Test-Access Mechanism Optimization for Core-Based Three-Dimensional SOCs

Xiaoxia Wu[1] Yibo Chen[1] Krishnendu Chakrabarty[2] Yuan Xie[1]
[1] Computer Science and Engineering Department, The Pennsylvania State University, University Park, PA 16802
[2] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708
[1] Email: {xwu,yxc236,yuanxie}@cse.psu.edu [2] Email: krish@ee.duke.edu

***Abstract*— Test-access mechanisms (TAMs) and test wrappers (e.g., the IEEE Standard 1500 wrapper) facilitate the modular testing of embedded cores in a core-based system-on-chip (SOC). Such a modular testing approach can also be used for emerging three-dimensional integrated circuits based on through-silicon vias (TSVs). Core-based SOCs based on 3D IC technology are being advocated as a means to continue technology scaling and overcome interconnect-related bottlenecks. We present an optimization technique for minimizing the test time for 3D core-based SOCs under constraints on the number of TSVs and the TAM bitwidth. The proposed optimization method is based on a combination of integer linear programming, LP-relaxation, and randomized rounding. Simulation results are presented for the ITC 02 SOC Test Benchmarks and the test times are compared to that obtained when methods developed earlier for two-dimensional ICs are applied to 3D ICs.**

I. Introduction

Embedded cores are now commonplace in large system-on-a-chip (SOC) designs. However, since embedded cores are not directly accessible via chip inputs and outputs, special access mechanisms are required to test them at the system level. A test-access architecture, also referred to as a test-access mechanism (TAM), provides the means for on-chip test data transport. Test wrappers form the interface between the embedded core and its environment, and connect the terminals of the embedded core to other cores in the SOC and to the TAM [1].

Modular testing allows embedded cores in SOCs to be tested as black boxes, and it facilitates the reuse of test patterns. Therefore, it is an attractive test solution for core-based SOCs. Test-wrapper design and TAM optimization are vital for modular testing, because they directly affect SOC testing time, and therefore the test cost. The recent IEEE 1500 standard addresses wrapper design, but it leaves TAM optimization to the system integrator. Therefore, TAM optimization and test scheduling at the SOC level has been an active area of research in recent years [2]–[5]. Prior work has been limited to traditional two-dimensional (2D) integrated circuits (ICs), which is the widely used technology today for IC manufacturing. However, with the emergence of three-dimensional (3D) ICs and the increasing likelihood of core-based design being the design style of choice for 3D ICs [6], [7], there is a need to develop modular testing methods for 3D ICs.

This work was supported in part by NSF CCF 0702617 and a grant from DARPA/IBM 3D program.

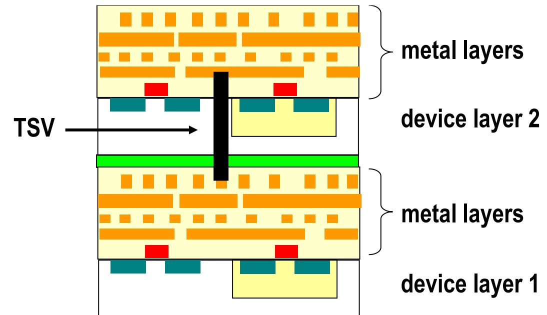

Fig. 1: A conceptual view of a 3D IC chip, with a through-silicon-via (TSV) used as interconnect between two dies or wafers.

With continued technology scaling, interconnect has emerged as the dominant source of circuit delay and power consumption. The reduction of interconnect delays and power consumption are of paramount importance for deep-submicron designs. 3D ICs have recently emerged as a promising means to mitigate these interconnect-related problems [8], [9]. Several 3D integration technologies have been explored recently, including wire bonded, microbump, contactless (capacitive or inductive), and through-silicon-via (TSV) vertical interconnects [8]. TSV 3D integration has the potential to offer the greatest vertical interconnect density, and therefore is the most promising one among all the vertical interconnect technologies. In 3D ICs that are based on TSV technology, multiple active device layers are stacked together (through wafer stacking or die stacking) with direct vertical TSV interconnects [9]. Figure 1 shows the conceptual view of a 3D IC using through-silicon-via interconnects.

3D ICs offer a number of advantages over traditional two-dimensional (2D) design [9]: (1) Higher performance because of the reduction of average interconnect length, as well as bandwidth improvement due to die stacking; (2) Lower interconnect power consumption due to wiring length reduction (reduced capacitance); (3) Higher packing density and smaller footprint; (4) Support for the implementation of mixed-technology chips: each die can have different technologies.

The fabrication of 3D ICs is now viable. For example, in early 2007, IBM announced breakthroughs that enable the transition from horizontal 2-D chip layouts to 3-D chip stacking [10]. Even though 3D manufacturing is becoming feasible, 3D IC design will not be commercially viable without the support of relevant 3D design-automation tools, which are needed to allow IC designers to efficiently exploit the benefits of 3D technologies. Design and test-automation tools are also needed to carry out appropriate tradeoffs because the number of available TSVs, which directly affects the chip area, is limited.

In this paper, we address TAM optimization for 3D SOCs using the wrapper design method from [2]. While there are obvious similarities between TAM optimization for 2D ICs and that for 3D ICs, non-trivial constraints related to the number of TSVs, placement of cores on different layers, and thermal limits must also be taken into account for 3D integration. In 3D SOCs, the various cores are placed on different layers; this approach reduces interconnect length, decreases chip footprint, and allows more efficient mixed-technology integration. However, all the chip pins are at the lowest layer [9], and due to the limited number of TSVs, the availability of bitwidth for test-access to cores on higher layers is adversely affected.

