Microelectronics Journal 41 (2010) 601–615

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Test-access mechanism optimization for core-based three-dimensional SOCs ☆, ☆☆

Xiaoxia Wu [a], Yibo Chen [b], Krishnendu Chakrabarty [c,*], Yuan Xie [b]

[a] Qualcomm, San Diego, CA 92121, USA
[b] Department of Computer Science and Engineering, Penn State University, University Park, PA 16802, USA
[c] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA

ARTICLE INFO

Article history:
Received 22 January 2010
Received in revised form
18 June 2010
Accepted 22 June 2010
Available online 22 July 2010

Keywords:
Three-dimensional integration
Through silicon via
Test access mechanism
Integer linear programming
Randomized rounding

ABSTRACT

Embedded cores in a core-based system-on-chip (SOC) are not easily accessible via chip I/O pins. Test-access mechanisms (TAMs) and test wrappers (e.g., the IEEE Standard 1500 wrapper) have been proposed for the testing of embedded cores in a core-based SOC in a modular fashion. We show that 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 post-bond test time for 3D core-based SOCs under constraints on the number of TSVs, the TAM bitwidth, and thermal limits. The proposed optimization method is based on a combination of integer linear programming, LP-relaxation, and randomized rounding. It considers the Test Bus and TestRail architectures, and incorporates wire-length constraints in test-access optimization. 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. The test time dependence on various 3D parameters (e.g. 3D placement, the number of layers, thermal constraints, and the number of TSVs) is also studied.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Embedded cores are widely used 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 SOC level. A test-access architecture, also referred to as a test-access mechanism (TAM), provides the means for on-chip test data transport. It can be used to transport test stimuli from the test-pattern source to the core-under-test, and to transport test responses from the core-under-test to the test-response sink [1]. Test wrappers are also important components of the test-access infrastructure in a core-based SOC. 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 mechanism [1].

Modular testing of core-based SOCs allows embedded cores in SOCs to be tested as black boxes and it facilitates the reuse of test patterns. Therefore, it is an especially attractive test solution for core-based SOCs. Test-wrapper design and TAM optimization are vital for modular testing because they directly impact 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 [2]. Therefore, TAM optimization and test scheduling at the SOC level has been an active area of research in recent years [3–11]. Prior work has, however, 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 [12], there is a need to develop modular testing methods for 3D ICs. While leveraging today's solutions, these test-access infrastructure methods must also address TSV limits, core placement, and thermal problems as constraints that are unique to 3D integration.

1.1. Background on 3D integration

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. A 3D ICs have recently emerged as a promising means to mitigate these

☆ A preliminary conference version of this paper was published in Proceedings of the IEEE International Conference on Computer Design, 2008.

☆☆ This work was supported in part by NSF 0905365, 0903432, 0702617 and SRC grants.

* Corresponding author.
E-mail address: krish@ee.duke.edu (K. Chakrabarty).

0026-2692/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mejo.2010.06.015

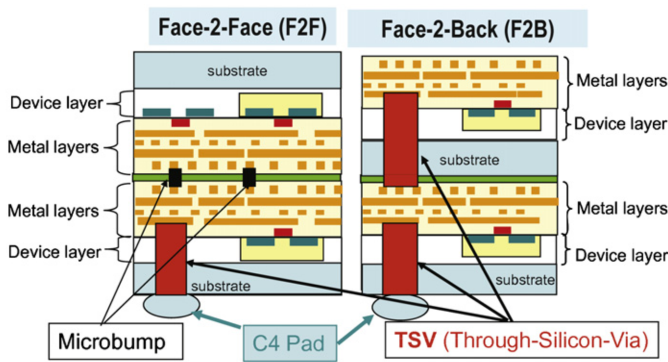

Fig. 1. 3D IC structures with face-to-face and face-to-back bonding.

interconnect-related problems [13–15]. Several 3D integration technologies have been explored recently, including wire bonded, microbump, contactless (capacitive or inductive), and through-silicon-via (TSV) vertical interconnects [16]. 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 [17]. Fig. 1 shows two 3D IC structures with face-to-face and face-to-back bonding.

3D ICs offer a number of advantages over traditional two-dimensional (2D) design [17]:

- Shorter global interconnect because the vertical distance (or the length of TSVs) between two layers is usually in the range of 10–100 μm [17], depending on manufacturing processes.
- Higher performance because of the reduction of average interconnect length, as well as bandwidth improvement due to die stacking.
- Lower interconnect power consumption due to wiring length reduction (reduced capacitance).
- Higher packing density and smaller footprint.
- Support for the implementation of mixed-technology chips: each die can be based on 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. Even though 3D manufacturing is 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.

1.2. Research need and paper contributions

Test techniques and design-for-testability solutions for 3D ICs have remained largely unexplored in the research community, even though experts in industry have identified a number of test challenges related to the lack of probe access for wafers, test access to modules in stacked wafers/dies, thermal concerns, test economics, and new defects arising from unique processing steps such as wafer thinning, alignment, and bonding. Solutions are needed for both pre-bond and post-bond test of 3D stacked chips. In pre-bond test, each die or wafer is tested separately, so that only fault-free parts are bonded and packaged. Pre-bond test increases the overall yield of 3D stacking but it requires major advances in probe access and DFT methods for partial designs on 3D layers. Post-bond test, in which a 3D chip is tested after stacking, is necessary to reduce defect escape and qualify the assembly process.

In this paper, we address TAM optimization for post-bond test time minimization for 3D SOCs using the wrapper design method from [3]. We do not consider pre-bond testing in this work. While there are obvious similarities between TAM optimization for 2D ICs and that for 3D ICs, additional constraints related to the number of TSVs, placement of cores on different layers, and thermal limits must be taken into account for 3D integration. In 3D SOCs, the various cores are placed on different layers [18]; 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 nearest to the package for post-bond test, and due to the limited availability of TSVs for testing, the availability of bitwidth for test-access to cores on higher layers is adversely affected. TSVs not only occupy considerable area on a chip (especially with the need for a "keep-out area" for each TSV), but most of them are needed for functional interconnects, power/ground routing, and clock signals. Therefore, the test-access infrastructure needs to be designed to use only a few TSVs.

The rest of the paper is organized as follows: Section 2 uses a simple example to illustrate the problem investigated in this paper. Section 3 presents related prior work on 3D ICs and TAM optimization for 2D ICs. Section 4 describes integer linear programming models, as well as a heuristic method based on LP-relaxation and randomized rounding. It addresses both the Test Bus and TestRail architectures, and constraints related to wire length. Section 5 presents experimental results on benchmark SOCs. Finally, Section 6 presents conclusions and outlines directions for future work.

2. Problem formulation

2.1. Motivational example

Fig. 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 2 W 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 the different TAMs used to access the cores. Fig. 2 shows two possible TAM designs for the same 3D SOC design:

- *Approach* 1: In Fig. 2(a), we have two TAMs of width $w\,1$ and $w\,2$, respectively, where $w\,1 + w\,2 = W$ and the core on Layer 1 is accessed using the second TAM. A total of $2 \cdot w1$ TSVs are needed for this test-infrastructure design.
- *Approach* 2: Fig. 2(b) presents an alternative design in which three TAMs are used and the core of Layer 1 is accessed using the third TAM ($v\,1 + v\,2 + v\,3 = W$).

