closest to the package is least. However, if we have uniform TSV constraints, the test time changes only when the bottom die changes. Any permutation of the upper dies without changing the bottom die does not affect the test time. Furthermore, if the pin and TSV constraints are equal, partitioning has no impact on the test time. If two partitions have the same test time when tested with the same number of TSVs, it follows that they both also have the same $TSV_{t,po}$. What this implies is that, during the partitioning process, we only need to update $TSV_{t,po}$ when the complexity of the bottom die changes. Its value is computed in the next section, using lower bounds.

These results are not restricted to our simulation settings, and it is possible to formally prove them. A formal proof is provided in Appendix Section A.

IV. Block-level partitioning

Block-level partitioning is the more general case of die-level partitioning. We study how the test time changes for different partitions under fixed test-TSV constraints, derive lower bounds on the test time, and use this lower bound to derive equations for $TSV_{t,po}$. As in the previous section, we start with the two die case, and extend the results to multiple dies. In this section, we assume uniform test-TSV constraints.

A. Two-die stack

We start with ckt2_p2 and start moving modules across the tiers. Each move results in a new partition. Two types of module moves are performed. The first is moving a module from one die to another,

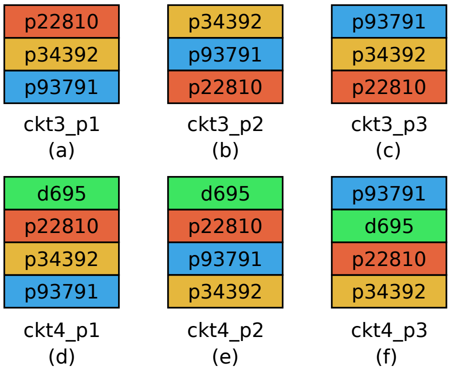

Fig. 3. Circuits considered for die-level partitioning of multi-die stacks. (a - c) three die stack, (d - f) four die stack. A larger number implies the die is more complex.

TABLE II
THE TEST TIMES FOR DIE-LEVEL PARTITIONING OF A THREE-DIE 3D-IC, CONSIDERING BOTH UNIFORM AND TAPERED TSV CONSTRAINTS.

P_{max}	TSV_{max}		Test time (cycles)		
	D2-D1	D3-D2	ckt3_p1	ckt3_p2	ckt3_p3
50	50	50	**2,197,060**	**2,197,060**	**2,197,060**
	30	30	2,252,535	**3,138,753**	**3,138,753**
	30	10	2,252,535	3,826,504	7,021,398
70	70	70	**1,541,308**	**1,541,308**	**1,541,308**
	30	30	1,753,753	**3,138,753**	**3,138,753**
	30	10	2,249,017	3,826,504	7,021,398

TABLE III
THE TEST TIMES FOR DIE-LEVEL PARTITIONING OF A FOUR-DIE 3D-IC, CONSIDERING BOTH UNIFORM AND TAPERED TSV CONSTRAINTS.

P_{max}	TSV_{max}			Test time (cycles)		
	D2-D1	D3-D2	D4-D3	ckt4_p1	ckt4_p2	ckt4_p3
50	50	50	50	**2,225,765**	**2,225,765**	**2,225,765**
	30	30	30	2,300,851	**2,597,776**	**2,597,776**
	30	20	10	2,418,438	2,971,786	7,021,398
70	70	70	70	**1,561,751**	**1,561,751**	**1,561,751**
	30	30	30	1,802,068	**2,597,776**	**2,597,776**
	30	20	10	1,919,655	2,971,786	7,021,398

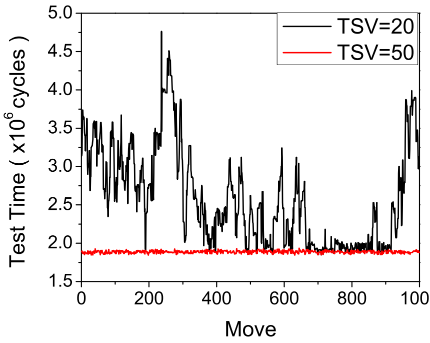

Fig. 4. The variation in test time observed for a two-die stack starting with ckt2_p2 and performing 1000 different random moves. We assume 50 test-pins and 2 different test-TSV constraints.

and the other is swapping two modules from different dies. A total of 1000 such moves are performed, and for each partition, ILP-based test scheduling is performed with 50 test-pins and two different TSV constraints. The results are plotted in Figure 4. As observed in the previous sections, if the test-TSV constraint is high enough, all partitions have similar test time. With lower test-TSV constraints (= 20), we observe that a significant number of partitions have much higher test time, indicating that their $TSV_{t,po}$ is higher. There are also partitions however (Moves 650-800), that have close to the minimum test time, indicating that their $TSV_{t,po}$ is close to 20. We next derive lower bounds, and identify what attributes of the partition determine $TSV_{t,po}$.

