Optimization Methods for Post-Bond Die-Internal/External Testing in 3D Stacked ICs*

Brandon Noia¹, Krishnendu Chakrabarty¹, and Erik Jan Marinissen²
¹Dept. Electrical and Computer Engineering
² IMEC

Duke University
Durham, NC 27708, USA
{brn2, krish}@ee.duke.edu

Kapeldreef 75
B-3001 Leuven, Belgium
erik.jan.marinissen@imec.be

Abstract

Testing of three-dimensional (3D) stacked ICs (SICs) is starting to receive considerable attention in the semiconductor industry. Since the die-stacking steps of thinning, alignment, and bonding can introduce defects, there is a need to test multiple subsequent partial stacks during 3D assembly. We address the problem of test-architecture optimization for 3D stacked ICs to minimize overall test time when either the complete stack only, or the complete stack and multiple partial stacks, need to be tested. We show that optimal test-architecture solutions and test schedules for multiple test insertions are different from their counterparts for a single final stack test. In addition, we present optimization techniques for the testing of TSVs and die-external logic in combination with the dies in the stack.

1 Introduction

As the semiconductor industry continues to scale CMOS technology to smaller feature sizes, long interconnects have emerged as the dominant contributor to circuit delay and a major cause for high power consumption. Three-dimensional (3D) integrated circuit (IC) technology provides many benefits for chip design [1, 2], including a reduction in the average interconnect length and alleviation of many of the problems caused by long global interconnects [3, 4, 5]. Since die can be stacked in a 3D environment, on-chip data bandwidth can be increased as well. Furthermore, since 3D ICs can scale “up” instead of “out”, higher packing density and smaller footprint can be achieved.

Advances in manufacturing have led to direct stacking and bonding of die on die, with short interconnects running between die. In this work, we focus on 3D stacked ICs (3D-SICs) that use through-silicon via (TSV) vertical interconnects, as this approach offers the promise of high vertical interconnect density. 3D-SICs can be created by placing multiple device layers together through wafer or die stacking, and connecting metal layers from die to die using TSVs [5]. A 3D-SIC using TSVs allows for short, high-density vertical interconnects between thinned dies. Memories have already been successfully manufactured with this technology [6] and stacks that include memory stacked on logic die [7], or multiple stacked logic die [8] are likely to be seen in the near future.

TSV-based 3D-SICs are likely to make a significant impact on core-based system-on-chip (SOC) design. The use of embedded cores in SOC provides numerous benefits for both design and test, so it is likely that embedded cores will be seen in 3D-SICs as well. Testing of core-based dies in 3D-SICs introduces new challenges [9, 10]. In order to test the dies in a stack, the embedded

cores, and the TSVs, a test access mechanism (TAM) must be included on the die to transport test data to the cores, and a 3D TAM is needed to transfer test data to the die from the stack input/output pins. In a 3D-SIC, a test architecture must be able to support testing of individual dies as well as testing of partial and complete stacks [10]. Furthermore, test-architecture optimization must not only minimize the test time, but also minimize the number of dedicated test TSVs used to route the 3D TAM, as each TSV has area costs associated with it and is a potential source of defects. Test bandwidth constraints due to a limited number of package pins available only at the lowest layer in a 3D stack must also be considered.

Compared to two-dimensional ICs that typically require two test insertions, namely wafer test and package test, 3D stacking introduces a number of natural test insertions [10]. Since the die-stacking steps of thinning, alignment, and bonding can introduce defects, there is a need to test multiple subsequent (partial) stacks during assembly. Figure 1 shows an example manufacturing and test flow for a 3D stack. First, wafer test (i.e., pre-bond test) can be used to test die prior to stacking to ensure correct functionality, as well as to match die in a stack for power and performance. Next, Die 1 and Die 2 are stacked, and then tested again. This is likely to be the first time the TSVs between Die 1 and Die 2 will be tested due to technology limitations that make pre-bond test of TSVs

Figure 1: 3D-SIC manufacturing and test flow with multiple test insertions.

*The work of Brandon Noia was supported in part by a Master's Scholarship from the Semiconductor Research Corporation (SRC).

infeasible [9]. This step also ensures that we can detect defects in the stack due to additional 3D manufacturing steps such as alignment and bonding. Then the third die is added to the stack and all die in the stack, including all TSV connections, are retested. Finally, the “known good stack” is packaged and the final product is tested. Optimization methods are needed to minimize test time not only for the final stack test, i.e., if the intermediate (partial) stacks are not tested, but also to minimize the total test time if the final stack and partial stacks are tested during bonding.

In this paper, we present optimization methods for 3D-SICs with *hard dies*, in which a test architecture already exists on each die. We minimize the test time considering all possible stack tests and the complete stack, while considering die-external tests as well. These optimization methods allow us to efficiently generate multiple options for testing a 3D-SIC. While it is possible to have multiple dies on a given layer in a stack, we only consider one die per layer. Also, a core is considered to be part of a single die only, i.e., we do not consider “3D cores” as they are not likely in the immediate future of 3D-SICs [10].

The key contributions of this paper are as follows:

- Generalized and rigorous optimization methods to minimize test time for multiple test insertions. These methods also provide optimal solutions for several additional problem instances of interest, e.g., final stack test only, and testing of any subset of partial stacks.
- Methods to support multiple test schedules for a test-access architecture that is defined to be globally optimal for multiple test insertions.
- Optimization techniques that consider die-external and die-internal tests during test-time minimization.
- Optimization methods that are compatible with a recently proposed die-wrapper architecture [11], instead of requiring unrealistic assumptions on a specific wrapper for each case.