The rest of the paper is organized as follows: Section II uses a simple example to illustrate the problem investigated in this paper. Section III presents related prior work on 3D ICs and TAM optimization for 2D ICs. Section IV describes integer linear programming models, as well as a heuristic method based on LP-relaxation and randomized rounding. Section V presents experimental results on benchmark SOCs. Finally, Section VI presents conclusions and outlines directions for future work.

II. Problem formulation

A. Motivating example

Figure 2 shows an example to illustrate the motivation of our research. A total of five (wrapped) cores are shown in a 3D IC. Four cores are placed in Layer 0, while the fifth core is placed on Layer 1. Suppose that a total of $2W$ channels (TAM wires) are available from the tester to access the cores, out of which W channels must be used for transporting test stimuli and the remaining W channels must be used for collecting the test responses. We also have an upper limit on the number of TSVs that can be used for the test-access infrastructure. The TAM design needs to be optimized to minimize the test application time. Therefore, the objective here is to allocate wires (bitwidths) to different TAMs used to access the cores. Figure 2 shows two possible TAM designs for the same 3D SOC design:

- **Approach 1:** In Figure 2(a), we have two TAMs of width $w1$ and $w2$, respectively, $(w1 + w2 = W)$ and the core on Layer 1 is accessed using the first TAM. A total of $2{\cdot}w1$ TSVs are needed for this test-infrastructure design.

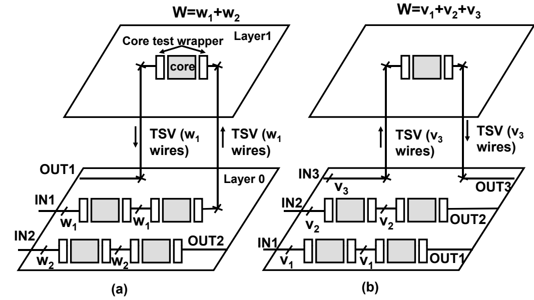

Fig. 2: Two solutions for TAM optimization in a 3D IC.

- **Approach 2:** Figure 2(b) presents an alternative design in which three TAMs are used and the core on Layer 1 is accessed using the third TAM $(v1 + v2 + v3 = W)$.

Note that both $2 \cdot w1$ (in Figure 2(a)) and $2 \cdot v3$ (in Figure 2(b)) are no more than the number of TSVs dedicated for test access. In *Approach 1*, there are only two TAMs and the width of each TAM could be wider, but *TAM1* has to sequentially test 3 cores; In *Approach 2*, there are three TAMs and the width of each TAM might be smaller, but each TAM connects a maximum of two cores. It is not obvious to the designers which approach is better in terms of test time. Therefore, optimization methods and design tools are needed to determine a test-access architecture (e.g., number of TAMs, width of the TAMs, assignment of cores to TAMs, etc.) that leads to the minimum test time under constraints on the number of TSVs.

B. Problem formulation

The general problem of 3D SOC test planning includes the design/optimization of a test-access architecture, optimization of core wrappers, and test scheduling. The goal is to minimize the testing time under limits on the number of TSVs and the SOC-level TAM width. Additional constraints are also imposed by thermal considerations that are of paramount importance in 3D integration. The TAM optimization problem $3DP_{PAW}$ that we address in this paper is as follows. Given the test set parameters (number of test patterns, number of I/Os, and scan chains, and scan-chain lengths) for the embedded cores in the SOC, the total TAM width, 3D-technology constraints (maximum number of TSVs, thermal limits, etc.), and the 3D placement for each core, determine the partition of the total TAM width among the TAM partitions, an optimal assignment of cores to each TAM partition, and an optimal wrapper design for each core, such that the overall SOC testing time is minimized.

In order to solve $3DP_{PAW}$, we first examine two simpler problems, along the same lines as the general TAM optimization problem was decomposed into a progression of simpler problems in [2]. The first problem $3DP_W$ addresses test-wrapper design, which is identical to 2D wrapper design [2] since we assume that any given core is placed on only one layer (i.e., a core is not partitioned across multiple layers). The second problem $3DP_{AW}$ addresses

3D TAM optimization with a predefined TAM width for each partition. The third (general) problem $3DP_{PAW}$ also addresses appropriate TAM partitioning. We formulate the relevant optimization problems and develop integer linear programming (ILP) models for both $3DP_{AW}$ and $3DP_{PAW}$. These two optimization problems have been shown to be NP-hard in [2]. Therefore, we also develop a heuristic technique based on a combination of ILP modeling, LP-relaxation, and randomized rounding [11] to efficiently obtain near-optimal results. To the best of our knowledge, this is the first paper to address TAM design in the 3D design space.

III. Related Prior Work

TAM design methods proposed in the literature include multiplexed access, partial isolation rings, core transparency, dedicated test bus, etc. Bus-based TAMs, being flexible and scalable, appear to be the most promising. TAM optimization and test scheduling problems have been tackled using ILP [2], rectangle packing [12], [13], and iterative refinement. However, all these methods are targeted at traditional 2D technologies for ICs.