1) Lower bound on test time: For a module m, let i_m, o_m, and b_m be the number of input, output, and bi-directional ports, respectively. Further, let p_m be the number of patterns required to test that module. Let f_m be the number of flip flops in that module. In the case of hard modules, f_m is simply the sum of the lengths of the internal scan chains. The number of stimulus (ts_m) bits is the sum of i_m, b_m, and f_m, and response bits (tr_m) of m is the sum of o_m, b_m, and f_m. We then define the complexity of a module m as

$$c_m = \max(ts_m, tr_m) \cdot pm + min(ts_m, tr_m) \tag{2}$$

Note that this is simply the test data volume of that particular module, neglecting the one cycle required to run the test. Given a set of modules M, the complexity of that set C_M is defined as the sum of the complexities of all its constituent modules i.e., $\sum_{m \in M} c_m$. Although similar to the ITC'02 [13] definition of complexity, our formulation is linear. This implies that irrespective of any partition of the modules M into M_1 and M_2, the sum of C_{M_1} and C_{M_2} will always result in C_M.

Given a set of modules M and P pins with which to test them, a lower bound on the test time of a 2D design based on the amount of data that needs to be pumped into it was given by [14], and can be re-written as:

$$LB_{2D}(M,P) = \left\lceil \sum_{m=1}^{|M|} \frac{c_m}{\lfloor P/2 \rfloor} - \sum_{m=1}^{|M|} \frac{\min(tr_m, ts_{m+1})}{\lfloor P/2 \rfloor} \right\rceil + \min_{m=1}^{|M|} p_m \tag{3}$$

Let M_{3D} be the set of all modules in our 3D stack. M_1 is the set of modules in the bottom die, and M_2 the set of modules in the top die. Let LB_{M_i} denote the lower bound of the test time of the set of modules M_i. First, we consider lower bounds induced by both the TSV and pin constraints. We assume that $TSV_{max} <= P_{max}$, as any additional TSVs will simply be wasted. The maximum test-pins available to the bottom and top dies are P_{max} and TSV_{max}, respectively. Therefore, a partition-dependant lower bound is given by

$$LB_{dep} = \max\{LB_{2D}(M_1, P_{max}), LB_{2D}(M_2, TSV_{max})\} \tag{4}$$

This lower bound can be improved by considering that every module in the 3D stack can be tested with no more than P_{max} pins. Such a lower bound is partition independent, and is given by

$$LB_{indep} = LB_{2D}(M_{3D}, P_{max}) \tag{5}$$

This lower bound holds irrespective of the partition or the TSV count. The overall lower bound is then given by the maximum of the partition independent and dependent lower bounds, and it can be reduced to

$$LB_{3D} = \max\{LB_{2D}(M_{3D}, P_{max}), LB_{2D}(M_2, TSV_{max})\} \tag{6}$$

Once our lower bound is defined, we want to see how this changes with the partition. We first need a metric that captures partition information. We define a complexity factor (CF) for a two-die stack as,

$$CF = \frac{C_{M_1}}{C_{M_1} + C_{M_2}} = 1 - \frac{C_{M_2}}{C_{M_1} + C_{M_2}} \tag{7}$$

By varying CF from 0 to 1, we cover all types of partitions. A CF of 0 means that all modules are in the top die, and a CF of 1 means that all modules are in the bottom die. We now see how the lower bound varies with the CF. It can be shown that the first term in Equation (6) is greater than the second term for low CF, and reduces with increasing CF until a certain threshold value is reached. Beyond this, the second term becomes greater, and since it is a constant, the lower bound is also a constant[1].