The rest of this paper is organized as follows. Section 2 provides an overview of related prior work. Section 3 uses simple examples to motivate this work and outlines how a 3D TAM can be designed and optimized. Section 4 presents integer linear programming (ILP) models for three TAM optimization problems—one considering multiple test insertions, and two considering stack TSV testing. Section 5 presents experimental results for various stacks constructed from several SOC from the ITC 2002 SOC test benchmarks [14]. Finally, Section 6 presents conclusions drawn from this work. An appendix with a complete ILP model for the first problem is included at the end of this paper.

2 Prior Work

As interest in 3D-SICs has increased, many papers on the topic of 3D-SIC testing have been published. Heuristic methods for designing core wrappers in 3D ICs were developed in [15]. ILP models for test architecture design for each die in a stack are presented in [16]. While these ILP models take into account some of the constraints related to 3D-SIC testing such as the TSV limit, this approach does not consider the reuse of die-level TAMs, multiple test insertions, or TSV tests. A TAM wire-length minimization technique based on simulated annealing is presented in [17]. While this work allows both pre-bond and post-bond tests, TAMs can start

and end on any stack tier, which is unlikely in a 3D-SIC. Heuristic methods for reducing weighted test cost while taking into account the constraints on test pin widths in pre-bond and post-bond tests are described in [18], but it still fails to allow for multiple schedules in multiple test insertions and die-external test during stacking.

In most prior work on 3D-SIC testing, TAM optimization is performed at die-level only, which leads to inefficient TAMs and non-optimal test schedules for partial/complete stack test. Furthermore, all previous methods assume that the designer can create TAM architectures on each die during optimization, which may not be possible in all cases. In [11] a die-level wrapper and associated 3D architecture is presented to allow for all pre-bond and post-bond tests. This approach relies on die-level wrappers and it leverages current standards, namely IEEE 1149.1 and IEEE 1500. In addition to functional and test modes, die-level wrappers allow bypass of test data to and from higher die in the stack and reduced test bandwidth during pre-bond tests. This is a realistic and practical look at test architectures in 3D-SICs, but it offers no insight into optimization and test scheduling. The optimization methods presented in this paper are compatible with the test architecture of [11], and they do not make any unrealistic assumptions on die wrappers or the 3D TAM.

In [19], the authors presented an expanded wrapper architecture for 2D ICs using modified wrapper cells in which each wrapper cell can be connected to two TAMs. As opposed to the 1500-standard wrapper (referred to in the rest of this paper as a “thin” or 1500-like wrapper), this expanded wrapper architecture, or “fat” wrapper, allows for core-external test (EXTEST) and core-internal test (INTEST) to be run in parallel. We consider both types of wrappers; in particular, the use of fat wrappers in this work is a natural extension of die-level wrappers to allow for die-external tests (TSV tests) and die-internal tests in parallel.

In our recent work, we introduced optimization techniques for minimizing the test time for final stack test [20]. A global limit was set on the number of dedicated TSVs to be used for test access and constraints were imposed on test bandwidth due to a limited number of test pins on the lowest die in the stack. A drawback of [20] is that it does not consider multiple test insertions for testing the partial stack. Another limitation is that the test time for TSVs and die-external logic is ignored in the optimization framework. Finally, in [20], TSV limits for the 3D TAM apply to the complete stack, and not on a per-layer basis.

In this paper, we generalize the optimization models from [20] to allow for multiple test schedules and optimization for any number of or all post-test stack tests. We maintain the test-bandwidth constraints but use more realistic constraints on dedicated test TSVs by considering a maximum number of TSVs per die, as opposed to a global limit. We further extend previous work by minimizing test time for die-internal and die-external tests using both fat and thin wrappers.

3 Problem Definitions

In a 3D-SIC, the lowest die is usually directly connected to chip I/O pins, therefore it can be tested using package pins. To test the other dies in the stack, TAMs that enter the stack from the lowest die should be provided. To transport test data up and down the stack, “test elevators” need to be included on each die except for

the highest die in the stack [10]. The number of test pins and test elevators, as well as the number of TSVs used, affect the total test time for the stack.

Many new manufacturing steps are needed for the production of 3D-SICs than for 2D-SICs, including TSV creation, wafer thinning, alignment, and bonding. These steps can introduce possible defects that do not arise for 2D-SICs [9]. Such defects include incomplete fill of TSVs, misalignment, peeling and delamination, and cracking due to back-side grinding. It is difficult to carry out pre-bond testing of TSVs due to limitations in probe technology and the need for contactless probing. Thus, post-bond partial-stack and in-stack TSV testing are needed to detect these new defects and reduce defect escapes.

In this paper, our optimization methods consider the use of IEEE 1500 style die level wrappers such as those proposed in [11]. We choose 1500-style wrappers for several reasons. They can be easily extended to the die level to provide a uniform interface for each die. They make no assumptions on the on-die TAM and they allow for greater flexibility. The test modes associated with a 1500-style wrapper, such as bypass and EXTEST, map naturally to the tests required for a 3D stack. The proposed optimization techniques, while requiring some means for test access on the lowest die, do not make any assumptions on the specific test methods being utilized. Thus, the proposed optimization methods can be used with any appropriate standard of choice.

It is important to note the costs associated with multiple test insertions for a 3D stack. Each additional test insertion increases the cumulative test time for the 3D stack, generally with a greater increase associated with a more complete stack. Multiple test insertions also require a single chip to be on a tester multiple times during stacking, which takes up valuable test resources. Our optimization methods do not explicitly take into account increased costs associated with multiple test insertions. Instead, it allows the designer to optimize a 3D test architecture for whichever test insertions he has deemed appropriate based on cost models [12, 13].

Figure 2: Example of 3D test access architecture [11].

Figure 2 shows an example of 3D test access architecture for a 3D stack with 3 dies, with the die on the left being the lowest die in the stack. In this example, 1149.1 style test access is used on the bottom die. As can be seen, a 3D TAM routes test data to and from each die in the stack, and a 2D TAM exists on each die for internal test data routing. Each die level wrapper contains its own instruction register for selecting between test modes such as bypass, INTEST, and EXTEST.