3D technologies have attracted considerable attention in the past few years. Research has been focused on 3D fabrication, 3D electronic design automation (EDA), and 3D system architectures [8], [9], [14], [15]. Among all EDA challenges for 3D IC design, tools and methodologies for 3D IC testing are regarded as the "No.1 challenge", according to a recent keynote speech [16] by Ted Vucurevich (CTO of Cadence Design System). However, research on testing for 3D ICs is still in its infancy. It is only recently that progress has been reported on the testing of 3D ICs. For example, Lewis et al. have proposed a scan-island based pre-bond test method to facilitate the testability of die-stacked microprocessors [17]. Wu et al. have proposed several 3D scan-chain design techniques [18].

IV. Optimization of 3D TAMs

In this paper, we assume the "test bus" model for TAM design. We assume that each of the B TAMs on the 3D SOC are independent; however, the cores on each TAM are tested in sequential order. We first review the wrapper-design problem $3DP_W$ from [19] and [2]. Next, we describe ILP models to solve the $3DP_{AW}$ and $3DP_{PAW}$ problems, and show how randomized rounding can be used to solve these problems efficiently.

A. Test-Wrapper Design

A test wrapper is a layer of design-for-test logic that connects a TAM to a core for the purpose of testing [1]. An embedded core typically contains multiple functional I/Os as well as several internal scan chains. To perform test-width adaptation, wrapper scan chains are constructed by connecting core I/Os with internal scan chains. The number of wrapper scan chains thus constructed is equal to the TAM width provided to the core; hence each wrapper scan chain is assigned to a unique TAM line. Thus the test-data width (number of core terminals) of the core is adapted to the TAM width. Balanced wrapper scan chains are those that are as equal in length to each other as possible. Balanced wrapper scan chains are important because the number of clock cycles to scan in (out) a test pattern to (from) a core is a function of the length of the longest wrapper scan-in (scan-out) chain. Let si (so) be the length of the longest wrapper scan-in (scan-out) chain for a core. The time required to apply the entire test set to the core is then given by $T = (1 + max(si, so)) \cdot p_i + min(si, so)$, where p is the number of test patterns [2].

The test-wrapper design problem P_W is defined as follows [2], [19]: Given a core with n functional inputs, m functional outputs, sc internal scan chains of lengths $l_1, l_2, ..., l_{sc}$, respectively, and TAM width k, assign the $n+m+sc$ wrapper scan chain elements to $k_1 \leq k$ wrapper scan chains such that (i) $\max\{si, so\}$ is minimized, and (ii) k_1 is minimum, subject to priority (i).

We adopt the approximation algorithm based on the Best Fit Decreasing (BFD) heuristic [2] [20] to solve P_W efficiently. The algorithm has three parts: (i) partition the internal scan chains among a minimum number of wrapper scan chains to minimize the longest wrapper scan chain length, (ii) assign the functional inputs to the wrapper scan chains created in part (i), and (iii) assign the functional outputs to the wrapper scan chains created in part (i).

B. ILP model for $3DP_{AW}$

We next present the ILP model for solving $3DP_{AW}$. This model serves as the basis for the ILP model for the more general problem $3DP_{PAW}$. The goal in ILP is to minimize a linear objective function on a set of integer variables, while satisfying a set of linear constraints [21]. A typical ILP model can be described as follows:

Minimize **Cx** subject to **Ax** $\leq$ **B**, where **x** ≥ 0. In this model, **x** represents a vector of variables, **C** is a cost vector, **A** is a constraint matrix, and **B** refers to a vector of constants.

The $3DP_{AW}$ problem is formally defined as follows: Given N cores that have placed in a 3D floorplan and B TAM partitions of bitwidths w_1, w_2, ..., w_B, respectively, and the maximum number of TSVs available for the TAM, determine an assignment of cores to TAM partitions and a wrapper design for each core, such that the testing time is minimized.

Let $T_i(w_j)$ be the number of clock cycles needed to test Core i if it is assigned to TAM partition j. The testing time is calculated as follows [2]: $T_i(w_j) = (1 + \max(s_i, s_o)) \cdot p_i + \min(s_i, s_o)$ where p_i is the number of test patterns for core i (which is known) and $s_i(s_o)$ is the length of the longest wrapper scan-in(scan-out) chain obtained from design wrapper. As in [2], we define the binary variable x_{ij} as follows:

$$x_{ij} = \begin{cases} 1 & \text{if core } i \text{ is assigned to TAM } j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The time needed to test all cores on TAM partition j is given by $\sum_{i=1}^{N} T_i(w_j) \cdot x_{ij}$. Since test data can be transferred

on the different TAM partitions in parallel, the SOC testing time is simply $\max_{1\le j\le B}\sum_{i=1}^{N}(T_i(w_j)\cdot x_{ij})$.

Next we introduce constraints that are imposed by 3D IC technology. Even though the various cores can be placed on different layers, all the I/O pins in a 3D IC are only at the bottom layer, i.e., Layer 0 [9]. We therefore assume that all the I/Os for the TAM are also in Layer 0. An integer L_i is introduced to represent the layer assignment for Core i, e.g., $L_1 = 2$ means that Core 1 is assigned to Layer 2. We also add a binary variable z_{ijk}, which is 1 only if both Core i and Core j are in TAM partition k. We use this variables to calculate the number of TSVs that are needed for 3D TAM design. It can be easily seen that $z_{ijk} = x_{ik}\cdot x_{jk}$. The nonlinear equation can be linearized by treating z_{ijk} as an independent variable and adding two new constraints:

1) $x_{ik} + x_{jk} - z_{ijk} \le 1$
2) $x_{ik} + x_{jk} - 2\cdot z_{ijk} \ge 0$.