To calculate the value of this threshold, we develop a linear approximation of Equation (6). We make the assumption that the scan unload and scan load of successive modules are not overlapped. We also neglect the third term in Equation (3), as it is small when compared with the first. Then, we have:

$$LB'_{2D}(M,P) \approx 2 \cdot C_M/P \tag{8}$$

The lower bound then becomes

$$LB'_{3D} = 2 \cdot \max\left(\frac{C_{M_{3D}}}{P_{max}}, \frac{C_{M_2}}{TSV_{max}}\right) \tag{9}$$

The threshold complexity ratio is the complexity ratio when both terms are equal, and beyond which test time does not change. It is given by

$$CF_{th} = 1 - \frac{TSV_{max}}{P_{max}} \tag{10}$$

Note that this threshold value only depends on the TSV and pin constraints and not on the actual design or partition.

With these simplifications, the approximate lower bound on the 3D test time can be written as

$$LB'_{3D} = 2C_{M_{3D}} \times \begin{cases} (1-CF)/TSV_{max} & 0 \le CF \le CF_{th} \\ 1/P_{max} & CF_{th} \le CF \le 1 \end{cases} \tag{11}$$

This gives us a linear model for the lower bound, with both design dependant and independent terms. The shape of the lower bound curve is independent of design and is shifted up and down depending on the overall design. What this linear model gives us is a way to predict what the lower bounds on the test time will be, without having any real partition information. The converse of Equation (10) can be used to find out the pareto optimal number of TSVs for a given partition. Given a partition P with complexity factor CF_P, $TSV_{t,po}$ can be given by

$$TSV_{t,po} = P_{max} \times (1 - CF_P) \tag{12}$$

This equation essentially finds the TSV count for which this partition is at the threshold complexity factor. Increasing the TSV count beyond this value implies that the first term in Equation (9) is greater than the second term, and since it is a constant, the test time does not reduce. This is the definition of $TSV_{t,po}$.

2) Test time versus lower bound: In this section, we plot the test time versus the CF, and see how different partitions affect the test time. In addition, we plot the approximate lower bound on the same scale to see how the test time curve compares to the lower bound curve, as shown in Figure 5. As expected, the test time curve follows the general shape of the lower bound, but is shifted upwards by some amount. Most importantly, the threshold complexity factor CF_{th} for both the test time and the lower bounds is similar. Therefore, the lower bound gives the designer a very good estimate of what the shape of the test time curve is. Therefore, $TSV_{t,po}$ is well estimated by Equation (12).

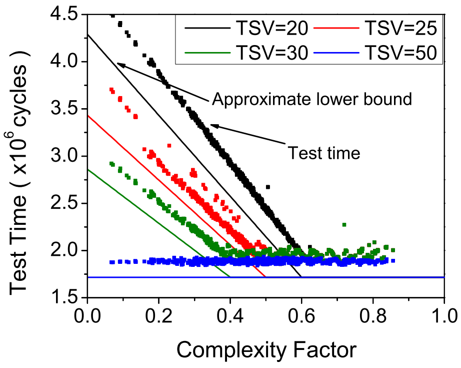

Fig. 5. Comparison between the measured test time and approximate lower bound of test time (= Equation 11) for a 2 die stack. We assume 50 test-pins and 4 different TSV constraints.

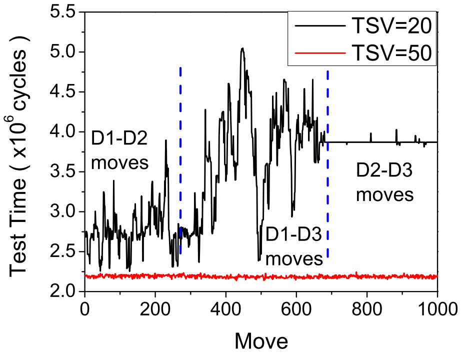

Fig. 6. Variation in test time observed while performing 1000 random moves, starting with ckt3_p1. The test time is computed assuming 50 test-pins, and under 2 different uniform TSV constraints (20 vs 50 per-die).

B. Multi-die stack

Similar to the experiment done with two dies, we use ckt3_p1 as our initial design. Then, we make 1000 random moves and observe the variation in test time. We focus on specific kinds of moves. The first one $1/3$ moves are performed only between Die 1 and Die 2. The next $1/3$ are only between Die 1 and Die 3. The third and final $1/3$ is made between Die 2 and Die 3. The test time is computed using ILP with a test-pin constraint of 50 and 2 different uniform TSV constraints. The results obtained are plotted in Figure 6.