The following example highlights the limitations of optimization techniques that are oblivious to multiple test insertions. In [20], optimization decisions were made considering only the final stack test after all die have been bonded. These models cannot be directly applied to optimize for multiple test insertions, as shown in Figure 3.

Figure 3: Example 1: 3D-SIC with three hard dies.

Example: We attempt to optimize for two test insertions, the first with three die on the stack as in Figure 3(a) and the second with all four die as in Figure 3(b), by building upon the test architecture created for the first test insertion. A global TSV limit of 90 is imposed for this example. As seen in Figure 3(a), Die 2 and Die 3 are tested in parallel with each other and in series with Die 1, resulting in a minimized test time of 1500 cycles (the parallel test time of Die 2 and Die 3 is 500 cycles) and the use of 80 TSVs as test elevators. We then attempt to add the fourth die while preserving the previous architecture, but this leads to a violation of the TSV limit since we now need 100 TSVs (the test elevators marked in the figure exceed the mandated upper limit). If we instead started from the complete stack and worked backwards, we obtain suboptimal results for the test time. This is because the architecture created for the final stack test will not be able to support optimal test schedules for other intermediate test insertions. Therefore, new optimization techniques are needed for partial stack testing.

With the above example as motivation for better optimization methods, we now formally state the problems addressed in this work. The problem of test-architecture optimization for 3D-SICs with hard dies for all stack test insertions is defined as follows.

Problem 1 [3D-SIC with Hard Dies and Multiple Test Insertions (P_{MTS}^H)]

We are given a stack with a set M of dies and total number of test pins W_{max} available for test. For each die $m \in M$, we are given its tier number l_m in the stack, the number of test pins w_m ($w_m \leq W_{max}$) required to test the die, the associated test time t_m , and a maximum number of TSVs (TSV_{max_m}) that can be used for TAM design between die $m - 1$ and m ($m > 1$). The goal is to determine an optimal TAM design for the stack and test schedule for each stage of stacking such that the total test time T , i.e., the sum of the test times for all the desired stack tests (final stack test or multiple test insertions), is minimized and the number of TSVs used per die does not exceed TSV_{max_m} . $\square$

We further consider testing of TSVs and die-external logic along with the cores in each die. There are two variants to this prob-

lem; see Figure 4. Figure 4(a) shows a three-die stack with a test architecture using fat wrappers to allow TSV tests to take place in parallel with module testing on the dies. Each die has a separate TAM for die-external tests, both to higher and lower die in the stack. Each TAM has its own width and utilizes different test pins. In this case, test pins used for TSV testing are not available for die testing. For example, internal test of Die 2 and external test of the TSVs and die-external logic between Die 2 and Die 3 can occur simultaneously, utilizing a total of 23 test pins. The second variant uses a thin wrapper which allows die-external tests to only take place serially with module testing on the die as seen in Figure 4(b). This design allows the TAMs for die-external test to access the full test width for the die, but test pins are shared between all TAMs on the die. In this architecture, the external tests for logic and TSVs between Die 2 and Die 3 can be carried out in parallel with the internal test of Die 1, but not in parallel with any other test.

Figure 4: Example of a test architecture in a 3D-SIC including die-external tests.

Below we present the problem definitions for these two variants, where “||” refers to parallel and “—” refers to serial. The problems of testing hard die including TSV tests for a single test insertion (final stack test only) are stated below as Problem 2(a) and Problem 2(b), respectively.

Problem 2 [3D-SIC with Hard Dies and (a) fat wrapper ($P_{DTSV,||}^H$) or (b) thin wrapper ($P_{DTSV,-}^H$)]

We are given a stack with a set M of dies and the total number of test pins W_{max} available for test. For each die $m \in M$, we are given its tier number l_m in the stack, the number of test pins w_m ($w_m \leq W_{max}$) required to test the die, the associated test time t_m , and a maximum number of test elevators ($TSVmax_m$) that can be used for TAM design between die $m-1$ and m . We are also given the number of functional TSVs (Tf_m) and test width given to the TSVs (Wt_{max}) between die $m-1$ and m , and the number of test patterns (Pf_m) for the functional TSVs between die $m-1$ and m (for $m > 1$). The goal is to determine an optimal TAM design and a test schedule for the stack for die-internal and die-external tests such that the total test time T is minimized and the number of test TSVs used per die does not exceed $TSVmax_m$. $\square$

All three problems presented above are $\mathcal{NP}$ -hard from “proof by restriction” [21], as they can be reduced using standard techniques to the rectangle packing problem, which is known to be $\mathcal{NP}$ -hard [22]. For example, for Problem P_{MTS}^H , if we remove the constraints related to maximum number of TSVs and consider only the final stack test insertion, each die can be represented as a rect-

angle with a width and height, where width is equal to its test time for its given TAM width and height is equal to the number of required test pins. Now we need to pack all these rectangles (dies) into a bin with width equal to the total number of test pins and height equal to the total test time for the stack, which needs to be minimized. Despite the $\mathcal{NP}$ -hard nature of these problems, they can be solved optimally since the number of layers in a 3D-SIC is expected to be limited, e.g., up to four layers have been predicted for logic stacks [23].

4 Test-Architecture Optimization

In this paper, we use ILP to solve the problems defined in the previous section. The problem instances in practice are relatively small for realistic stacks with anywhere from two to eight dies, therefore ILP methods are good candidates for solving these optimization problems.