Note that both $2 \cdot w1$ (in Fig. 2(a)) and $2 \cdot v3$ (in Fig. 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 *TAM* 1 has to sequentially test 3 cores; In

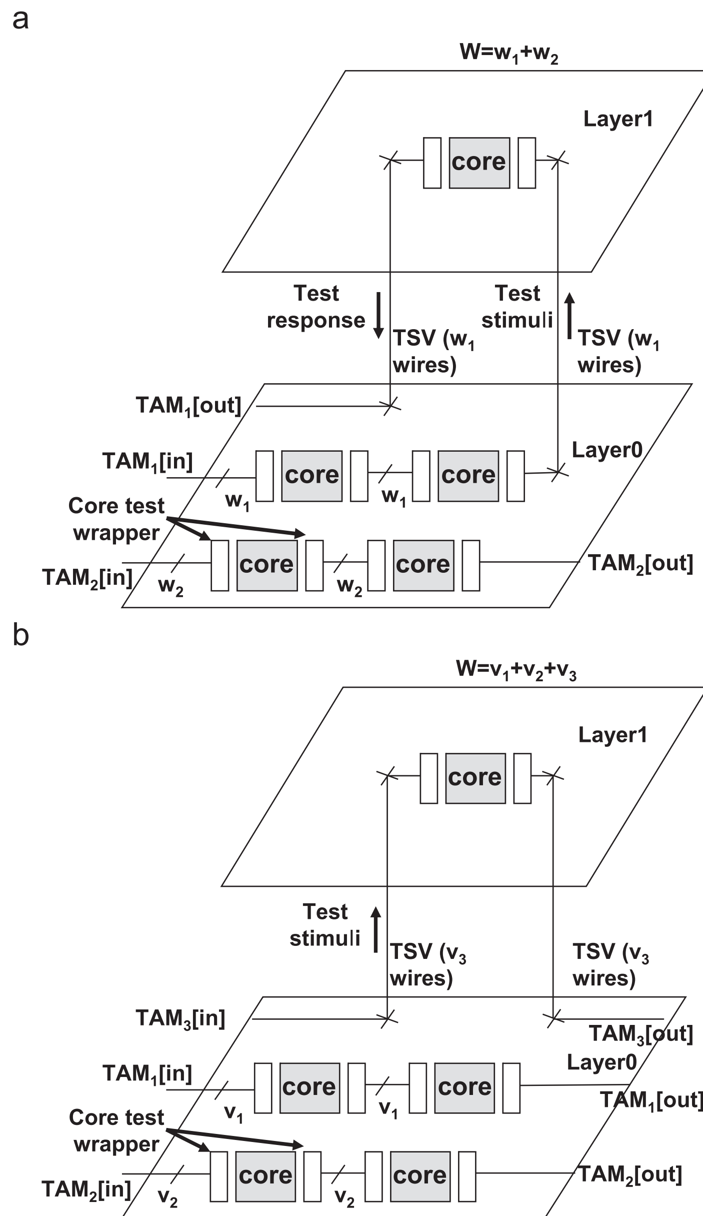

Fig. 2. Two solutions for TAM optimization in a 3D IC.

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.

2.2. Problem statement

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 $3DP_{PAW}$ (PAW stands for Partition of TAM width, Assignment of cores to each TAM, and Wrapper design for each core) 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 [3]. The first problem $3DP_W$ (W stands for Wrapper design for each core) addresses test-wrapper design, which is identical to 2D wrapper design [3] since we assume that any given core is placed on only one layer (i.e., a core is not partitioned across multiple layers). The reason that we choose coarse partitioning (one single core is placed in one layer) instead of fine grain partitioning

(one core is partitioned into several layers) is that fine grain partitioning results in larger number of TSVs and increases design and testing complexity. The second problem $3DP_{AW}$ (*AW* stands for Assignment of cores to each TAM, and Wrapper design for each core) addresses 3D TAM optimization; it considers a predefined TAM width for each TAM partition. 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 [19] in [3]. Therefore, we also develop a heuristic technique based on a combination of ILP modeling, LP-relaxation, and randomized rounding [20] to efficiently obtain near-optimal results.

3. Related prior work

A number of test-access architectures and TAM optimization methods have been proposed in the literature. These include multiplexed access [21], partial isolation rings [22], core transparency [23], dedicated test bus [24], TestRail architecture [25,26], etc. TAM optimization and test scheduling problems have been tackled using ILP [3,27], rectangle packing [10,28], iterative refinement [4,29], and other heuristics [7,30]. Wirelength and test time minimization has also been studied [31,32]. However, all these methods are targeted at traditional 2D technologies.

The 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 [13–15]. Among all EDA challenges for 3D IC design, tools and methodologies for 3D IC testing are regarded as the "No.1 challenge" [33]. 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 [34]. Wu et al. have proposed several 3D scan-chain design techniques [35]. A recently developed TAM optimization technique does not impose any limits on the number of TSVs used for the TAM, but considers pre-bond test considerations and wirelength limits [36]. A drawback of this approach is that since it does not limit the number of TSVs for the TAM, it ignores constraints related to the "keep-out area" that is associated with a TSV. Recent work has also been reported on the testing of TSVs [37] and an optimization method for reusing the TAM for both pre-bond and post-bond test [38].

4. Optimization of 3D TAMs

In this section, we first 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. In Section 4.6, we show that the model can be easily modified for TestRail architecture. We do not address the design of hierarchical TAMs [39,40] or test scheduling for hierarchical SOCs [40]. The SOC hierarchy is flattened for the purpose of TAM design and hierarchical cores are considered to be at the same design level in test mode. We first review the wrapper-design problem $3DP_W$ from [41] and [3]. 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.

4.1. 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 s_i (s_o) 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 $T=(1+\max(s_i,s_o))\cdot p_i+\min(s_i,s_o)$, where p is the number of test patterns [41]. Note that T decreases as both s_i and s_o are reduced, i.e., as the wrapper scan-in (and scan-out) chains become more equal in length.

The test-wrapper design problem P_W is defined as follows [41,3]: Given a core with n functional inputs, m functional outputs, sc internal scan chains of lengths $l_1, l_2,\ldots, l_{sc}$, respectively, and TAM width k, assign the $n+m+sc$ wrapper scan chain elements to $k_1 \le k$ wrapper scan chains such that (i) $\max\{s_i, s_o\}$ is minimized, where s_i (s_o) is the length of the longest wrapper scan-in (scan-out) chain, and (ii) k_1 is minimum, subject to priority (i).

We adopt the approximation algorithm based on the best fit decreasing (BFD) heuristic [3,19] to solve P_W efficiently. The algorithm has three main 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).

4.2. 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 [42]. A typical ILP model can described as follows:

Minimize $\mathbf{Cx}$ subject to $\mathbf{Ax} \le \mathbf{B}$, where $\mathbf{x} \ge 0$ ($\le$ means less than or equal to). In this model, $\mathbf{x}$ represents a vector of variables, $\mathbf{C}$ is a cost vector, $\mathbf{A}$ is a constraint matrix, and $\mathbf{B}$ 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 $w_1, w_2,\ldots,w_B$, 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 [3]: $T_i(w_j)=(1+\max(s_i,s_o))\cdot p_i+\min(s_i,s_o)$ 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 [3], we define the binary variable x_{ij} as follows:

$$x_{ij} = \begin{cases} 1 & \text{if Core } i \text{ is assigned to TAM } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

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 [17]. 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*

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

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 k 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 k 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 k, 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

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

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 for each TAM partition. The complete mathematical programming model for $3DP_{AW}$ is shown in Fig. 3 and as a final step, the min–max objective and max constraints can be linearized as in [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 $N + 2 \cdot B + 2 \cdot N^2 \cdot B = O(N^2 \cdot B)$.

Objective:

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

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 mathematical programming model for $3DP_{AW}$.

4.3. 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 [3]. Therefore, we next address the more general problem $3DP_{PAW}$, as defined in Section 2.

From Theorem 1 in [3], 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

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

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

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

The testing time $T_i(k)$ is obtained from the Design Wrapper procedure [3] 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) $x_{ij} + d_{jk} - y_{ijk} \le 1$
(2) $x_{ij} + d_{jk} - 2 \cdot y_{ijk} \ge 0$.

The number of TSVs for chain k is as follows:

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

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 $d_{km} \cdot z_{ijk}$. Two more constraints are added:

(1) $d_{km} + z_{ijk} - n_{ijkm} \le 1$
(2) $d_{km} + z_{ijk} - 2 \cdot n_{ijkm} \ge 0$.

Now the number of TSVs becomes

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

where L_{ij} is defined as $L_{ij} = \max_{1 \le i,j \le N}\{(L_i - L_j), L_i, L_j\}$. The complete mathematical programming model for the $3DP_{PAW}$ problem is shown in Fig. 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 $1 + 4 \cdot B + N + 2 \cdot N^2 \cdot B + 2 \cdot N^2 \cdot B \cdot W = O(N^2 \cdot B \cdot W)$.