From these results, we again see that if sufficient TSVs are available, the test time does not vary much, indicating that all partitions have at least $TSV_{t,po}$ TSVs. If, however, we do not have sufficient TSVs, there is significant variation in the test time. Most interestingly however, similar to the die-level partitioning, moves between the upper dies do not change the test time. These results are explained on the basis of lower bounds on test time, as described next.

1) Lower bound on test time: In this section, we generalize the results obtained for the two-tier case. We start at the top die and calculate its lower bound. Next, we calculate the lower bound on the top two dies similar to the two die case. This procedure is repeated one die at a time until we reach the bottom die. Then, this lower bound is linearized similar to the procedure followed for the two die

[1] A formal proof is provided in Appendix Section B.

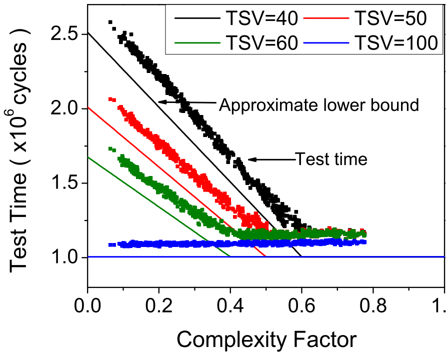

Fig. 7. Comparison between the measured test time and approximate lower bound for a four-die stack. We assume 100 test-pins and 4 different uniform TSV constraints.

case. If D denotes the set of all dies, we obtain[2]:

$$LB'_{3D} = \max \left\{ \max_{i=2}^{|D|} \frac{2 \cdot \sum_{j=i}^{|D|} C_{M_i}}{TSV_{max,i}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}} \right\} \tag{13}$$

This equation is general and applies to both tapered and uniform TSV constraints. Assuming uniform TSV constraints, say TSV_{max}, we get

$$LB'_{3D,eq} = \max \left\{ \frac{2 \cdot \sum_{j=2}^{|D|} C_{M_i}}{TSV_{max}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}} \right\} \tag{14}$$

This shows that the lower bound is independent of the partition of the upper dies. For uniform TSV constraints, we define a complexity factor

$$CF = \frac{C_{M_1}}{C_{M_{3D}}} = 1 - \frac{\sum_{j=2}^{|D|} C_{M_i}}{TSV_{max}} \tag{15}$$

Note that this CF has a slightly different meaning from that of the two-die case. Here, if $CF = 1$, then all modules are in the bottom die as usual, but a CF of 0 simply means that no modules exist in the bottom die. Using this definition, we get identical definitions of the threshold complexity factor CF_{th}, and $TSV_{t,po}$ as the two-die case.

2) Test time versus lower bound : Here we plot the test time vs CF for a four-die circuit using ckt4_p1, and performing 1000 different moves. The test-pin constraint is assumed to be 100, and we assume a uniform TSV constraint. We deliberately choose TSV numbers such that the TSV-to-pin ratio is the same as that of the two-die case. This would imply that the shape of the approximate lower bounds is exactly the same but with a different magnitude. The purpose of this is to demonstrate that different circuits tested under the same TSV-to-pin ratio indeed have similar test time curves. This is plotted in Figure 7. As observed from this figure, the slope of the test time curve as well as the threshold complexity values are dependent only on the TSV and pin constraints, and not on the circuit being tested. This implies that Equation (12) gives us a good estimate of $TSV_{t,po}$, even for more than two tiers.

V. Case Studies

In this section, we pick benchmark circuits from the IWLS'05 benchmark suite, and observe how the developed theory applies to it. We choose two circuits, the details of which are given in Table IV. ATPG for each module is performed using Synopsys Tetramax, and this table lists the average and standard deviation of test data volume (TDV) among all modules. We also assume uniform TSV constraints in all experiments involving more than two dies.