4.1 ILP Formulation for Problem P_{MTS}^H

To create an ILP model for this problem, we need to define the set of variables and constraints. We first define a binary variable x_{ijk} , which is equal to 1 if die i is tested in parallel with die j for a test insertion when there are k die in the stack, and 0 otherwise. There are $M-1$ test insertions, one for each additional die added to the stack such that k ranges from 2 to M . Constraints on variable x_{ijk} can be defined as follows:

$$x_{iik} = 1 \quad \forall k, i \leq k \quad (4.1)$$

$$x_{ijk} = x_{jik} \quad \forall k, \{i, j\} \leq k \quad (4.2)$$

$$1 - x_{iqk} \geq x_{ikq} - x_{jqk} \geq x_{ijk} - 1 \quad \forall k, \{i, j, q\} \leq k, i \neq j \neq q \quad (4.3)$$

The first constraint indicates that every die is always considered to be tested with itself for every test insertion. The second constraint states that if die i is tested in parallel with die j for insertion k , then die j is also tested in parallel with die i for insertion k . The last constraint ensures that if die i is tested in parallel with die j for insertion k , then it must also be tested in parallel with all other dies that are tested in parallel with die j for insertion k .

Next, we define a second binary variable y_{ik} , which is equal to 0 if die i is tested in parallel with die j on a lower layer ($l_i > l_j$) for insertion k , and 1 otherwise. The total test time T for the stack is the sum of test times of all dies that are tested in series plus the maximum of the test times for each of the sets of parallel tested dies for all test schedules at every test insertion. Using variables x_{ijk} and y_{ik} , the total test time T for all test insertions with the set of dies M can be defined as follows.

$$T = \sum_{k=2}^{|M|} \sum_{i=1}^k y_{ik} \cdot \max_{j=i..k} \{x_{ijk} \cdot t_j\} \quad (4.4)$$

It should be noted that Equation (4.4) has two non-linear elements, the max function, and the product of y_{ik} variable and the max function. We linearize this by introducing two new variables. The variable c_{ik} takes the value of the max function for each die i for test insertion k and the variable u_{ik} represents the product $y_{ik} \cdot c_{ik}$. The variables u_{ik} and c_{ik} are defined using standard linearization

techniques. The linearized function for total test time can be written as follows.

$$T = \sum_{k=2}^{|M|} \sum_{i=1}^k u_{ik} \quad (4.5)$$

As number of test pins used for parallel testing of dies should not exceed the given test pins W_{max} across all test schedules for every test insertion, a constraint on the total number of pins used to test all dies in a parallel set in any given test insertion can be defined as follows for all k .

$$\sum_{j=1}^k x_{ijk} \cdot w_j \leq W_{max} \quad \forall i \leq k \quad (4.6)$$

Similarly, the total number of used TSVs should not exceed the given TSV limit (TSV_{max_m}) for each die face across all test insertions. It should be noted that TSV_{max_2} is the limit for the upper face of die 1 and the lower face of Die 2, TSV_{max_3} is for the upper face of die 2 and lower face of Die 3, and so forth. The number of TSVs used to connect layer i to layer $i - 1$ is the maximum of the number of pins required by the layer at or above layer i that takes the most test pin connections, and the sum of parallel tested die at or above layer i in the same parallel tested set across all test insertions. Based on this, we can define the constraint on the total number of TSVs used in a test architecture as follows.

$$\max_{i \leq k \leq |M|} \left\{ w_i, \sum_{j=i}^k w_j \cdot x_{kj} \right\} \leq TSV_{max_i} \quad \forall i \geq 2, \quad (4.7)$$

We can linearize the above set of constraints by representing the max function by a variable d_i . Finally, to complete the ILP model for Problem P_{MTS}^H , we must define constraints on binary variable y_{ik} and the relationship between binary variable y_{ik} and x_{ijk} . For this purpose, we first define a constant C that approaches but is less than 1. We then define y_{ik} as follows:

$$y_{1k} = 1 \quad \forall k \quad (4.8)$$

$$y_{ik} \geq \frac{1}{1-i} \sum_{j=1}^{i-1} (x_{ijk} - 1) - C \quad \forall k, i \leq k, i \neq 1 \quad (4.9)$$

The first equation forces y_{1k} to 1, since the lowest layer can not be tested in parallel with any layer lower than itself. Constraint (4.9) defines y_{ik} for the other layers. To understand this constraint, we first make the observation that the objective function (as shown in Equation (4.4)) would be minimized if each y_{ik} is zero. This would make the objective function value equal to 0, which is an absolute minimum test time. Thus, we only need to restrict y_{ik} to 1 where it is absolutely necessary, and then we can rely on the objective function to assign a value 0 to all unrestricted y_{ik} variables. This equation considers the range of values that the sum of x_{ijk} can take. The fraction in the equation normalizes the sum to a value between 0 and 1 inclusive, while the summation considers all possible cases for a die being in parallel with die below it. The complete ILP model is shown in Figure 11 in the Appendix.

Figure 5 illustrates the ILP model for P_{MTS}^H . The 3D-SIC in Figure 5 consists of three die, with test times of 1333098, 700665, and 106391 cycles moving up the stack. There are 22 test pins available, so W_{max} is 22. TSV_{max} is set to 20, such that there can be no more than 20 dedicated test TSVs between any two die

(this limits the TSVs per die to 40). There are two test insertions, the first when Die 1 and Die 2 are stacked and the second for the complete stack. In the first test insertion ($k = 2$), we calculate the optimal solution that Die 1 and Die 2 are tested in parallel. As such, $x_{1,1,2}$, $x_{1,2,2}$, $x_{2,1,2}$, and $x_{2,2,2}$ are all equal to 1. Die 2 is tested in parallel with a die below it (Die 1), so $y_{2,2}$ is 0 and $y_{1,2}$ is 1. The test time for this test insertion is 1333098 cycles, since $u_{1,2}$ is 1333098 and $u_{2,2}$ is 0.