We next calculate the minimum number of TSVs needed to implement TAM partition k.

Theorem 1: The minimum number of TSVs, v_k, needed to implement TAM partition k of width w_k is given by:

$$v_k = 2\cdot w_k\cdot \max_{1\le i,j\le N}\{z_{ijk}\cdot(\max_{1\le i,j\le N}\{(L_i - L_j), L_i, L_j\})\}$$

Proof: We prove the theorem using the principle of mathematical induction. Let $L_{ij} = \max_{1\le i,j\le N}((L_i - L_j), L_i, L_j)$. We therefore have $v_k = 2\cdot w_k\cdot \max_{1\le i,j\le N}(z_{ijk}\cdot L_{ij}))$.

Induction Basis: Suppose all cores assigned to TAM partition j are on Layer 0. No TSVs are needed for this case since for all i and j, $L_{ij} = 0$, so we get $v_k = 0$. Next suppose that all the cores assigned to TAM partition j are on Layer 1. Now we have $L_{ij} = 1$ for all i and j, and therefore $v_k = 2\cdot w_k\cdot 1 = 2\cdot w_k$, which is valid since TAM partition k is w_k bit wide and access to Layer 1 needs $2\cdot w_k$ bits (w_k bits for input and w_k bits for output). Finally, suppose some cores assigned to TAM partition k are on Layer 0 while other are on Layer 1. There exists a pair i, j such that $z_{ijk} = 1$, therefore $L_{ij} = 1$ and $v_k = 2\cdot w_k$.

Induction Hypothesis: Suppose the equation holds when there are m layers (numbered 0 to $m-1$).

Inductive Step: We have to consider two cases:

- Case 1: If no core on Layer m is connected to TAM partition j, it is obvious that the equation for v_k holds.
- Case 2: There is at least one core on Layer m that is connected to TAM partition k. The number of TSVs is given by:

$$v_k = 2\cdot w_k\cdot \max_{1\le i,j\le N}\{z_{ijk}\cdot L_{ij})\} + 2\cdot w_k \quad (2)$$
$$= 2\cdot w_k\cdot \max_{1\le i,j\le N}\{z_{ijk}\cdot (L_{ij}+1))\} \quad (3)$$
$$= 2\cdot w_k\cdot \max_{1\le i,j\le N}(z_{ijk}\cdot \tilde{L}_{ij})) \quad (4)$$

where $\tilde{L}_{ij} = L_{ij}+1 = \max_{1\le i,j\le N}\{(\tilde{L}_i - \tilde{L}_j), \tilde{L}_i, \tilde{L}_j\}$. The parameters $\tilde{L}_i$ and $\tilde{L}_j$ denote the fact that we are moving from m layers to $m+1$ layers.

Therefore, the number of TSVs for chain k can be represented by:

$$v_k = 2\cdot w_k\cdot \max_{1\le i,j\le N}\{z_{ijk}\cdot(\max_{1\le i,j\le N}((L_i - L_j), L_i, L_j))\}$$

□

The total number of TSVs for the TAM architecture is given by $\sum_{k=1}^{B} v(k)$, which must be smaller than a predefined limit. We can also add an upper limit on the number of TSVs limit for each TAM partition. The complete ILP model for $3DP_{AW}$ is shown in Figure 3. The number of variables for this model is $B + N\cdot B + N^2\cdot B = O(N^2\cdot B)$ and the number of constraints is $N + 2\cdot B + 2\cdot N^2\cdot B = O(N^2 B)$.

Objective:

Minimize $\max_{1\le j\le B}\sum_{i=1}^{N} T_i(w_j)\cdot x_{ij}$

Subject to:

$\sum_{j=1}^{B} x_{ij} = 1 (1\le i\le N)$

$x_{ik} + x_{jk} - z_{ijk} \le 1$

$x_{ik} + x_{jk} - 2\cdot z_{ijk} \ge 0$

$L_{ij} = \max_{1\le i,j\le N}((L_i - L_j), L_i, L_j)$

$v_k = 2\cdot w_k\cdot \max_{1\le i,j\le N}(z_{ijk}\cdot L_{ij}))$

$\sum_{k=1}^{B} v(k) \le V_{total}$

$\forall k \quad v(k) \le V_{single}$

Fig. 3: The ILP model for $3DP_{AW}$.

C. ILP model for $3DP_{PAW}$

In the $3DP_{AW}$ problem, the TAM width for each TAM chain is predefined. However, appropriate sizing of each TAM partition results in lower test time for an SOC [2]. Therefore, we next address the more general problem $3DP_{PAW}$, as defined in Section II.

From Theorem 1 in [2], the width of each TAM partition does not need to exceed an upper bound k_{max} for any core in the SOC. We denote this upper bound on the width of the individual TAM partitions as w_{max}. If the width of a TAM partition is greater than w_{max}, there is no further decrease in testing time. The objective of $3DP_{PAW}$ problem is therefore: Minimize $\max_j \sum_{i=1}^{N} T_i(w_j)\cdot x_{ij}$.