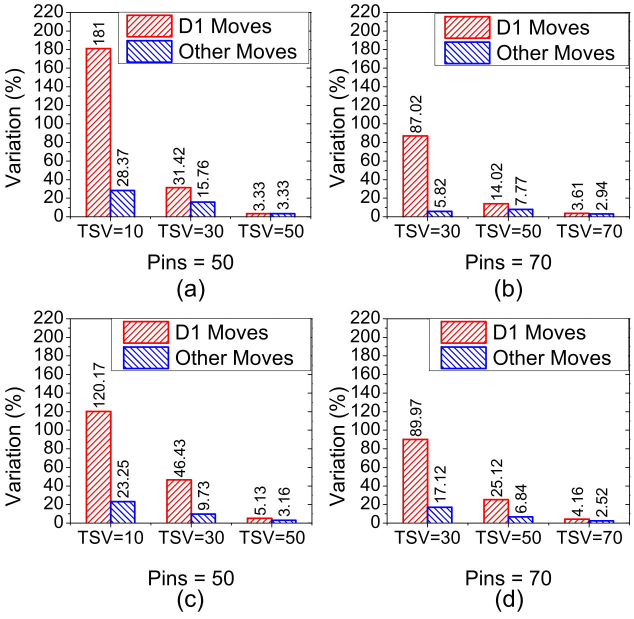

Fig. 8. Comparison of the variation in test time observed between moves involving the bottom die (= D1 moves), and all other moves. The numbers are reported for four-die implementations of (a,b) b19, (c,d) des_perf.

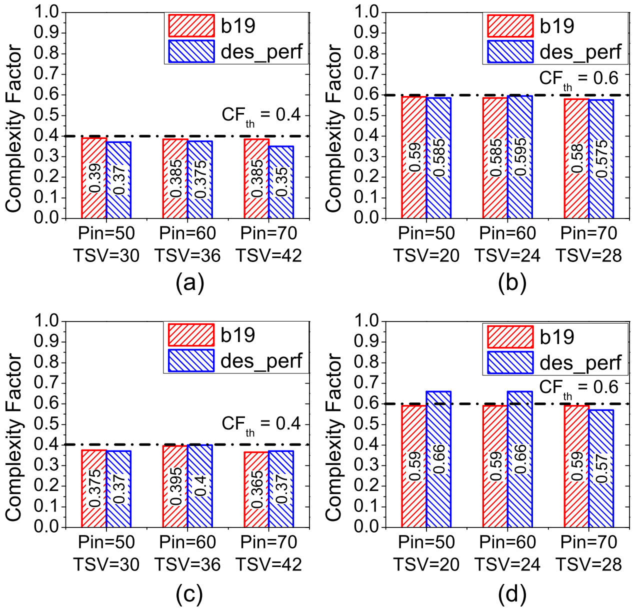

Fig. 9. Comparison of theoretical and experimental threshold complexity factors under various TSV and pin constraints. (a,b) Two-die stack, (c,d) Four-die stack.

A. Test time variation

In this experiment, we wish to confirm that different partitions with the same bottom die have similar test time. This will justify our definition of complexity factor, which in turn translates to a more accurate $TSV_{t,po}$. We start with four die implementations of the two benchmarks, and first perform 500 moves that change the complexity

[2] A detailed derivation is provided in Appendix Section C.

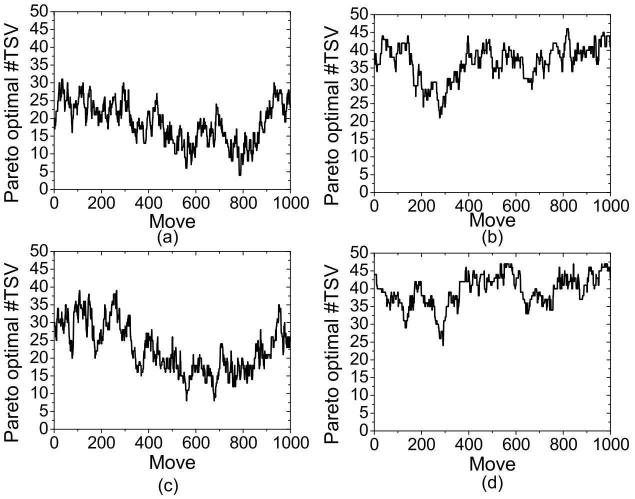

Fig. 10. The variation in $TSV_{t,po}$ observed while performing 1000 different random moves, assuming 50 test-pins. (a) b19 two-dies, (b) b19 four-dies, (c) des_perf two-dies and (d) des_perf four-dies.