For the second test insertion ($k = 3$), the optimal solution is to test Die 1 and Die 2 in parallel, and then test Die 3. Since Die 1 and Die 2 are tested in parallel again, $x_{1,1,3}$, $x_{1,2,3}$, $x_{2,1,3}$, and $x_{2,2,3}$ are equal to 1. For Die 3, $x_{3,3,3}$ is 1. All other x_{ijk} variables are 0. As before, $y_{2,3}$ is 0 since Die 2 is tested in parallel with Die 1. The variables $y_{1,3}$ and $y_{3,3}$ are 1. The test time for this test insertion is 1439489, as $u_{1,3}$ is 1333098, $u_{2,3}$ is 0, and $u_{3,3}$ is 106391. In this architecture, d_2 is 20 and d_3 is 14, neither of which violate the TSV limit. All 44 test pins are utilized.

Figure 5: Example of a test architecture in an optimized 3D-SIC with hard dies.

The above ILP model is a generalization of the special case presented in [20], in which test-time was minimized only for the final stack test. If the variable k is constrained to take only one value, namely M , in P_{MTS}^H , then optimization will produce a test architecture and test schedule that minimizes test time only for the final stack, i.e., the objective in [20]. We refer to optimization for only the final stack test as P_{FS}^H and P_{FS}^S for hard and soft die, respectively. Therefore, an advantage of the optimization model proposed here is that it is flexible—it can be easily tailored to minimize test time for any number of stack tests, from one final test to two or more intermediate test insertions. Multiple options can be automatically generated for testing the stack. For example, suppose we are interested in two test insertions—after the second die is bonded to the first die and the final stack test. By allowing k to now take two values, namely 2 and M , we can minimize the test time for the stack considering only these two insertions.

4.2 ILP Formulations for $P_{DTSV,||}^H$ and $P_{DTSV,--}^H$

The ILP formulation in Section 4.1 does not account for the TAM architecture and test time needed to test the TSVs. In order to include TSV tests, we treat the TSV connections between two layers as a “virtual” die. This die has a predefined test width, similar to an actual hard die for the model. In order to cal-

culate the test time for the TSV “layer”, we use the formula $\lceil T_{f_m} / \lfloor (pins/4) \rfloor \rceil (Pf_m + 1)$, where *pins* refers to the number of test pins available for TSV testing.

The above expression begins by taking the floor of *pins*/4. We assume that, between two die, there are any number of functional TSVs that are used for either input to the die or output from the die. The TSV tips are latched and the corresponding flip-flops are connected to form one or more scan-chains. Thus, the scan-flops are also assumed to be unidirectional. In order to quickly test all TSVs, we must allow for shift-in, shift-out, and capture on both sides of the TSVs simultaneously. This requires 4 test pins for each TSV scan-chain, i.e., an input and output pin for each scan-chain for both sides of the TSVs as seen in Figure 6(a). As can be seen from Figure 6(b), there is no reduction in test time for stack tests if we consider unequal numbers of TSV scan-chains for die on either side of the TSVs, as the bottleneck in test time would then be the die with the fewest TSV scan-chains (TSV testing requires use of TSV scan-chains on both sides of the TSVs in parallel). At least four more test pins must be added for a reduction in test time as shown in Figure 6(c). Thus, we evenly divide test-pin use among the die for TSV tests, and the number of scan-chains on either side of the TSVs is $\lceil T_{f_m} / \lfloor (pins/4) \rfloor \rceil$. We multiply this by the number of patterns required for TSV testing plus one, to accomodate for shift-in and shift-out operations. Without loss of generality, we do not consider the number of test pins required for control signals to a die-level wrapper for TSV testing.

Figure 7 shows the difference between fat and thin wrappers. In Figure 7(a), the die-internal and die-external TAMS utilize different test pins. Thus, EXTEST can be performed in parallel with either or both INTEST of Die 1 and Die 2. This is representative of fat wrappers. For thin wrappers, Figure 7(b) shows that the same test pins are utilized for die-internal and die-external tests, and as such test data must be multiplexed to the correct TAM. In this case, the INTEST of each die can be performed in parallel, but EXTEST cannot be performed in parallel with INTEST on Die 1 or Die 2 (though it can be in parallel with INTEST of other dies in the stack).

Figure 6: Example of scan chains for die-external tests.

The ILP formulation for 3D-SICs with hard cores including TSVs is derived in a similar manner as the 3D-SIC with hard cores in Problem 1. We begin by removing the subscript *k* from all variables that accounts for multiple test insertions—we only consider the final stack test. As stated before, we consider a set of TSVs between two layers to be a virtual die for purposes of optimization, such that Die 1 is the lowest die in the stack, Die 2 represents the TSVs between Die 1 and Die 3, Die 3 is the second die in the stack, Die 4 represents the TSVs between Die 3 and Die 5, and so on. In this way, odd-numbered die are actual die and even-numbered die

represent the TSVs between two odd-numbered dies.

Figure 7: Simplified illustration of fat wrappers and thin wrappers.