Since both w_j and x_{ij} are variables, the objective function needs to be linearized. We add a new binary variable d_{jk}, $1\le j\le B, 1\le k\le w_{max})$, defined as:

$$d_{ij} = \begin{cases} 1 & \text{if TAM partition } j \text{ is } k \text{ bits wide} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Therefore, $T_i(w_j)$ is described as: $T_i(w_j) = \sum_{k=1}^{w_{max}} d_{jk}\cdot T_i(k)$. In addition, two other constraints are included:

1) $w_j = \sum_{k=1}^{w_{max}} k\cdot d_{jk}$, a TAM partition can have width between 1 and w_{max})
2) $\sum_{k=1}^{w_{max}} d_{jk} = 1$, i.e., the width of a TAM partition must be unique.

The objective function can therefore be written as:

$$\max_j \sum_{i=1}^{N} \sum_{k=1}^{w_{max}} d_{jk} \cdot T_i(k) \cdot x_{ij} = 1$$

The testing time $T_i(k)$ is obtained from the Design Wrapper procedure [2] and stored in a look-up table for use in ILP models. Next, we linearize the non-linear term $d_{jk}x_{ij}$ by introducing a new binary variable y_{ijk} and two constraints:

1) $x_{ij} + d_{jk} - y_{ijk} \leq 1$
2) $x_{ij} + d_{jk} - 2 \cdot y_{ijk} \geq 0.$

The number of TSVs for chain k is as follows:

$$v_k = 2 \cdot w_k \cdot \max_{1 \leq i,j \leq N} \{y_{ijk} \cdot (\max_{1 \leq i,j \leq N}((L_i - L_j), L_i, L_j))\}$$

Since w_k is also a variable and it is given by $w_k = \sum_{m=1}^{w_{max}} m \cdot d_{km}$, a new binary variable n_{ijkm} is introduced to replace $d_{km} \cdot y_{ijk}$. Two more constraints are added:

1) $d_{km} + z_{ijk} - n_{ijkm} \leq 1$
2) $d_{km} + z_{ijk} - 2 \cdot n_{ijkm} \geq 0.$

Now the number of TSVs becomes

$$v_k = \sum_{m=1}^{w_{max}} 2 \cdot m \cdot \max_{1 \leq i,j \leq N, 1 \leq k \leq B} \{n_{ijkm} \cdot L_{ij})\}$$

where L_{ij} is defined as $L_{ij} = \max_{1 \leq i,j \leq N}\{(L_i - L_j), L_i, L_j\}$. The complete ILP model for the $3DP_{PAW}$ problem is shown in Figure 4. The number of variables for this model is $B + B \cdot W + N \cdot B + N^2 \cdot B + N^2 \cdot B \cdot W = O(N^2 \cdot B \cdot W)$ and the number of constraints is $1 + 4 \cdot B + N + 2 \cdot N^2 \cdot B + 2 \cdot N^2 \cdot B \cdot W = O(N^2 BW)$.

Objective:

Minimize $\max_j \sum_{i=1}^{N} \sum_{k=1}^{w_{max}} y_{ijk} \cdot T_i(k)$

Subject to:

$\sum_{j=1}^{B} x_{ij} = 1 (1 \leq i \leq N)$

$x_{ij} + d_{jk} - y_{ijk} \leq 1$

$x_{ij} + d_{jk} - 2 \cdot y_{ijk} \geq 0$

$w_j = \sum_{k=1}^{w_{max}} k \cdot d_{jk}$

$\sum_{k=1}^{w_{max}} d_{jk} = 1$

$\sum_{j=1}^{B} w_j \leq W$

$d_{km} + z_{ijk} - n_{ijkm} \leq 1$

$d_{km} + z_{ijk} - 2 \cdot n_{ijkm} \geq 0$

$L_{ij} = \max_{1 \leq i,j \leq N}\{(L_i - L_j), L_i, L_j\}$

$v_k = \sum_{m=1}^{w_{max}} 2 \cdot m \cdot \max_{1 \leq i,j \leq N, 1 \leq k \leq B}\{n_{ijkm} \cdot L_{ij})\}$

$\sum_{k=1}^{B} v(k) \leq V_{total}$

$\forall k \quad v(k) \leq V_{single}$

Fig. 4: The ILP model for $3DP_{PAW}$

D. Randomized rounding

While the ILP models discussed in this section can be used to optimally solve 3D TAM optimization problems, they do not scale well for large SOC designs. ILP problems are known to be NP-hard [20]; however, linear programming (LP) problems can be solved optimally in polynomial time [21]. Therefore, we adopt the method of LP-relaxation and combine it with randomized rounding [11]. In LP-relaxation, the binary variables are relaxed to real-valued variables such that the solution to the relaxed LP problem provides a lower bound for the cost function (test time in this case). However, the fractional values obtained for the x_{ij} variables are inadmissible in practice; these variables must be mapped to either 0 or 1. For this purpose, we use the method of randomized rounding.

The randomized rounding technique for ILP problems consists of three steps. The first step is to solve the corresponding LP problem, fixing all x_{ij} variables that are assigned to 1. The second step is to randomly pick a variable from the set of variables with fractional values and assign it to 1 with a probability equal to the fractional value. For example, if the solution to the LP problem assigns the value 0.4 to a variable, we generate a random number between 0 and 1. If this random number is less than or equal to 0.4, the variable is set to 1. Otherwise it is set to be 0. In the third step, the LP problem is solved again, and the randomized rounding step is repeated until all variables are set to either 0 or 1.