TABLE IV
DETAILS OF BENCHMARK CIRCUITS USED, SHOWING THE AVERAGE AND STANDARD DEVIATION OF THE TEST DATA VOLUME AMONG ALL MODULES.

Circuit	#Modules	Average TDV	Std.Dev TDV
b19	57	141,489	168,833
des_perf	51	18,820	18,857

of the bottom die. Next, we make an additional 500 moves that change the complexity of the upper dies but maintain the bottom die constant. We plot the variation observed for each type of move in Figure 8. The variation is computed as $(t_{max} - t_{min})/t_{min}$, where t_{max} and t_{min} are the maximum and minimum test times respectively. We observe that moves involving the bottom die have significantly higher variation when compared with moves that do not, confirming our assumption. We also observe that if the test-TSV constraint is increased, the variation in the moves involving the bottom die is decreased. This is because with increased test-TSV constraints, a greater fraction of all possible partitions already meet $TSV_{t,po}$.

B. Threshold complexity factor prediction

The correct prediction of CF_{th} is important, as it directly translates in to the correct prediction of $TSV_{t,po}$. Theoretically, it is computed by Equation (10). According to this equation, CF_{th} is independent of design and only depends on the ratio between TSV and pin constraints. We want to see how this holds up in practice.

The experimental CF_{th} is computed as follows. We consider one thousand partitions of a design, and compute the CF and test time of each one. We create bins with respect to CF, with a bin size of 0.005. For each bin, we compute the average test time of all the partitions (using ILP) that lie within that bin. The threshold CF is computed as the first bin for which the test time is within 10% of the minimum test time observed for that particular pin and TSV constraint.

We plot the theoretical and experimental results observed in Figure 9. We consider different TSV and pin constraints that lead to the same CF_{th}. We also plot this for two and four die implantations of both designs. From this figure, we see that the theoretical formula does indeed give results close to the experimentally observed ones, which means that we can quickly and accurately estimate CF_{th}, and equivalently $TSV_{t,po}$.

C. Over-design reduction

In this section, we compute $TSV_{t,po}$ during a simulated partitioning process, and observe how it varies. The partitioning process is simulated by taking an initial circuit, and performing 1000 different random moves on it. The results are plotted assuming 50 test-pins in Figure 10. From this figure, we observe that if we use a fixed TSV constraint, that there is the possibility of over-design depending on what that constraint is. If it is quite low (e.g., 10), then the $TSV_{t,po}$ is always greater than this, and no resources are wasted. If however the fixed TSV constraint is high (e.g., 40), then the actual number of TSVs required can be much lesser than this, and correct prediction of $TSV_{t,po}$ helps eliminate resource wastage. We also observe that increasing the number of tiers increases $TSV_{t,po}$. This is expected, as more tiers require more TSVs to test them with minimum test time.

VI. CONCLUSION

In this paper, we have demonstrated the impact of both partitioning, and the number of TSVs on test time. Results show that different partitions of the same design can obtain comparable test time, if the test-TSV budget is varied accordingly. We have also derived the lower bounds on the test time of a 3D-IC, and used it to find out the pareto-optimal test-TSV count ($TSV_{t,po}$) for a given partition. For the multi-tier case with uniform TSV constraints, we have shown that this number is primarily determined by the complexity of the bottom tier, and moving modules between the upper tiers have little impact. Finally, we have validated our predictions, and shown that significant room for over-design reduction exists.

REFERENCES

[1] M. Pathak, Y. J. Lee, T. Moon, and S. K. Lim, “Through Silicon Via Management during 3D Physical Design : When to Add and How Many ?” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2010.

[2] E. J. Marinissen and Y. Zorian, “Testing 3D Chips Containing Through Silicon Vias,” in *Proc. IEEE Int. Test Conference*, 2009.

[3] D. Lewis and H. H. S. Lee, “A Scan Island Based Design Enabling Pre-Bond Testability in Die Stacked Microprocessors,” in *Proc. IEEE Int. Test Conference*, 2007.

[4] B. Noia and K. Chakrabarty, “Pre-bond probing of TSVs in 3D stacked ICs,” in *Proc. IEEE Int. Test Conference*, 2011.

[5] E. Marinissen, J. Verbree, and M. Konijnenburg, “A structured and scalable test access architecture for TSV-based 3D stacked ICs,” in *IEEE VLSI Test Symposium*, 2010.