We must add variables and constraints in order to accurately model the dies representing TSVs. We begin by defining the variable $p_i = 1$ if a die *i* representing TSVs can be tested in parallel with the dies below and above it and 0 otherwise. In the case of fat wrappers, we leave p_i as a decision variable and in the case of thin wrappers we force p_i to 0 for all die representing TSVs. For actual (even-numbered) die, p_i is always 1. If a die representing TSVs can be tested in parallel with the die around it, then the number of test pins given to TSV testing reduces the number of pins available to the die for testing. Otherwise, the TSVs can use all the test pins that are utilized by the dies around it. We define the set *die* to contain all even-numbered die, or the actual die, in the stack. We then place the following restrictions on p_i for fat wrappers (the first constraint applies for thin wrappers as well):

$$p_i = 1 \quad \forall i \in \text{die} \quad (4.10)$$

$$p_{i+1} \leq w_i - w_{i+1} \quad \forall i \in \text{die}, i \neq |M| \quad (4.11)$$

$$p_{i+1} \leq w_{i+2} - w_{i+1} \quad \forall i \in \text{die}, i \neq |M| \quad (4.12)$$

This restricts p_i to 0 if all pins in a die are shared for TSV tests, and otherwise leaves the variable open to optimization. We use p_i to accurately represent the number of TSVs and test pins used in the stack and use these constraints to further restrict the variable p_i . We redefine the variable d_i as follows:

$$d_i \geq \sum_{j=k}^{|M|} w_j \cdot x_{kj} \quad \forall i \in \text{die}, i \neq |M|, k = i..|M| \quad (4.13)$$

$$d_i \geq w_j \quad \forall i \in \text{die}, i \neq |M|, j = i..|M| \quad (4.14)$$

$$d_i \geq w_j + p_{j-1} \cdot w_{j-1} + p_{j+1} \cdot w_{j+1} \quad \forall i \in \text{die}, j \geq i \quad (4.15)$$

Thus, we tally test elevator use only between actual die, but take into account the number of extra test elevators used for the TSV layer considering parallel testing. For thin wrappers, the last constraint on d_i is removed. For Die 1, the last constraint is instead $d_i \geq w_j + p_{j+1} \cdot w_{j+1}$. Test pin constraints therefore reduce to:

$$\sum_{j=1}^{|M|} x_{ij} \cdot w_j \leq W_{max} \quad \forall i \quad (4.16)$$

$$w_i + p_{i-1} \cdot w_{i-1} + p_{i+1} \cdot w_{i+1} \leq W_{max} \quad \forall i \in \text{die}, i \neq |M| \quad (4.17)$$

The constraint (4.20) accounts for the combined test-pin use by dies and TSVs, when they are tested in parallel or they are tested serially. This constraint is removed for thin wrappers, and for the first die instead reads $w_i + p_{i+1} \cdot w_{i+1} \leq W_{max}, \forall i \in \text{die}, i \neq |M|$.

It is further necessary to update the variable x_{ij} as follows:

$$x_{i,i+1} \leq p_{i+1} \quad \forall i \in \text{die}, i \neq |M| \quad (4.18)$$

$$x_{i+2,i+1} \leq p_{i+1} \quad \forall i \in \text{die}, i \neq |M| \quad (4.19)$$

For $P_{DTSV,-}^H$, we further add the constraint:

$$x_{i-1,i+1} = 0 \quad \forall i \in \text{die}, i \neq \{1, |M|\} \quad (4.20)$$

5 Experimental Results

In this section, we present experimental results for the ILP models given in Section 4. As benchmarks, we have handcrafted two 3D SICs (as shown in Figure 8) using several SOC test Benchmarks as dies inside SICs. These are equivalent to two of the benchmark 3D-SICs in [20]. The SOC test Benchmarks used are d695, f2126, p22810, p34292, and p93791. In SIC 1, the most complex die (p93791) is placed at the bottom, with die complexity decreasing as one moves up the stack. The order is reversed in SIC 2.

Figure 8: Two 3D SIC benchmarks.

To determine the test architecture and test time for a given die (SOC) with a given TAM width, we have used the TAM design method in [24] for daisychain TestRail architectures [25]. For problem instances with hard dies, the test times (cycles) and TAM widths for different dies are listed in Table 1. Note that test pins were assigned to dies based on their sizes in order to avoid very large test times for any individual die.

For a fixed value of TSV_{max} and range of values of W_{max} , Table 2 presents results for P_{MTS}^H for the two benchmark SICs. They are compared against optimized results for only the final post-bond stack test (referred to as P_{FS}^H). The ILP models were run and optimal results obtained using XPRESS-MP [26]. The CPU times for the experiments on an AMD Opteron 250 with four Gigabytes of memory was in the range of a few seconds to eight minutes.

Die	d695	f2126	p22810	p34392	p93791
Test Length	106391	700665	1333098	2743317	2608870
Test Pins	15	20	25	25	30

Table 1: Test lengths and number of test pins for hard dies used in optimization.

In this table, Column 1 lists the optimization model that was run and Column 2 lists the benchmark SIC. Column 3 shows the maximum number of TSVs allowed for each die (TSV_{max}), while Column 4 lists the number of available test pins W_{max} at the lowest die. Column 5 represents the total test length (cycles) for all four post-bond test insertions in a five die stack. Columns 6 gives the percent reduction in test time over the optimization case for only the final stack test. Let the test length for P_{FS}^H be T_1 and let the test length for P_{MTS}^H be T_2 . Then the percentage reduction is $((T_1 - T_2)/T_1) \cdot 100\%$. The percentage reductions for P_{FS}^H are zero by default. Columns 7 through 10 show the resulting test schedule for the 3D-SIC at each test insertion, where the symbol “||” indicates parallel testing of dies, and a “,” represents serial testing. Finally, Column 11 gives the number of test pins needed for each die. From Table 2 we can see that the proposed method can provide considerable reductions in test time over the optimization methods of [20] that is targeted towards final stack test.

(a) SIC 1

(b) SIC 2

Figure 9: The test time with respect to TSV_{max} and W_{max} for SIC 1 and SIC 2 with hard dies and multiple test insertions.