Note that in the Step 2 of the randomized rounding technique, the violation of constraints must be prevented in order to ensure that the LP problem is feasible. For example, in the $3DP_{AW}$ problem, we randomly pick a variable x_{ij} with a fractional value, and assign the value 1 to it according to the random number that we generate. Before the assignment, we check if x_{kj} $(i \neq k)$ is already assigned to 1. If this is the case, we can only assign 0 to x_{ij}, otherwise the first constraint is violated, and the LP problem becomes infeasible. Therefore, we need to guarantee that there is no constraint violation with existing fixed-variable values.

V. Experiments and Results

A. SOC test benchmarks and experimental setup

To illustrate the proposed 3D TAM optimization method and to demonstrate its effectiveness, we use four representative SOCs from the ITC'02 SOC test benchmarks, namely d695, p22810, p34392, and p93791. We use Xpress-MP [22], a commercial ILP solver, to solve the ILP model with randomized rounding for the $3DP_{PAW}$ problem.

B. Thermal-aware 3D SOC floorplanning

One of the major concerns in 3D SOC design is the potential increase in chip temperature. The stacking of multiple active layers can lead to higher power densities, and the on-chip temperature depends on the power density [8], [15], [23]. Therefore, it is essential to have a thermal-aware floorplanner for 3D SOC design. In this paper, we adopt the thermal-aware floorplanner from [23] for the placement of the 3D SOC benchmarks. This 3D thermal-aware floorplanner is based on a simulated annealing algorithm. The inputs to the floorplanning algorithm are the area and the functional power of all functional modules in a 3D SOC. Since there is no detailed area information for the ITC'02 SOC test benchmarks, we estimate the area of each module based on the number of flip-flops. Since only test power for each module is available [24], we scale down the test power to

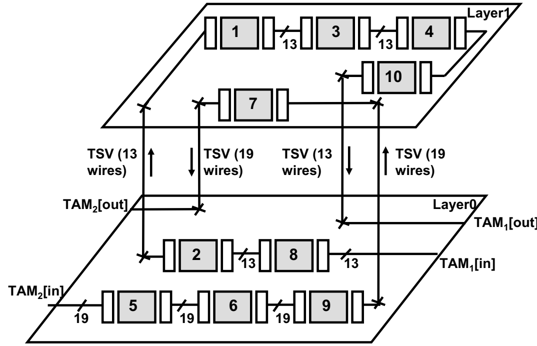

Fig. 5: TAM design obtained for d695 ($B = 2, W = 32$, two layers).

approximate the functional power for 3D SOC floorplanning. After we feed the area and power values into the placement algorithm, it outputs the thermal-aware floorplan, which is used as an input to our 3D TAM optimization models.

C. Results for $3DP_{PAW}$ problem

Figure 5 shows a TAM design obtained for SOC d695 ($B = 2, W = 32$, 2 layers). The upper limit on the number of TSVs is set to 60/80, where the maximum number of TSVs for each TAM partition is 60 and the maximum number of TSVs for the SOC is 80. Cores 2, 5, 6, 8, and 9 are placed in Layer 0 and Cores 1, 3, 4, 7, and 10 are placed in Layer 1. The first partition of width 19 bits is connected to Cores 1, 2, 3, 4, 8, 10. The second TAM partition of width 13 bits is used to access Core 5, 6, 7, 9. This TAM partition results in minimum test time under constraints of TSV limit and the thermal-aware placement.

Table I shows the testing time and the TAM partitions for SOC d695 for $B = 2, 3, 4$ and for different values of W. The upper limit on the number of TSVs is set to 48/60 for each case, where the maximum number of TSVs for each TAM partition is 48 and the maximum number of TSVs for the SOC is 60. The first two columns list the total TAM width W and the number of TAM chains B, respectively. The third column shows the TAM partition for different values of W and B. Column 4 shows the baseline 2D test time in clock cycles (#CC), where we apply the TAM optimization method (based on LP-relaxation and randomized rounding) separately to each layer. The TAM width limit for each layer is set as follows: if $W > TSV/2$, for Layer 0, the width limit is set to W and for Layers 1, 2, ..., the TAM width limit is set to $TSV/2$. If $W \leq TSV/2$, for each layer the TAM width limit is W. After computing the test time for each layer, we obtain the total test time by summing up the test times for all the layers. Other "baseline methods" that attempt to parallelize the testing of multiple layers are implicitly considered in the proposed optimization method.

The test time (in clock cycles, #CC) for the proposed 3D method is shown in Column 5. Column 6 lists the test time results obtained from LP relaxation before randomized rounding, which provides a lower bound on the test time. Column 7 lists the run time of LP-relaxation combined with randomized rounding for this SOC. Table II, Table III, and Table IV present similar results for SOCs p22810, p34392, and p93791 with $B = 2, 3, 4$ (three layers), $B = 2, 3, 4$ (three layers), and $B = 2, 3, 4, 5$ (four layers), respectively. The TSV limit for the three larger SOCs are set to 60/80.

TABLE I: Results on TAM optimization for SOC d695 for $B = 2, 3, 4$ (two layers, TSV limits 48/60).