4.4. Thermal-aware $3DP_{PAW}$

Power and thermal issues have become a primary concern in traditional 2D IC testing. The design under test (DUT) can consume significantly more power in test mode than in normal operation. Consequently, the heat dissipation during test mode becomes an major problem since the temperature depends on the power density. The stacking of multiple active layers in 3D design leads to even higher power densities than for the 2D counterpart, thereby exacerbating thermal problems. With increased temperature, the chip under test may be damaged permanently. In addition, the test results may be invalidated due to changes in the path delay caused by "hotspots". Therefore, it is essential to conduct thermal-aware 3D TAM optimization, maintaining test time requirements without violating a temperature threshold.

If several cores with high test power consumption are assigned to different TAM partitions, they may be tested at the same time depending on the test scheduling, causing very high test power dissipation. There are two ways to tackle this problem. First, "post-processing" can be done to reorder the cores in test scheduling to ensure that the parallelism is limited. During test scheduling, the order in which the cores are tested is changed. In this way, cores with high power consumption are tested in parallel to a lesser extent. The second solution is to add constraints in the TAM design to limit test parallelism of "high-power" cores. Since we do not directly consider test scheduling in our model, the second solution is easier to implement. The constraints can be added to 3D TAM optimization problem $3DP_{PAW}$. We count the number of pairs of cores that are assigned to different TAM partitions to estimate the amount of parallelism. A new binary constant P_{ij} is added. If Core i are j are high-power cores then $P_{ij} = 1$. We also set an upper limit Pr on the number of pairs of cores that can be tested in parallel. Therefore, we have a new constraint:

$$\sum_{i=1,j=1}^{N} P_{ij} \cdot x_{ik_1} \cdot x_{jk_2} \le 2 \cdot Pr$$

where $1 \le k_1, k_2 \le B, k_1 \ne k_2$.

As before, the nonlinear equation $q_{ijk_1k_2} = x_{ik_1} \cdot x_{jk_2}$ can be linearized by treating $q_{ijk_1k_2}$ as an independent variable and adding two new constraints:

(1) $x_{ik_1} + x_{jk_2} - q_{ijk_1k_2} \le 1$
(2) $x_{ik_1} + x_{jk_2} - 2 \cdot q_{ijk_1k_2} \ge 0$.

Objective:

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

Subject to:

$$\sum_{j=1}^{B} x_{ij} = 1 (1 \le i \le N)$$
$$x_{ij} + d_{jk} - y_{ijk} \le 1$$
$$x_{ij} + d_{jk} - 2 \cdot y_{ijk} \ge 0$$
$$w_j = \sum_{k=1}^{w_{max}} k \cdot d_{jk}$$
$$\sum_{k=1}^{w_{max}} d_{jk} = 1$$
$$\sum_{j=1}^{B} w_j \le W$$
$$d_{km} + z_{ijk} - n_{ijkm} \le 1$$
$$d_{km} + z_{ijk} - 2 \cdot n_{ijkm} \ge 0$$
$$L_{ij} = \max_{1 \le i,j \le N} \{(L_i - L_j), L_i, L_j\}$$
$$v_k = \sum_{m=1}^{w_{max}} 2 \cdot m \cdot \max_{1 \le i,j \le N, 1 \le k \le B} \{n_{ijkm} \cdot L_{ij}\}$$
$$\sum_{k=1}^{B} v_k \le V_{total}$$
$$\forall k \quad v_k \le V_{single}$$

Fig. 4. The mathematical programming model for $3DP_{PAW}$.

4.5. 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 [19]; however, linear programming (LP) problems can be solved optimally in polynomial time [42]. Therefore, we adopt the method of LP-relaxation and combine it with randomized rounding [20]. 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 $x_{kj} (i \ne k)$ 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.

4.6. TestRail architecture

Previous subsections have focused on Test Bus architecture, in which modules connected to the same test bus can only be tested

sequentially. Therefore, the overall SOC test time is the maximum test time over all the test buses. Another test architecture that has widely advocated for core-based SOC testing is called TestRail [25,26]. In this architecture, modules connected to the same TestRail can be tested simultaneously and sequentially as well. The advantage of TestRail architecture over the Test Bus architecture is that it allows module-external testing, i.e., to test the circuitry and wires between the modules. The extension of the optimization framework to TestRail is straightforward. Assuming that the modules connected to the same TestRail can be tested simultaneously, the test time on a TAM partition for the TestRail architecture is the maximum test time of the individual modules $\max_{1 \le j \le B, 1 \le i \le N}(T_i(w_j) \cdot x_{ij})$ on that partition plus the bypass time and load pattern latency. The constraints are the same as for Test Bus architecture. We have evaluated 3D test-access optimization for the TestRail architecture, and results are presented in Section 5.6.

4.7. Test-application time and wire-length co-optimization

In SOC test-architecture design, wire-length minimization is also an important objective, besides test time minimization [31,32]. In this section, we discuss how to incorporate wire length into our optimization model for the Test Bus architecture. Our wire-length cost model extends [32] by including the vertical interconnect length due to TSVs. The placement of cores on the various layers and within a layer is obtained from a thermal-aware 3D floorplanning and placement tool [15]. The distance between two modules is calculated in terms of the Manhattan distance, including vertical TSV length if these two modules are in different layers.

Our objective is to minimize the test time as well as the total wire length connecting the core to form the TAM partitions (chains). Assume that the number of TAM chains is B and the number of cores is N. We define a binary variable c_{ijk}, $1 \le i \le N, 1 \le j \le N$, such that $c_{ijk}=1$ if Core j immediately connects Core i in TAM chain k. Each TAM chain k has TAM-in pin b_k and TAM-out pin e_k, which are predefined in our model. The interconnect length from Core i to Core j is defined by t_{ij}, which is obtained from the thermal-aware floorplanning and placement tool. The vertical length of TSVs is set to be 50 μm. The additional constraints due to wire-length incorporation are listed in Fig. 5.

In Fig. 5, line 1 provides a constraint that there is only one immediate successor per core in each TAM chain. Line 2 ensures that there is only one immediate predecessor per core in each TAM chain. Line 3 implies that there is no immediate predecessor for TAM-in pin in each TAM chain. Line 4 ensures that there is no immediate successor for TAM-out pin in each TAM chain. Line 5 models the limitation that one core cannot connect to itself. Line 6 and 7 indicate that one core can only be in one TAM chain. Line 8–11 prevent the generation of cycles in each TAM chain. A new binary variable u_{ijk} is defined, indicating whether cell i is located before cell j in the ordered scan chain. Since the term $c_{ihk} \cdot u_{hjk}$ in line 8 is nonlinear, it is replaced by a new binary variable s_{ijhk}. In addition, two new constraints are added:

$$c_{ihk}+u_{hjk} \le s_{ijhk} \quad \text{and} \quad c_{ihk}+u_{hjk} \ge 2 \cdot s_{ijhk}$$

Line 9 introduces the constraint that Core i either is before Core j or after Core j in the TAM chain if those two cores are in that chain. Line 10 and 11 constrain the source nodes and end nodes of each chain since the source node is before every other node and the end node is after every other node in each chain. The wire length of each TAM chain is the sum of the distance between the TAM-in pin and the core connected to that pin, the distances between all subsequent pairs of cores in that TAM chain, and the

$$
\begin{array}{ll}
1. & \forall i \quad \sum_{j=1}^{N} \sum_{k=1}^{B} c_{ijk} = 1 \\
2. & \forall j \quad \sum_{i=1}^{N} \sum_{k=1}^{B} c_{ijk} = 1 \\
3. & \forall j \quad \sum_{i=1}^{N} \sum_{k=1}^{B} c_{iu_j k} = 0 \\
4. & \forall i \quad \sum_{j=1}^{N} \sum_{k=1}^{B} c_{v_i jk} = 0 \\
5. & \forall i, k \quad c_{iik} = 0 \\
6. & \forall i \quad \sum_{j=1}^{N} \sum_{k=1}^{B} c_{ijk} = 1 \\
7. & \forall j \quad \sum_{i=1}^{N} \sum_{k=1}^{B} c_{ijk} = 1 \\
8. & \forall i, j, k \quad u_{ijk} = \sum_{h=1}^{N} c_{ihk} \cdot u_{hjk} + c_{ijk} \\
9. & \forall i, j, k \quad u_{ijk} + u_{jik} \le 1 \\
10. & \sum_{k=1}^{B} \sum_{i=1}^{N} \sum_{j=1}^{N} u_{b_k ij} = N \\
11. & \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{B} u_{ije_k} = N
\end{array}
$$

Fig. 5. New constraints for wire-length incorporation in the optimization flow.