Optimization Problem	SIC	TSV_{max}	W_{max}	Test Length (cycles)	Reduction (%)	Test Schedule 1 (Die 1-2)	Test Schedule 2 (Die 1-2-3)	Test Schedule 3 (Die 1-2-3-4)	Test Schedule 4 (Die 1-2-3-4-5)	No. of Test Pins used per Die
P_{FS}^H	SIC 1	70	49	21254500	0.00	1,2	1,2 3	1,2 3,4	1 5,2 3,4	30,25,25,20,15
	SIC 1	70	50	19091000	0.00	1,2	1,2,3	1,2,3 4	1,2 5,3 4	30,25,25,20,15
	SIC 1	70	69	10819000	0.00	1 2	1 2 3	1 2 3,4	1 2 3,4 5	30,25,25,20,15
	SIC 1	70	70	10819000	0.00	1 2	1 2 3	1 2 3,4	1 2 3,4 5	30,25,25,20,15
P_{MTS}^H	SIC 1	70	49	12901800	39.30	1 2	1,2 3	1 4,2 3	1 4,2 3,5	30,25,25,20,15
	SIC 1	70	50	11042700	42.16	1 2	1,2 3	1 2,3 4	1,2 5,3 4	30,25,25,20,15
	SIC 1	70	69	9554160	11.69	1 2	1 2 3	1 3 4,2	1 2 3,4 5	30,25,25,20,15
	SIC 1	70	70	8959880	17.18	1 2	1 2 3	1,2 3 4	1 4 5,2 3	30,25,25,20,15

Table 2: Experimental results and comparisons for multiple test insertions.

For a different number of TSVs (TSV_{max}), Figure 9(a) and Figure 9(b) show the variation in test time T with an increase in the number of test pins W_{max} for SIC 1 and SIC 2. From the figures, we can see that both TSV_{max} and W_{max} determine which dies should be tested in parallel, and thus the total test time for the stack. For a given value of TSV_{max} , increasing W_{max} does not always decrease the test time for P_{MTS}^H , though test time never increases. Similarly, increasing TSV_{max} for a given W_{max} does not always decrease the test time. These Pareto-optimal points can easily be seen in Figure 9 for both benchmarks. Furthermore, we see that optimizing for the final stack test does not always reduce test time when we consider multiple test insertions. In fact, we often see that the test time is higher when we increase the values of TSV_{max} and W_{max} .

Figure 10: The test time with respect to TSV_{max} and W_{max} for SIC 1 and SIC 2 with hard dies including die-external tests.

It should be noted that for all optimizations with the same values of TSV_{max} and W_{max} , the stack configuration (SIC 2) with the largest die at the highest layer and the smallest die at the lowest layer is the best for reducing test time while using the minimum number of TSVs. This is because the most complex dies with the longest test times are tested in the fewest insertions. For example, the die at the top of the five-die stack is only tested once, while the die at the bottom is tested four times.

For $P_{DTSV,||}^H$, two extra test pins were added to the hard die for each EXTEST TAM on a die. For the highest and lowest dies in the stack, this implies an addition of two test pins, while for other dies it implies an addition of four test pins. For our problem instances, we assumed, without loss of generality, that each die had 10,000 functional TSVs, requiring 20 patterns. It has been reported that the number of tests for TSVs is likely to grow log-

arithmically with TSV count [10]. In order to make comparisons between $P_{DTSV,--}^H$ and $P_{DTSV,||}^H$, either two or four test pins were added to the total TAM width of the hard dies. As can be seen in Figure 10, there is a large dependence of test length on both TSV_{max} and W_{max} . These parameters and the hard TAM widths determine whether serial testing or parallel testing of TSVs with the dies is adopted to lower test times. For a majority of values for TSV_{max} and W_{max} for the TAM widths chosen, parallel testing leads to lower test times. We note here that these optimizations are only for the final stack test, so SIC 2 results in longer test times than SIC 1, as expected from previous work [20].

6 Conclusions

We have presented generalized optimization methods to minimize test time for a 3D-SIC, either for the final stack test or for any number of multiple test insertions during bonding. These methods provide us with optimized test schedules for each test insertion. We have presented optimization techniques that consider both die-internal and die-external test for both fat and thin die-wrappers. The proposed optimization methods incorporate constraints on both test bandwidth and the number of dedicated test elevators per die. Results have been presented for two different stack configurations with die made up of five SOC's taken from the ITC'02 SOC Test Benchmarks. These results show that optimization methods that only consider the final stack test provide significantly suboptimal results (higher test times) if multiple test insertions are carried out. Optimizations for hard die that take into account die-internal and die-external tests show that in general fat wrappers reduce test time more than thin wrappers.

References

- [1] K. Banerjee et al., "3-D ICs: a Novel Chip Design for Improving Deep-submicrometer Interconnect Performance and Systems-on-chip Integration", *Proc. IEEE*, vol. 89, no. 5, pp. 602-633, 2001.
- [2] R. Weerasekera et al., "Extending Systems-on-chip to the Third Dimension: Performance, Cost and Technological Tradeoffs", in *Int. Conference on Computer-Aided Design*, pp.212-219, 2007.
- [3] G. Loh, Y. Xie, and B. Black, "Processor Design in 3D Die Stacking Technologies", *IEEE Micro* Vol 27, No. 3, 2007, pp.31-48
- [4] W.R. Davis et al., "Demystifying 3D ICs: the Pros and Cons of Going Vertical", *IEEE Design and Test of Computers*, vol. 22, no. 6, pp. 498-510, 2005.
- [5] Y. Xie, G. H. Loh, and K. Bernstein, "Design Space Exploration for 3D Architectures", *J. Emerg. Technol. Comput. Syst.*, 2(2):65-103, 2006.
- [6] Uksong Kang et al. "8Gb 3D DDR3 DRAM Using Through-Silicon-Via Technology", In *Proceedings International Solid State Circuits Conference (ISSCC)*, pp. 130-132, February 2009.
- [7] C.C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari, "Bridging the Processor-Memory Performance Gap with 3D IC Technology," *IEEE Design & Test of Computers*, 22(6):556-564, November/December 2005.
- [8] B. Black, D.W. Nelson, C. Webb, and N. Samra, "3D Processing Technology and its Impact on iA32 Microprocessors", In *Proceedings International Conference on Computer Design (ICCD)*, pp. 316-318, October 2004.
- [9] H.-H.S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", *IEEE Design & Test of Computers*, vol. 26, pp. 26-35, September/October 2009