W	B	3D TAM partition	Test time 2D (#CC)	Test time 3D (#CC)	Lower bound (#CC)	Run Time for 3D (seconds)
16	2	(7,9)	44225	24257	20355	2.70
	3	(4,5,7)	46108	22984	19714	8.67
	4	(1,1,4,10)	51707	27403	23380	21.3
24	2	(5,19)	31070	21953	18868	2.70
	3	(1,4,19)	31077	20764	16221	8.77
	4	(1,3,4,16)	41611	16549	13984	30.36
32	2	(11,21)	28524	20934	17101	2.43
	3	(4,10,18)	23944	16085	13798	8.97
	4	(2,4,10,16)	33782	13505	10334	63.58
40	2	(19,21)	25761	18604	15364	2.59
	3	(4,17,19)	21070	13095	11720	9.35
	4	(4,4,13,19)	26961	11282	8437	113.56
48	2	(21,27)	25761	17331	14232	3.82
	3	(7,19,22)	21070	11808	9375	9.99
	4	(3,4,7,34)	25086	9522	6775	58.54
56	2	(23,33)	23256	15392	13768	3.67
	3	(4,19,33)	21070	11808	8112	10.45
	4	(2,4,16,34)	24852	8452	6177	69.97
64	2	(23,41)	23232	13984	12153	3.71
	3	(5,22,37)	21070	9731	6860	12.47
	4	(13,16,16,19)	24852	7923	5752	137.92

We draw several conclusions from the results in Tables I-IV. First, the test time is considerably higher if we simply use 2D TAM optimization on a layer-by-layer basis. Therefore, 3D TAM optimization is especially important for 3D SOCs in order to reduce test cost. Second, as the total TAM width W is increased, the testing time decreases, which is consistent with the results obtained using earlier methods for 2D ICs [2]. The third observation is that the run time becomes larger when the total TAM width increases because there are more variables and constraints in the LP models. However, with a combination of LP-relaxation and randomized rounding techniques, results can be obtained much more efficiently compared to previous 2D counterparts [2].

VI. Conclusions

We have shown that a modular testing approach can be used for emerging three-dimensional integrated circuits based on through-silicon vias (TSVs). We have presented an optimization technique for minimizing the test time for 3D core-based SOCs under constraints on the number of TSVs and the TAM width. The proposed optimization method is based on a combination of integer linear programming, LP-relaxation, and randomized rounding. We have carried out a series of simulations for four ITC'02 SOC test benchmarks by considering thermal-aware placement on a 3D substrate. Simulation results show that the proposed method leads to lower test times compared to a baseline method that applies 2D TAM optimization to each layer of the 3D SOC. As part of ongoing work, we are studying the use of bandwidth matching techniques to interface high-speed narrow TSVs to wider, but slower TAMs on the different layers.

TABLE II: Results on TAM optimization for SOC p22810 for $B = 2, 3, 4$ (three layers, TSV limits 60/80).

W	B	3D TAM partition	Test time 2D (#CC)	Test time 3D (#CC)	Lower bound (#CC)	Run Time for 3D (seconds)
16	2	(5,11)	334276	289986	265329	20.24
	3	(4,4,8)	323526	275537	257791	222.24
	4	(1,4,4,7)	421172	286734	286146	96.50
24	2	(11,13)	326153	263892	236842	30.89
	3	(1,10,13)	299590	254732	220664	216.21
	4	(1,1,10,12)	352227	249274	233497	120.36
32	2	(7,25)	314522	230925	223984	48.53
	3	(10,4,18)	289293	232810	181696	216.11
	3	(1,10,10,11)	348318	205831	166188	546.28
40	2	(13,27)	310323	215506	203672	32.59
	3	(10,13,17)	287143	201221	158628	1247.43
	4	(7,10,4,19)	348318	181687	144182	1560.73
48	2	(13,35)	306389	205782	192256	53.82
	3	(7,13,28)	283272	172191	133103	292.11
	4	(10,10,7,21)	289710	169392	145543	827.23
56	2	(15,41)	304831	204155	179474	122.21
	3	(10,13,33)	281965	153282	123137	576.54
	4	(4,10,16,26)	285825	162818	141406	623.21
64	2	(15,49)	304810	179688	170123	33.71
	3	(13,19,32)	279400	142210	121908	1279.17
	4	(10,13,19,22))	285825	157460	138737	7838.44

TABLE III: Results on TAM optimization for SOC p34392 for $B = 2, 3, 4$ (three layers, TSV limits 55/70).

W	B	3D TAM partition	Test time 2D (#CC)	Test time 3D (#CC)	Lower bound (#CC)	Run Time for 3D (seconds)
16	2	(7,9)	1338043	1134919	1113301	12.20
	3	(1,7,8)	1365702	999543	999201	55.97
	4	(1,4,4,7)	2025402	1288341	1174558	28.35
24	2	(9,15)	1202389	913550	859758	6.35
	3	(7,7,10)	1055242	762841	762669	50.12
	4	(1,1,10,12)	1377077	798449	743508	1112.65
32	2	(11,21)	1183301	843301	779966	7.26
	3	(10,7,15)	1045242	665445	592325	18.59
	3	(4,7,10,11)	1359846	684524	563604	554.70
40	2	(13,27)	1047913	752782	716114	12.59
	3	(13,7,20)	982127	552231	506506	47.64
	4	(4,1,16,19)	1026819	584301	497810	600.22
48	2	(13,35)	1026760	693465	671796	23.82
	3	(7,16,25)	982127	546152	490218	87.66
	4	(1,7,19,21)	995186	544579	474016	812.23
56	2	(13,43)	1014317	637606	621753	122.21
	3	(16,7,33)	922127	540069	485039	215.41
	4	(1,7,19,29)	972797	544579	455916	400.40
64	2	(13,51)	987564	559758	539403	33.71
	3	(16,7,41)	932476	534212	481386	1026.22
	4	(10,7,19,28)	948945	544330	429201	1145.44

TABLE IV: Results on TAM optimization for SOC p93791 for $B = 2, 3, 4, 5$ (four layers, TSV limits 60/80).