distance between TAM-out pin and the core connected to it. The location of TAM-in and TAM-out pins are predefined. The length for TAM chain k is calculated as $l_k = w_k \cdot (\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ijk} \cdot t_{ij})$, where w_k is the width of the TAM chain. The total wire length is the sum of the wire lengths of each TAM chain, i.e. $L = \sum_{k=1}^{B} l_k$. The objective function in the ILP model is changed to

$$A \cdot \left(\max_j \sum_{i=1}^{N} \sum_{k=1}^{w_{max}} d_{jk} \cdot T_i(k) \cdot x_{ij} \right) + (1-A) \cdot L$$

where A is a weight used in the multiobjective cost function ($0 \le A \le 1$). Since wire length and test time are measures with different units, we use normalized values in the objective function. We divide the test time T_i by the maximum test time (corresponding $A=0$ when the wire-length minimization is given maximum weight), and divide wire length L by the maximum wire length (corresponding to $A=1$ when test-time minimization is given maximum weight). Since w_k in problem $3DP_{PAW}$ is also a variable, the multiplication of two non-binary variables is not supported by standard ILP solvers. Therefore, we incorporate wire-length minimization into problem $3DP_{AW}$, in which w_k is predefined. In order to incorporate wire length to $3DP_{PAW}$, a heuristic approach is needed, which is left for future work.

5. Experiments and results

5.1. 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 [43]. We use Xpress-MP [44], a commercial ILP solver, to solve the ILP model with randomized rounding for the $3DP_{PAW}$ problem.

5.2. 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 [14–16]. Therefore, it is essential to have a thermal-aware floorplanner for 3D SOC

design. In this paper, we adopt the thermal-aware floorplanner from [15] 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. In thermal-aware floorplanning, each core is considered as a block with fixed coordinates for the block. The inputs of the floorplanner include the area of each core (including x and y positions, assume the layout is a square) and the power consumption. 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 [43], we scale down the test power to 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 with layer assignment as well as x and y positions of each core, which is used as an input to our 3D TAM optimization models.

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

Fig. 6 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.

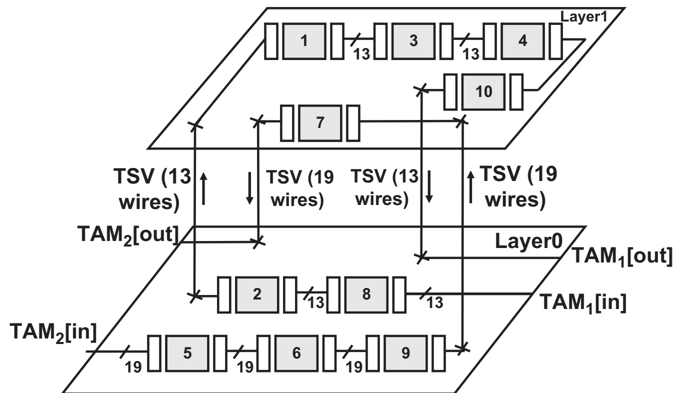

Fig. 6. TAM design obtained for d695 (B=2, W = 32, two layers).

Table 1 presents 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 $W \le TSV/2$, 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. Tables 2–4 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.

We draw several conclusions from the results in Tables 1–4. 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

Table 1
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)	CPU time for 3D (min)
16	2	(7,9)	44 225	24 257	20 355	0.04
	3	(4,5,7)	46 108	22 984	19 714	0.13
	4	(1,1,4,10)	51 707	27 403	23 380	0.3
24	2	(5,19)	31 070	21 953	18 868	0.04
	3	(1,4,19)	31 077	20 764	16 221	0.14
	4	(1,3,4,16)	41 611	16 549	13 984	0.5
32	2	(11,21)	28 524	20 934	17 101	0.04
	3	(4,10,18)	23 944	16 085	13 798	0.15
	4	(2,4,10,16)	33 782	13 505	10 334	1.0
40	2	(19,21)	25 761	18 604	15 364	0.04
	3	(4,17,19)	21 070	13 095	11 720	0.15
	4	(4,4,13,19)	26 961	11 282	8437	2.0
48	2	(21,27)	25 761	17 331	14 232	0.06
	3	(7,19,22)	21 070	11 808	9375	0.16
	4	(3,4,7,34)	25 086	9522	6775	1.0
56	2	(23,33)	23 256	15 392	13 768	0.06
	3	(4,19,33)	21 070	11 808	8112	0.17
	4	(2,4,16,34)	24 852	8452	6177	1.1
64	2	(23,41)	23 232	13 984	12 153	0.06
	3	(5,22,37)	21 070	9731	6860	0.21
	4	(13,16,16,19)	24 852	7923	5752	2.2

Table 2
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)	CPU time for 3D (min)
16	2	(5,11)	334 276	289 986	265 329	0.3
	3	(4,4,8)	323 526	275 537	257 791	3.7
	4	(1,4,4,7)	421 172	286 734	286 146	1.6
24	2	(11,13)	326 153	263 892	236 842	0.6
	3	(1,10,13)	299 590	254 732	220 664	3.8
	4	(1,1,10,12)	352 227	249 274	233 497	2.1
32	2	(7,25)	314 522	230 925	223 984	0.8
	3	(10,4,18)	289 293	232 810	181 696	3.8
	4	(1,10,10,11)	348 318	205 831	166 188	9.0
40	2	(13,27)	310 323	215 506	203 672	0.5
	3	(10,13,17)	287 143	201 221	158 628	21.0
	4	(7,10,4,19)	348 318	181 687	144 182	26.1
48	2	(13,35)	306 389	205 782	192 256	0.9
	3	(7,13,28)	283 272	172 191	133 103	4.7
	4	(10,10,7,21)	289 710	169 392	145 543	13.8
56	2	(15,41)	304 831	204 155	179 474	2.3
	3	(10,13,33)	281 965	153 282	123 137	9.8
	4	(4,10,16,26)	285 825	162 818	141 406	10.1
64	2	(15,49)	304 810	179 688	170 123	0.5
	3	(13,19,32)	279 400	142 210	121 908	21.3
	4	(10,13,19,22))	285 825	157 460	138 737	130.0

Table 3
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)	CPU time for 3D (min)
16	2	(7,9)	1 338 043	1 134 919	1 113 301	0.2
	3	(1,7,8)	1 365 702	999 543	999 201	0.9
	4	(1,4,4,7)	2 025 402	1 288 341	1 174 558	0.5
24	2	(9,15)	1 202 389	913 550	859 758	0.1
	3	(7,7,10)	1 055 242	762 841	762 669	0.8
	4	(1,1,10,12)	1 377 077	798 449	743 508	18.5
32	2	(11,21)	1 183 301	843 301	779 966	0.1
	3	(10,7,15)	1 045 242	665 445	592 325	0.3
	3	(4,7,10,11)	1 359 846	684 524	563 604	9.2
40	2	(13,27)	1 047 913	752 782	716 114	0.2
	3	(13,7,20)	982 127	552 231	506 506	0.8
	4	(4,1,16,19)	1 026 819	584 301	497 810	100
48	2	(13,35)	1 026 760	693 465	671 796	0.4
	3	(7,16,25)	982 127	546 152	490 218	1.4
	4	(1,7,19,21)	995 186	544 579	474 016	13.5
56	2	(13,43)	1 014 317	637 606	621 753	2.3
	3	(16,7,33)	922 127	540 069	485 039	4.5
	4	(1,7,19,29)	972 797	544 579	455 916	6.7
64	2	(13,51)	987 564	559 758	539 403	0.5
	3	(16,7,41)	932 476	534 212	481 386	17.0
	4	(10,7,19,28)	948 945	544 330	429 201	19.0

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 [3]. 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 [3]. The fourth conclusion is that the results obtained before and after randomized rounding are within reasonable range, indicating the efficiency of the randomized rounding technique.