[6] H. Lee and K. Chakrabarty, “Test Challenges for 3D Integrated Circuits,” *IEEE Design Test of Computers*, vol. PP, no. 99, 2009.

[7] X. Wu, P. Falkenstern, K. Chakrabarty, and Y. Xie, “Scan Chain Design and Optimization for Three-Dimensional Integrated Circuits,” *ACM Journal on Emerging Technologies in Computing Systems*, 2009.

[8] X. Zhao, D. Lewis, H.-H. S. Lee, and S. K. Lim, “Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2009.

[9] D. Lewis, *et al.*, “Designing 3D test wrappers for pre-bond and post-bond test of 3D embedded cores,” in *Proc. IEEE Int. Conf. on Computer Design*, 2011.

[10] B. Noia, *et al.*, “Test-Architecture Optimization and Test Scheduling for TSV-Based 3-D Stacked ICs,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, nov. 2011.

[11] B. Noia, *et al.*, “Test Architecture Optimization for TSV Based 3D Stacked ICs,” in *Proc. European Test Symposium*, 2010.

[12] D. H. Kim, R. O. Topaloglu, and S. K. Lim, “Block-Level 3D IC Design with Through-Silicon-Via Planning,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2012, pp. 335–340.

[13] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, “A Set of Benchmarks for Modular Testing of SOCs,” in *Proc. IEEE Int. Test Conference*, 2003.

[14] S. K. Goel and E. Marinissen, “SOC Test Architecture Design for Efficient Utilization of Test Bandwidth,” *ACM Trans. on Design Automation of Electronics Systems*, 2003.

Appendix

We provide detailed derivations and proofs of some of the equations and theorems presented in this paper. The terminology used in this section is as follows. M_{3D} is the set of all modules in the design. D is the set of all dies, and our design is partitioned into dies $D_1, \dots, D_{|D|}$. The modules allocated to die D_i is M_i. The test pin constraint is P_{max}, and the maximum number of test TSVs allowed between D_i and D_{i-1} is $TSV_{max,i}$ (either dedicated or multiplexed). $LB_{2D}(M, P)$ is the lower bound on test time of the set of modules M, with P pins, and is given by Equation (3).

A. Proof of Claim in Section III-B

In this section, we formally prove that for die-level partitioning under uniform TSV constraints, the partition of the upper dies does not affect test time.

Lemma 1: Assume that TSV_{max} is a uniform TSV constraint to test the set of dies D. Let $D_p \subseteq D$ be a subset of the dies tested in parallel within a single test session. Let $p_d = (p_1, \cdots, p_{|D_p|})$ be a division of pins within this test session. If we swap two dies D_i and D_j , $i \neq j \neq 1$ within the die stack , then p'_d obtained from p_d by swapping p_i and p_j does not violate P_{max} and TSV_{max} constraints.

Proof: The number of TSVs in die k (TSV_k) satisfies

$$TSV_k = \max_{l=k}^{|D|} \sum_{m=l}^{|D|} p_m \leq TSV_{max} \;\; \forall k > 1 \tag{16}$$

Since the set of dies D is known to be tested with p_d, we know that Equation (16) is satisfied. We need to prove that is also satisfied if D' is tested with $p'd$. Clearly, the greatest term in Equation (16) occurs when $k = 2$, or at the die immediately above the bottom die. Therefore $\sum_{m=2}^{|D|} p_m$ satisfies the TSV_{max} constraint. If D' is tested with p'_d, this sum does not change, and therefore p'_d also satisfies the TSV_{max} constraint. ■

This lemma proves that if two dies are tested in parallel, and then interchanged in the stack, they can still be tested in parallel with the same division of pins. It does not claim that the same old division of pins will be optimal for the new partitioning, just that it is possible without violating TSV and pin constraints.

Lemma 2: If the set of dies D is tested with a certain test schedule (with uniform TSV_{max} constraints), then any different partition D' with the same bottom die D_1, can be tested with the same test schedule.

Proof: A test schedule is merely a series of test sessions with dies tested in parallel within the same test session. Since TSVs are multiplexed between two different sessions, it is enough to show that a single test session can be repeated for D'. From the previous lemma, the test session can be repeated for a different partition with two dies interchanged. It is clear that D' can be obtained from D with a series of two die exchanges. Therefore D' can also be tested with the same test schedule. ■

Again, this lemma does not claim that the same test schedule is optimal for the new partition, but simply that it is possible. Finally, we prove that the test time is independent of the partition of upper dies.