- [10] E.J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias", *International Test Conference*, E 1.1, 2009.
- [11] E.J. Marinissen, J. Verbree, and M. Konijnenburg, "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs", *VLSI Test Symposium*, 2010.
- [12] X. Dong and Y. Xie, "System-level Cost Analysis and Design Exploration for Three-dimensional Integrated Circuits (3D ICs)", *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, pp. 234-241, 2010.
- [13] Y. Chen, D. Niu, Y. Xie, and K. Chakrabarty, "Cost-effective Integration of Three-dimensional (3D) ICs Emphasizing Testing Cost Analysis", accepted for publication in *IEEE/ACM International Conference on CAD*, 2010.
- [14] E.J. Marinissen, V. Iyengar, and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs", In *International Test Conference*, pp. 519-528, Oct. 2002.
- [15] B. Noia, K. Chakrabarty, and Y. Xie, "Test-Wrapper Optimization for Embedded Cores in TSV-Based Three-Dimensional SOCs", *Proc. IEEE International Conference on Computer Design*, pp. 70-77, 2009.
- [16] X. Wu, Y. Chen K. Chakrabarty, and Y. Xie, "Test-access Mechanism Optimization for Core-based Three-dimensional SOCs", *IEEE International Conference on Computer Design*, pp.212-218, 2008.
- [17] L. Jiang, L. Huang, and Q. Xu, "Test Architecture Design and Optimization for Three-dimensional SoCs", *Design, Automation, and Test in Europe*, pp. 220-225, 2009.
- [18] L. Jiang, Q. Xu, K. Chakrabarty and T.M. Mak, "Layout-driven Test-architecture Design and Optimization for 3D SoCs under Prebond Test-pin-count Constraint", *Proc. IEEE International Conference on Computer-Aided Design*, pp.191-196, 2009.
- [19] S. K. Goel, E. J. Marinissen, A. Sehgal and K. Chakrabarty, "Testing of SOCs with Hierarchical Cores: Common Fallacies, Test-access Optimization, and Test Scheduling", *IEEE Transactions on Computers*, vol. 58, pp. 409-423, March 2009.
- [20] B. Noia, S.K. Goel, K. Chakrabarty, E.J. Marinissen, and J. Verbree, "Test-Architecture Optimization for TSV-Based 3D Stacked ICs", *European Test Symposium*, 2010.
- [21] M.R. Garey and D.S. Johnson, "Computers and Intractability A Guide to the Theory of NP-Completeness", New York: Freeman, 1979.
- [22] E.G. Coffman, Jr., M.R. Garey, D.S. Johnson and R.E. Tarjan. "Performance Bounds for Level-oriented Two-dimensional Packing Algorithms", *SIAM J. Computing*, vol. 9, pp. 809826, 1980.
- [23] X. Dong and Y. Xie. "System-level Cost Analysis and Design Exploration for 3D ICs", *Proceedings of Asia-South Pacific Design Automation Conference (ASP-DAC)*, pp.234-241, Jan. 2009.
- [24] S.K. Goel, E.J. Marinissen, "Control-Aware Test Architecture Design for Modular SOC Testing", *European Test Workshop*, pp. 57-62, 2003.
- [25] S.K. Goel and E.J. Marinissen. "SOC Test Architecture Design for Efficient Utilization of Test Bandwidth", *ACM Transactions on Design Automation of Electronic Systems*, 8(4):399429, 2003.
- [26] FICO. Xpress-MP.
<http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx>

Appendix

For the sake of completeness, we present the complete ILP model for P_{MTS}^H .

Objective:

$$\text{Minimize } \sum_{k=2}^{|M|} \sum_{i=1}^k u_{ik}$$

Subject to:

$$\begin{aligned} t_{max} &= \max_{i=1}^{|M|} t_i \\ c_{ik} &\geq x_{ijk} \cdot t_j \quad \forall k, \{i, j\} \leq k \\ u_{ik} &\geq 0 \quad \forall k, i \leq k \\ u_{ik} - t_{max} \cdot y_{ik} &\leq 0 \quad \forall k, i \leq k \\ u_{ik} - c_{ik} &\leq 0 \quad \forall k, i \leq k \\ c_{ik} - u_{ik} + t_{max} \cdot y_{ik} &\leq t_{max} \quad \forall k, i \leq k \\ \sum_{j=1}^k x_{ijk} \cdot w_j &\leq W_{max} \quad \forall k, i \leq k \\ x_{iik} &= 1 \quad \forall k, i \leq k \\ x_{ijk} &= x_{jik} \quad \forall k, \{i, j\} \leq k \\ 1 - x_{iqk} &\geq x_{iqk} - x_{jqk} \geq x_{ijk} - 1 \quad \forall k, \{i, j, q\} \leq k, i \neq j \neq q \\ d_i &\leq TSVmax_i \quad \forall i \neq 1 \\ d_i &\geq \sum_{j=i}^k w_j \cdot x_{ijk} \quad \forall i \neq 1, k \geq i \\ d_i &\geq w_j \quad \forall i, j = i..|M| \\ y_{1k} &= 1 \quad \forall k \\ y_{ik} &\geq \frac{1}{1-i} \sum_{j=1}^{i-1} (x_{ijk} - 1) - C \quad \forall k, i \leq k, i \neq 1 \end{aligned}$$

Figure 11: ILP model for the 3D TAM optimization Problem P_{MTS}^H .