5.4. *Dependence of test time on 3D placement, the number of layers, and the maximum number of TSVs*

In this subsection, we examine the dependence of test time on different 3D placements, the number of layers, and upper limits on the number of TSVs. Fig. 7 shows the dependence of the test time on different 3D placements, instead of the thermal-aware 3D floorplanner [15], for all benchmarks. The limit on the number of TSVs is assumed to be the same for each benchmark. Three different values of W, namely $W=16$, 32, 64 are considered. We examine three placements: tough, random, and nice. Tough placement places larger cores on top layers, starving the test

Table 4
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)	CPU time for 3D (min)
16	2	(7,9)	1 869 200	1 800 413	1 712 598	0.3
	3	(4,4,8)	2 021 836	1 779 401	1 779 298	1.6
	4	(1,4,4,7)	3 651 006	2 179 527	2 177 620	1.1
	5	(1,1,4,1,9)	4 379 506	2 030 868	1 913 331	1.4
24	2	(7,17)	1 294 512	1 259 711	1 169 532	0.3
	3	(4,4,16)	1 316 252	1 197 683	1 108 672	1.6
	4	(1,7,7,9)	1 709 129	1 337 682	1 303 728	6.3
	5	(1,1,7,7,8)	2 115 963	1 263 801	1 152 322	8.4
32	2	(9,23)	966 752	894 463	860 629	0.3
	3	(4,4,24)	970 585	910 957	891 713	2.4
	4	(1,10,10,11)	1 296 534	944 807	944 706	14
	5	(4,4,7,4,13)	1 913 443	1 008 684	972 991	17
40	2	(15,25)	778 480	778 296	767 018	0.7
	3	(7,10,23)	874 504	817 805	750 008	46.4
	4	(1,4,16,19)	1 017 802	890 145	786 207	63.0
	5	(4,4,10,1,21)	1 430 320	780 202	728 064	40.8
48	2	(15,33)	766 270	758 806	644 230	1.3
	3	(7,16,25)	827 277	722 042	632 076	7.7
	4	(4,13,13,18)	933 243	713 347	650 231	25.2
	5	(1,10,10,7,20)	1 349 575	727 978	660 292	35.8
56	2	(11,45)	686 577	662 686	626 878	3.1
	3	(13,16,27)	685 314	635 095	536 410	72.8
	4	(1,7,22,26)	889 290	664 447	608 740	60.0
	5	(1,1,16,1,37)	1 300 670	680 635	628 872	65.1
64	2	(15,49)	662 198	630 365	614 287	5.5
	3	(16,22,26)	662 808	572 342	488 366	85.5
	4	(1,13,22,28)	870 037	604 553	520 967	50.4
	5	(1,3,16,7,37)	1 273 410	621 235	533 928	77.1

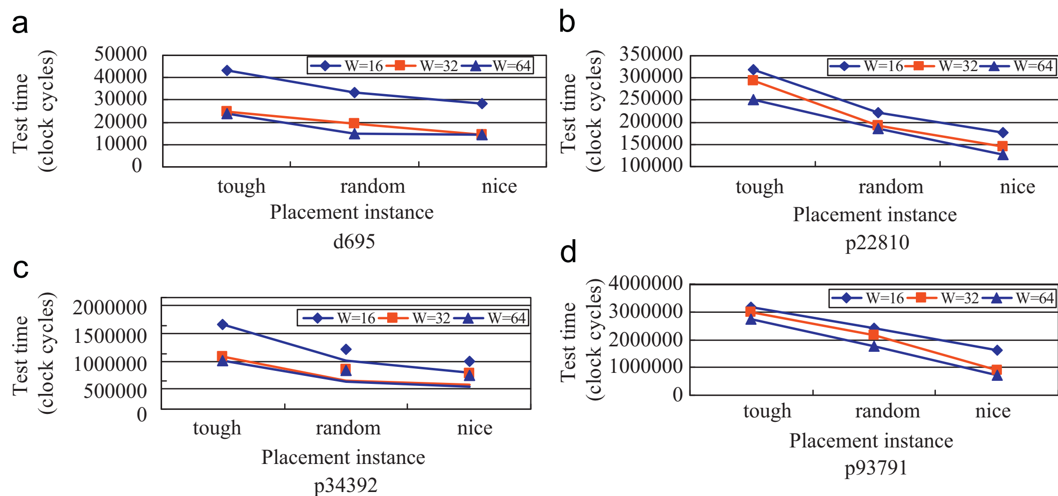

Fig. 7. Test time dependence on 3D placement for different benchmarks.

bitwidth under the limit on the number of TSVs. Random placement is generated randomly. Nice placement places larger cores on the bottom layer. We note that different placements result in considerably different test times for the same value of W. Therefore, an effective placement in the 3D SOC design space is essential for minimizing the test time. Fig. 8 shows the dependence of test time on the limit on the number of TSVs for all benchmarks. When the number of TSVs exceeds a certain limit, there is no further reduction in the test time because sufficient bitwidth is now available for the higher layers. Similar results are obtained for the three larger benchmark SOCs. Fig. 9 shows the dependence of test time on the number of layers for all benchmarks. With an increase in the number of layers, the test time usually increases. The limit in the number of TSVs reduces the bitwidth for cores placed on higher layers.

5.5. Results for thermal-aware $3DP_{PAW}$

Finally, we present results for thermal-aware $3DP_{PAW}$. Tables 5–8 compare the TAM partitions, test times, and CPU run times for $3DP_{PAW}$ with that for thermal-aware $3DP_{PAW}$ for benchmarks d695, p22 810, p34 392, and p93 791, respectively. The test power for each module is obtained from [43]. For example, in d695, there are four cores, namely Cores 2,5,7, and 10, which consume significantly more power than the other cores in test mode. Therefore, we

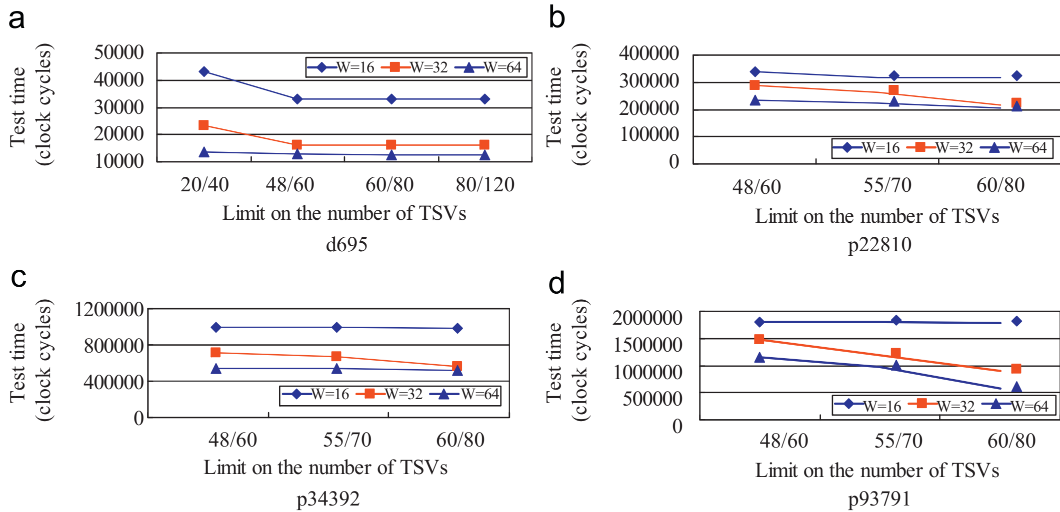

Fig. 8. Test time dependence on limits on the number of TSVs for different benchmarks.