Theorem 1: All partitions of a set of dies D with same bottom die D_1 have the same test time under a uniform TSV_{max} constraint.

Proof: Let D_{all} be the set of all partitions of D with the same bottom die D_1. Using identical TSV_{max} constraints, find the partition with the minimum test time, say D_{min}. Then, from the previous lemma, any other partition $D' \in D_{all}$ can be tested with the same test schedule as D_{min}, and hence also has minimum test time. ■

Tables II and III also show that if the number of test pins is equal to the number of test TSVs, then all partitioning results have the same test time. The proof of this follows from the fact that if $P_{max} = TSV_{max}$, lemma 1 holds for interchanging any two dies including the bottom die.

B. Proof of Claim in Section IV-A

For a block-level two-tier design, we assumed that there exists a threshold complexity factor, above which the test time is constant. We justify this assumption by the following proof

Theorem 2: $LB_{2D}(M_2, TSV_{max})$ decreases with increasing CF, and intersects LB_{indep} for all values of $TSV_{max} < P_{max}$.

Proof: The first statement is trivial. If CF increases, it implies that C_{M_2} reduces, and this will reduce the lower bound on M_2. Next, when $CF = 0$, all the modules are in Die 2, M_{top} becomes M_{3D}. Since $TSV_{max} < P_{max}$, $LB_{2D}(M_{3D}, TSV_{max}) > LB_{2D}(M_{3D}, P_{max})$. When $CF = 1$, the top die is empty with lower bound zero, and therefore, $LB_{2D}(M_{top}, TSV_{max}) < LB_{indep}$. This shows that somewhere in between a CF of 0 and 1, they intersect. ■

C. Derivation of Equation (13)

We start at the top die and work our way downwards. For the top-most die, the lower bound on test time can be written as

$$LB_{M_{|D|}} = LB_{2D}(M_{|D|}, TSV_{max,|D|}) \tag{17}$$

For the die $|D| - 1$, the lower bound can be written as

$$LB_{M_{|D|-1}} = LB_{2D}(M_{|D|-1}, TSV_{max,|D|-1}) \tag{18}$$

However, we also have to consider the fact that all the modules in the upper two dies can be tested with at most $TSV_{max,|D|-1}$ TSVs. We get

$$LB_{M_{|D|,|D|-1}} = LB_{2D}(M_{|D|} \cup M_{|D|-1}, TSV_{max,|D|-1}) \tag{19}$$

The true lower bound on the test time of the upper two dies is simply the maximum of Equations (17), (18), and (19). Inductively, we can work backwards defining similar lower bounds on all dies except the last die. The lower bound of test time to test all upper tiers can be written as

$$LB_{D-D_1} = \max_{i=2}^{|D|} \{LB_{2D}(\cup_{j=i}^{|D|} M_j, TSV_{max,i})\} \tag{20}$$

This is the time to test the upper die with $TSV_{max,|D|}$ TSVs, the upper two dies with $TSV_{max,|D|-1}$ and so on. The test time of the entire 3D stack can than be given by.

$$LB_{3D} = \max(LB_{3D-D_1}, LB_{2D}(M_{3D}, P_{max})) \tag{21}$$

This is a general equation, for arbitrary TSV constraints. However, for the special case when all the TSV constraints are equal, say TSV_{max}, this can be reduced to

$$LB_{3D,eq} = \max(LB_{2D}(\cup_{i=2}^{|D|} M_i, TSV_{max}), LB_{2D}(M_{3D}, P_{max})) \tag{22}$$

Approximate formulae can then be obtained by linearisation

$$LB'_{3D} = \max \left\{ \max_{i=2}^{|D|} \frac{2 \cdot \sum_{j=i}^{|D|} C_{M_i}}{TSV_{max,i}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}} \right\} \tag{23}$$

If we have uniform TSV constraints TSV_{max}, then we get

$$LB'_{3D,eq} = \max \left\{ \frac{2 \cdot \sum_{j=2}^{|D|} C_{M_i}}{TSV_{max}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}} \right\} \tag{24}$$