W	B	3D TAM partition	Test time 2D (#CC)	Test time 3D (#CC)	Lower bound (#CC)	Run Time for 3D (seconds)
16	2	(7,9)	1869200	1800413	1712598	15.70
	3	(4,4,8)	2021836	1779401	1779298	90.62
	4	(1,4,4,7)	3651006	2179527	2177620	68.3
	5	(1,1,4,1,9)	4379506	2030868	1913331	89.2
24	2	(7,17)	1294512	1259711	1169532	18.70
	3	(4,4,16)	1316252	1197683	1108672	100.71
	4	(1,7,7,9)	1709129	1337682	1303728	378.67
	5	(1,1,7,7,8)	2115963	1263801	1152322	513.39
32	2	(9,23)	966752	894463	860629	16.43
	3	(4,4,24)	970585	910957	891713	144.78
	4	(1,10,10,11)	1296534	944807	944706	814.42
	5	(4,4,7,4,13)	1913443	1008684	972991	990.96
40	2	(15,25)	778480	778296	767018	40.59
	3	(7,10,23)	874504	817805	750008	2784.94
	4	(1,4,16,19)	1017802	890145	786207	3780.47
	5	(4,4,10,1,21)	1430320	780202	728064	2448.20
48	2	(15,33)	766270	758806	644230	81.82
	3	(7,16,25)	827277	722042	632076	462.14
	4	(4,13,13,18)	933243	713347	650231	1512.31
	5	(1,10,10,7,20)	1349575	727978	660292	2149.62
56	2	(11,45)	686577	662686	626878	180.67
	3	(13,16,27)	685314	635095	536410	4324.81
	4	(1,7,22,26)	889290	664447	608740	3588.18
	5	(1,1,16,1,37)	1300670	680635	628872	3971.02
64	2	(15,49)	662198	630365	614287	332.71
	3	(16,22,26)	662808	572342	488366	5103.23
	4	(1,13,22,28)	870037	604553	520967	3024.11
	5	(1,3,16,7,37)	1273410	621235	533928	4726.23

REFERENCES

[1] Y. Zorian, E. J. Marinissen, and S. Dey, "Testing embedded-core-based system chips," *Computer*, vol. 32, no. 6, pp. 52–60, 1999.

[2] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-chip," *Journal of Electronic Testing:Theory and Applications*, vol. 18, pp. 213–230, 2002.

[3] Q. Xu and N. Nicolici, "Resource-constrained system-on-a-chip test: a survey," *IEE Proc. Computers and Digital Techniques*, vol. 152, no. 1, pp. 67–81, 2005.

[4] E. Larsson, K. Arvidsson, H. Fujiwara, and Z. Peng, "Efficient test solutions for core-based designs," *TCAD*, vol. 23, no. 5, pp. 758–775, 2004.

[5] T. E. Yu et al., "Using domain partitioning in wrapper design for IP cores under power constraints," in *VTS*, 2007.

[6] K. Banerjee et al., "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 602–633, 2001.

[7] R. Weerasekera et al., "Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs," in *ICCAD*, 2007.

[8] 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.

[9] Y. Xie, G. H. Loh, and K. Bernstein, "Design space exploration for 3D architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, 2006.

[10] "http://www-03.ibm.com/press/us/en/pressrelease/21350.wss."

[11] P. Raghavan and C. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[12] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip," *IEEE Trans. on Computers*, vol. 52, pp. 1619–1632, 2003.

[13] Y. Huang et al., "Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm," in *ITC*, 2002.

[14] R. Reif et al., "Fabrication technologies for three-dimensional integrated circuits," in *ISQED*, 2002.

[15] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *ICCAD*, 2004.

[16] T. Vucurevich, "The Long Road to 3-D Integration: Are We There Yet?" in *Keynote speech at the 3D Architecture Conference*, 2007.

[17] D. L. Lewis and H. S. Lee, "A scan-island based design enabling pre-bond testability in die-stacked microprocessors," in *ITC*, 2007.

[18] X. Wu, P. Falkenstern, and Y. Xie, "Scan chain design for Three-dimensional (3D) ICs," in *ICCD*, 2007.

[19] E. Marinissen, S. Goel, and M. Lousberg, "Wrapper Design for Embedded Core Test," in *ITC*, 2000.

[20] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[21] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1997.

[22] Xpress-MP, "www.dashoptimization.com," 2007.

[23] W. L. Hung et al., "Interconnect and thermal-aware floorplanning for 3D microprocessors," in *ISQED*, 2006.

[24] S. Samii et al., "Cycle-Accurate Test Power Modeling and its Application to SoC Test Scheduling," in *ITC*, 2006.