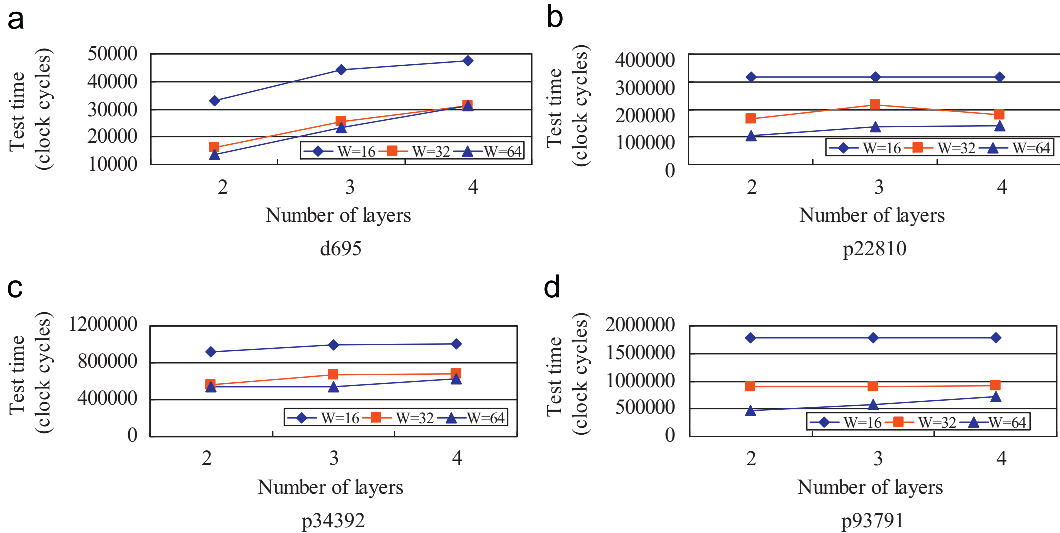

Fig. 9. Test time dependence on number of layers for different benchmarks.

set $P_{ij} = 1$ for $i,j \in \{2,5,7,10\}$. We set the upper limit Pr to be 2, limiting the test parallelism for high-power cores. For p22 810, p34 392, and p93 791, the upper limit Pr is set to be 2, 3, and 4 since there are 6, 8, and 10 high-power cores for each benchmark. The results show that the optimized TAM partitions are different when thermal constraints are taken into account. The test time is higher with thermal constraints but it is still less than that for the 2D case in d695, p22 810 and p34 392. However, for p93 791, the test time with thermal constraints is even higher than for the 2D case because of the limited amount of test parallelism available. As a tradeoff, the thermal constraints can be relaxed according to the requirement of the test-time budget and the cooling solutions available.

5.6. Result for TestRail architecture

In this section, we provide the test-time results for benchmark *d* 695 for the TestRail architecture. Table 9 illustrates the test time comparison of *d* 695 between Test Bus architecture and TestRail architecture. Column 1 lists the total TAM width. The number of TestRails is provided in column 2. Columns 3 and 4 give the test time in 2D case for Test Bus and TestRail architecture. Columns 5 and 6 present the TAM partition and the test time for the Test Bus architecture. Columns 7 and 8 show the TAM partitions and the test time for the TestRail architecture. Since we assume a parallel schedule in TestRail architecture, the cores connected to one TestRail are tested in parallel. When one of the cores finishes testing using its given test patterns, the bypass mode of this core is turned on. Other cores continue to be tested. The bypass takes one cycle in the test access path. The latency to load patterns (at the beginning of the test) is normally several cycles depending on how many cores are connected to the TestRail. Since the number of cores in one TestRail is small in *d* 695, the bypass and load pattern latency is small enough to be ignored in the test time result.

We observe that the test time for TestRail is less than that for Test Bus since it allows parallel testing for each TestRail. Another observation is that when the total TAM width is larger than 24, the test time for TestRail remains unchanged, even though the TAM partitions are different. The reason is that the test time in the TestRail architecture is determined by individual cores instead of the sum of cores connected to one chain. We also show that a 3D-aware TestRail optimization method, as described in this

Table 5
Comparison between $3DP_{PAW}$ and thermal-aware $3DP_{PAW}$ for SOC d695 for $B = 2, 3, 4$ (two layers, TSV limits 48/60).

W	B	2D test time (#CC)	Thermal-oblivious			Thermal-aware		
			TAM partition	Test time (#CC)	CPU time (min)	TAM partition	Test time (#CC)	CPU time (min)
16	2	44 225	(7,9)	24 257	0.04	(5,11)	44 815	0.8
	3	46 108	(4,5,7)	22 984	0.13	(2,5,9)	44 809	1.3
	4	51 707	(1,1,4,10)	27 403	0.3	(1,1,5,9)	44 809	14.5
24	2	31 070	(5,19)	21 953	0.04	(3,21)	31 170	0.2
	3	31 077	(1,4,19)	20 764	0.14	(1,2,21)	31 132	7.5
	4	41 611	(1,3,4,16)	16 549	0.5	(1,1,1,21)	31 132	53.8
32	2	28 524	(11,21)	20 934	0.04	(16,16)	26 578	0.4
	3	23 944	(4,10,18)	16 085	0.15	(4,9,19)	21 856	46.8
	4	33 782	(2,4,10,16)	13 505	1.0	(1,2,10,19)	22 427	20.0
40	2	25 761	(19,21)	18 604	0.04	(20,20)	22 848	0.6
	3	21 070	(4,17,19)	13 095	0.15	(5,9,26)	21 005	10.9
	4	26 961	(4,4,13,19)	11 282	2.0	(1,4,9,26)	21 005	7.6
48	2	25 761	(21,27)	17 331	0.06	(16,32)	20 314	0.9
	3	21 070	(7,19,22)	11 808	0.16	(4,12,32)	18 799	2.0
	4	25 086	(3,4,7,34)	9522	1.0	(2,2,12,32)	18 799	43.4
56	2	23 256	(23,33)	15 392	0.06	(24,32)	19 457	0.9
	3	21 070	(4,19,33)	11 808	0.17	(5,17,34)	13 575	1.3
	4	24 852	(2,4,16,34)	8452	1.1	(1,4,17,34)	13 575	13.8
64	2	23 232	(23,41)	13 984	0.06	(18,32)	19 457	0.8
	3	21 070	(5,22,37)	9731	0.21	(7,17,39)	12 941	1.3
	4	24 852	(13,16,16,19)	7923	2.2	(3,6,16,39)	12 841	14.2

Table 6
Comparison between $3DP_{PAW}$ and thermal-aware $3DP_{PAW}$ for SOC p22810 for $B = 2, 3, 4$ (two layers, TSV limits 60/80).

W	B	2D test time (#CC)	Thermal-oblivious			Thermal-aware		
			TAM partition	Test time (#CC)	CPU time (min)	TAM partition	Test time (#CC)	CPU time (min)
16	2	334 276	(5,11)	289 986	0.3	(6,10)	324 902	0.2
	3	323 526	(4,4,8)	275 537	3.7	(4,4,8)	321 172	1.6
	4	421 172	(1,4,4,7)	286 734	1.6	(1,1,1,13)	309 905	0.4
24	2	326 153	(11,13)	263 892	0.6	(11,13)	308 674	1.1
	3	299 590	(1,10,13)	254 732	3.8	(1,10,13)	282 849	0.5
	4	352 227	(1,1,10,12)	249 274	2.1	(1,4,4,15)	271 146	4.2
32	2	314 522	(7,25)	230 925	0.8	(15,17)	270 379	0.3
	3	289 293	(10,4,18)	232 810	3.8	(6,10,16)	258 778	38.9
	4	348 318	(1,10,10,11)	205 831	9.0	(4,1,13,14)	240 120	45.5
40	2	310 323	(13,27)	215 506	0.5	(19,21)	265 452	0.7
	3	287 143	(10,13,17)	201 221	21.0	(10,13,17)	230 462	47.8
	4	348 318	(7,10,4,19)	181 687	26.1	(3,7,13,17)	220 194	73.5
48	2	306 389	(13,35)	205 782	0.9	(19,29)	265 452	0.5
	3	283 272	(7,13,28)	172 191	4.7	(10,13,25)	226 844	37.4
	4	289 710	(10,10,7,21)	169 392	13.8	(4,13,10,21)	214 379	297.5
56	2	304 831	(15,41)	204 155	2.3	(19,37)	245 234	0.3
	3	281 965	(10,13,33)	153 282	9.8	(10,13,33)	226 813	17.6
	4	285 825	(4,10,16,26)	162 818	10.1	(10,4,13,29)	189 146	315
64	2	304 810	(15,49)	179 688	0.5	(19,45)	245 234	0.7
	3	279 400	(13,19,32)	142 210	21.3	(13,10,41)	223 079	6.5
	4	285 825	(10,13,19,22))	157 460	130.0	(1,13,10,40)	181 698	407

paper, leads to lower test time than the use of a 3D-oblivious 2D optimization approach, as published earlier.

5.7. Result for test time and wire-length co-optimization

Table 10 presents the test time and wire length results for problem $3DP_{AW}$, in which the width of each TAM chain is predefined. Column 1 lists the number of TAM chains. Column 2 lists the width of each TAM chain. Columns 3–5 show the test time with different values of weight A. Columns 6–8 show the wire length for different values of A. We make some key observations: (1) the test time reduces or remains unchanged with an increase of A since larger values of A add more weight to the test time in the objective function. Due to the predefined width of each TAM chain, the optimization search space is limited, so that for some cases, the test time is the same for $A=0.5$ and $A=1$; (2) the wire length increases or remains unchanged with an increase in A since larger A leads to less weight for the test time in the objective function; (3) when the number of TAM chains increases, the test time decreases as expected

Table 7
Comparison between $3DP_{PAW}$ and thermal-aware $3DP_{PAW}$ for SOC p34392 for $B = 2, 3, 4$ (two layers, TSV limits 55/70).

W	B	2D test time (#CC)	Thermal-oblivious			Thermal-aware		
			TAM partition	Test time (#CC)	CPU time (min)	TAM partition	Test time (#CC)	CPU time (min)
16	2	1 338 043	(7,9)	1 134 919	0.2	(8,8)	1 081 570	1.9
	3	1 365 702	(1,7,8)	999 543	0.9	(1,7,8)	1 005 088	1.0
	4	2 025 402	(1,4,47)	1 288 341	0.5	(1,1,1,13)	1 391 607	3.8
24	2	1 202 389	(9,15)	913 550	0.1	(9,15)	1 064 994	5.5
	3	1 055 242	(7,7,10)	762 841	0.8	(1,10,13)	1 049 435	2.6
	4	1 377 077	(1,1,10,12)	798 449	18.5	(1,1,1,21)	1 391 607	0.8
32	2	1 183 301	(11,21)	843 301	0.1	(15,17)	831 177	6.7
	3	1 045 242	(10,7,15)	665 445	0.3	(10,7,15)	933 906	10.8
	4	1 359 846	(4,7,10,11)	684 524	9.2	(1,1,7,23)	834 826	7.0
40	2	1 047 913	(13,27)	752 782	0.2	(9,31)	831 177	7.0
	3	982 127	(13,7,20)	552 231	0.8	(7,16,17)	859 998	4.8
	4	1 026 819	(4,1,16,19)	584 301	100	(1,10,1,28)	790 995	45.5
48	2	1 026 760	(13,35)	693 465	0.4	(16,9)	808 844	5.7
	3	982 127	(7,16,25)	546 152	1.4	(7,13,28)	843 401	34.5
	4	995 186	(1,7,19,21)	544 579	13.5	(1,13,13,21)	754 307	26.5
56	2	1 014 317	(13,43)	637 606	2.3	(16,9)	808 844	5.6
	3	922 127	(16,7,33)	540 069	4.5	(10,13,33)	832 498	19.6
	4	972 797	(1,7,19,29)	544 579	6.7	(7,13,10,26)	661 844	225.3
64	2	987 564	(13,51)	559 758	0.5	(16,9)	808 844	5.6
	3	932 476	(16,7,41)	534 212	17.0	(10,13,41)	689 711	2.2
	4	948 945	(10,7,19,28)	544 330	19.0	(7,13,10,26)	661 844	201.7

Table 8
Comparison between $3DP_{PAW}$ and thermal-aware $3DP_{PAW}$ for SOC p93791 for $B=2, 3, 4$ (Four layers, TSV limits 60/80).

W	B	2D test time (#CC)	Thermal-oblivious			Thermal-aware		
			TAM partition	Test time (#CC)	CPU time (min)	TAM partition	Test time (#CC)	CPU time (min)
16	2	1 869 200	(7,9)	1 800 413	0.3	(3,13)	2 103 700	8.2
	3	2 021 836	(4,4,8)	1 779 401	1.6	(1,1,16)	2 229 152	1.9
	4	3 651 006	(1,4,4,7)	2 179 527	1.1	(1,1,2,12)	4 023 112	2.2
24	2	1 294 512	(7,17)	1 259 711	0.3	(4,20)	1 406 320	22.0
	3	1 316 252	(4,4,16)	1 197 683	1.6	(1,4,19)	1 412 113	3.5
	4	1 709 129	(1,7,7,9)	1 337 682	6.3	(1,1,1,21)	3 823 747	7.0
32	2	966 752	(9,23)	894 463	0.3	(5,20)	1 405 440	160.6
	3	970 585	(4,4,24)	910 957	2.4	(1,7,24)	1 009 262	6.9
	4	1 296 534	(1,10,10,11)	944 807	14	(1,1,4,26)	3 695 454	2.0
40	2	778 480	(15,25)	778 296	0.7	(5,20)	1 405 440	160.6
	3	874 504	(7,10,23)	817 805	46.4	(1,16,23)	999 931	53.8
	4	1 017 802	(1,4,16,19)	890 145	63.0	(1,10,1,28)	2 699 795	55.4
48	2	766 270	(15,33)	758 806	1.3	(5,20)	1 241 245	160.6
	3	827 277	(7,16,25)	722 042	7.7	(7,16,25)	995 650	96.3
	4	933 243	(4,13,13,18)	713 347	25.2	(4,13,1,30)	1 903 700	220.5
56	2	686 577	(11,45)	662 686	3.1	(5,20)	1 152 364	160.6
	3	685 314	(13,16,27)	635 095	72.8	(13,19,24)	995 522	302.8
	4	889 290	(1,7,22,26)	664 447	60.0	(4,1,25,26)	1 121 150	252.2
64	2	662 198	(15,49)	630 365	5.5	(5,20)	1 008 220	160.6
	3	662 808	(16,22,26)	572 342	85.5	(4,25,35)	995 522	46.6
	4	870 037	(1,13,22,28)	604 553	50.4	(1,10,25,28)	995 522	339.7

in most cases. However, when $A=0$, i.e., test time is not considered in the objective function, the test time does not follow this trend. Our results demonstrate that our optimization framework is general and it can easily incorporate wire-length constraints.

6. Conclusions

We have shown that a modular post-bond testing approach can be used for emerging 3D ICs. 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. 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. We have also presented results to study how the test time varies with the number of IC layers, core placement, and the upper limit on the number of TSVs. This work is expected to pave

Table 9
Test time comparison between Test Bus architecture and TestRail architecture (B=2, 3, 4, two layers, TSV limits 48/60).

W	B	Test Bus 2D test time (#CC)	TestRail 2D Test time (#CC)	Test Bus TAM partition	Test Bus test time (#CC)	TestRail TAM partition	TestRail test time (#CC)
16	2	44 225	33 437	(7,9)	24 257	(4,12)	14 873
	3	46 108	33 437	(4,5,7)	22 984	(1,3,12)	14 873
	4	51 707	33 782	(1,1,4,10)	27 403	(1,1,2,12)	14 873
24	2	31 070	17 750	(5,19)	21 953	(2,22)	7881
	3	31 077	17 750	(1,4,19)	20 764	(4,3,17)	7881
	4	41 611	17 750	(1,3,4,16)	16 549	(4,1,2,17)	7881
32	2	28 524	17 750	(11,21)	20 934	(8,22)	7881
	3	23 944	17 750	(4,10,18)	16 085	(3,7,22)	7881
	4	33 782	17 750	(2,4,10,16)	13 505	(3,4,7,18)	7881
40	2	25 761	17 750	(19,21)	18 604	(12,22)	7881
	3	21 070	17 750	(4,17,19)	13 095	(2,12,22)	7881
	4	26 961	17 750	(4,4,13,19)	11 282	(2,8,12,18)	7881
48	2	25 761	17 750	(21,27)	17 331	(3,24)	7881
	3	21 070	17 750	(7,19,22)	11 808	(12,14,22)	7881
	4	25 086	17 750	(3,4,7,34)	9522	(4,10,12,22)	7881
56	2	23 256	17 750	(23,33)	15 392	(16,32)	7881
	3	21 070	17 750	(4,19,33)	11 808	(15,17,24)	7881
	4	24 852	17 750	(2,4,16,34)	8452	(4,8,20,24)	7881
64	2	23 232	17 750	(23,41)	13 984	(16,40)	7881
	3	21 070	17 750	(5,22,37)	9731	(4,18,24)	7881
	4	24 852	17 750	(13,16,16,19)	7923	(3,9,16,24)	7881

Table 10
Test time and wire length results for SOC d695 with different B (2 layers, TSV limits 48/60).

B	TAM partition	Test time (#CC)			Wire length (μm)		
		A=0	A=0.5	A=1	A=0	A=0.5	A=1
2	8+16	74 224	69 619	69 619	223 200	223 200	505 600
3	8+16+32	69 683	69 619	69 619	512 800	512 800	1 025 600
4	8+12+16+20	74 224	69 619	69 619	511 600	511 600	874 000
5	3+5+17+19+23	175 861	66 024	66 024	610 500	610 500	711 300
6	4+7+16+17+28+31	131 122	49 248	49 248	938 400	938 400	1 109 900
7	5+7+11+24+27+34+37	105 895	46 952	46 952	1 320 700	1 320 700	1 350 400

the way for core-based testing of emerging 3D SOCs. As part of ongoing work, we are studying post-bond testing of 3D ICs designed using hierarchical cores [40] and the optimization of a pre-bond test infrastructure.

References

[1] Y. Zorian, E.J. Marinissen, S. Dey, Testing embedded-core-based system chips, Computer 32 (6) (1999) 52–60 0018-9162..
[2] IEEE Std. 1500: IEEE Standard Testability Method for Embedded Core-based Integrated Circuits. IEEE, New York, 2005.
[3] V. Iyengar, K. Chakrabarty, E.J. Marinissen, Test wrapper and test access mechanism co-optimization for system-on-chip, Journal of Electronic Testing: Theory and Applications 18 (2002) 213–230.
[4] S. K. Goel, E.J. Marinissen, Effective and efficient test architecture design for SOCs, in: International Test Conference, 2002, pp. 529–538.
[5] Q. Xu, N. Nicolici, Resource-constrained system-on-a-chip test: a survey, IEE Proceedings Computers and Digital Techniques 152 (1) (2005) 67–81 1350-2387.
[6] E. Larsson, H. Fujiwara, Optimal system-on-chip test scheduling, in: Asian Test Symposium, 2003, pp. 306–311.
[7] E. Larsson, K. Arvidsson, H. Fujiwara, Z. Peng, Efficient test solutions for core-based designs, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 23 (5) (2004) 758–775 0278-0070.
[8] Q. Xu, N. Nicolici, Modular SOC testing with reduced wrapper count, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 24 (12) (2005) 1894–1908 0278-0070.
[9] Z. Wei, S.M. Reddy, I. Pomeranz, Y. Huang, SOC test scheduling using simulated annealing, in: VLSI Test Symposium, 2003, pp. 325–330.
[10] Y. Huang, S.M. Reddy, W. Cheng, P. Reuter, N. Mukherjee, C.Tsai, O. Samman, Y. Zaidan, Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm, in: International Test Conference, 2002, pp. 74–82.
[11] T. Yoneda, M. Imanishi, H. Fujiwara, An SoC test scheduling algorithm using reconfigurable union wrappers, in: DATE, 2007, pp. 231–236.
[12] R. Weerasekera, L. Zheng, D. Pamunuwa, A. Hannu Tenhunen, Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs, in: ICCAD, 2007, pp. 212–219.
[13] B. Black, D.W. Nelson, C. Webb, N. Samra, 3D processing technology and its impact on iA32 microprocessors, in: ICCD, 2004, pp. 316–318.
[14] J. Cong, J. Wei, Y. Zhang, A thermal-driven floorplanning algorithm for 3D ICs, in: International Conference on Computer Aided Design, 2004, pp. 306–313.
[15] W.L. Hung, G.M. Link, Y. Xie, V. Narayanan, M.J. Irwin, Interconnect and thermal-aware floorplanning for 3D microprocessors, in: International Symposium on Quality Electronic Design, 2006, pp. 98–104.
[16] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, P.D. Franzon, Demystifying 3D ICs: the pros and cons of going vertical, IEEE Design and Test of Computers 22 (6) (2005) 498–510.
[17] Y. Xie, G.H. Loh, B. Black, K. Bernstein, Design space exploration for 3D architectures, Journal on Emerging Technologies in Computing Systems 2 (2) (2006) 65–103.
[18] J.W. Joyner, P. Zarkesh-Ha, J.D. Meindl, Global interconnect design in a three-dimensional system-on-a-chip, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 12 (4) (2004) 367–372 1063-8210.
[19] M.R. Gary, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, 1979.
[20] P. Raghavan, C. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs, Combinatorica 7 (4) (1987) 365–374.
[21] V. Immaneni, S. Raman, Direct access test scheme-design of block and core cells for embedded ASICs, in: International Test Conference, 1990, pp. 488–492.
[22] N.A. Touba, B. Pouya, Using partial isolation rings to test core-based designs, IEEE Design & Test of Computers 14 (4) (1997) 52–59 0740-7475.

[23] I. Ghosh, S. Dey, N.K. Jha, A fast and low cost testing technique for core-based system-on-chip, in: Design Automation Conference, 1998, pp. 542–547.
[24] P. Varma, B. Bhatia, A structured test re-use methodology for core-based system chips, in: International Test Conference, 1998, pp. 294–302.
[25] E.J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M.Lousberg, C. Wouters, A structured and scalable mechanism for test access to embedded reusable cores, in: International Test Conference, October 1998, pp. 284–293.
[26] S.K. Goel, E.J. Marinissen, Cluster-based test architecture design for system-on-chip, in: VLSI Test Symposium, 2002, pp. 259–264.
[27] K. Chakrabarty, Optimal test access architectures for system-on-a-chip, ACM Transactions on Design Automation of Electronic Systems 6 (2001) 26–49.
[28] V. Iyengar, K. Chakrabarty, E.J. Marinissen, Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip, IEEE Transactions on Computers 52 (2003) 1619–1632.
[29] S.K. Goel, E.J. Marinissen, SOC test architecture design for efficient utilization of test bandwidth, ACM Transactions on Design Automation of Electronic Systems 8 (4) (2003) 399–429.
[30] Q. Xu, N. Nicolici, Multi-frequency test access mechanism design for modular SOC testing, in: Asian Test Symposium, 2004, pp. 2–7.
[31] K. Chakrabarty, Design of system-on-a-chip test access architectures under place-and-route and power constraints, in: Design Automation Conference, 2000, pp. 432–437.
[32] S.K. Goel, E.J. Marinissen, Layout-driven SOC test architecture design for test time and wire length minimization, in: Design, Automation and Test in Europe Conference, 2003, pp. 738–743.
[33] H.-H.S. Lee, K. Chakrabarty, Test challenges for 3D integrated circuits, IEEE Design & Test of Computers 26 (5) (2009) 26–35.
[34] D.L. Lewis, H.S. Lee, A scan-island based design enabling pre-bond testability in die-stacked microprocessors, in: Proceedings of the International Test Conference, 2007, pp. 1–8.
[35] X. Wu, P. Falkenstern, K. Chakrabarty, Y. Xie, Scan-chain design and optimization for three-dimensional integrated circuits, Journal on Emerging Technologies in Computing Systems 5 (2) (2009) 1–26.
[36] L. Huang L. Jiang, Q. Xu, Test architecture design and optimization for three-dimensional SoCs, in: DATE, 2009, pp. 220–225.
[37] P.-Y. Chen, C.-W. Wu, On-chip TSV testing for 3D IC before bonding using sense amplification, in: Asian Test Symposium, 2009, pp. 450–455.
[38] L. Jiang, Q. Xu, K.Chakrabarty, T.M. Mak, Layout driven test-architecture design and optimization for 3D SoCs under pre-bond testpin-count constraint, in: IEEE International Conference on Computer-Aided Design, 2009, p. 191–196.
[39] K. Chakrabarty, V. Iyengar, M. Krasniewski, Test planning for modular testing of hierarchical SOCs, IEEE Transactions on Computer Aided-Design of Integrated Circuits & Systems 24 (2005) 435–448.
[40] S. Goel, E.J. Marinissen, A. Sehgal, K. Chakrabarty, Testing of SoCs with hierarchical cores: common fallacies, test access optimization and test scheduling, IEEE Transactions on Computers 58 (3) (2009) 409–423.
[41] E.J. Marinissen, S.K. Goel, M. Lousberg, Wrapper design for embedded core test, in: International Test Conference, 2000, pp. 911–920.
[42] D. Bertsimas, J. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, 1997.
[43] S. Samii, E. Larsson, K. Chakrabarty, Z. Peng, Cycle-accurate test power modeling and its application to SoC test scheduling, in: ITC, 2006, pp. 1–10.
[44] Xpress-MP. 〈www.dashoptimization.com〉, 2007.