Multiobjective Optimization of Deadspace, a Critical Resource for 3D-IC Integration

Johann Knechtel*, Igor L. Markov**, Jens Lienig*, and Matthias Thiele*
* Institute of Electromechanical and Electronic Design, Dresden University of Technology, Dresden, Germany
** Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA
johann.knechtel@ifte.de, imarkov@eecs.umich.edu, jens.lienig@ifte.de, matthias.thiele@ifte.de

***Abstract*—In 3D-IC integration and its implied resource optimization, a particularly critical resource is *deadspace* — regions *between* floorplan blocks. Deadspace is required for through-silicon via (TSV) planning and other related design tasks, but the effective use of this limited and highly-contested resource requires effort. While most previous work focuses on a single design issue at a time, we propose a lightweight *multiobjective deadspace-optimization methodology* that simultaneously optimizes interconnect, IR-drop, clock-tree size and maximal temperature. This methodology repeatedly re-evaluates design quality during early chip planning and uses resulting information to guide further optimization. Experimental results indicate that constructing an appropriate deadspace distribution improves design tradeoffs and is effective in practice.**

I. Introduction

Three-dimensional (3D)-IC integration is an increasingly attractive design option to balance the requirements of functionality, performance, and cost of ICs. *Chip-level integration* (Figure 1) is facilitated by *through-silicon vias (TSVs)* and promises shorter and lower-power interconnects compared to traditional wire-bonded systems [4]. This type of integration can also increase yield through separate die testing [24] and support heterogeneous dies [5]. These benefits not only play a major role in business decisions [10], but also favor a coarser integration where large circuit blocks are laid out on individual dies. Such *block-level integration* facilitates the use of conventional 2D intellectual property (IP) blocks in 3D assemblies without changing their original layouts [20].

In this context, both 2D and 3D block-level integration must account for *deadspace* between blocks, i.e., on-chip regions not occupied by floorplan blocks.[1] Traditionally, area (with deadspace as a proxy) and wirelength have been the key objectives for floorplanning and thus subject to minimization. However, deadspace is required for a multitude of subsequent chip-design tasks. For 3D ICs, deadspace is essential for TSV insertion. In 2D (and 3D) design, it may be required for power delivery, global interconnect (bus) routing, as well as the insertion of decaps and glue logic [17, Chapter 3]. Deadspace optimization seeks to improve block and TSV placement such that TSV overhead is diminished while accounting for design constraints (e.g., TSV placement between blocks) and

[1]We differentiate *deadspace* from *whitespace* as follows. Deadspace is used during floorplanning; whitespace is used during placement and refers to locally unoccupied space that is distributed among cells. Whitespace is used to facilitate routing, gate sizing, net buffering and detail placement [2], [6]. Due to its late and highly-local allocation, whitespace is not suitable for global design tasks like TSV planning—deadspace is required for such tasks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 2012, November 5-8, 2012, San Jose, California, USA
Copyright © 2012 ACM 978-1-4503-1573-9/12/11 ...$15.00.

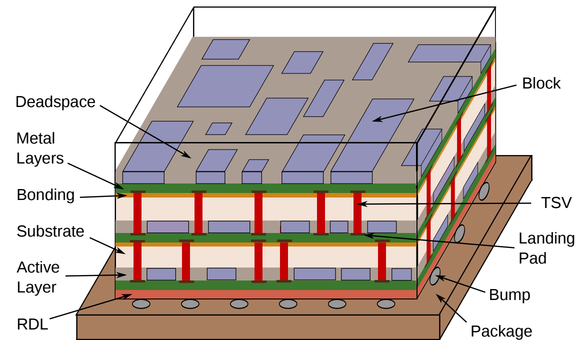

Fig. 1. A 3D IC with three dies, stacked using face-to-back technology. To enhance clarity, the top substrate layer and the heatsink atop are not illustrated; chip fronts are cut. TSVs must not obstruct blocks and are thus placed in the deadspace between them. Note that some TSVs in adjacent dies are aligned.

for optimization goals (e.g., reducing IR-drop by inserting additional power/ground TSVs). Such a *multiobjective deadspace optimization* is challenging—focusing on one particular objective may undermine the remaining objectives.

Previous work mostly limits deadspace optimization for 3D ICs to meet one or a few objectives. One study considers deadspace redistribution for thermal-TSV insertion [26]. Other studies propose deadspace insertion/distribution during floorplanning to facilitate subsequent insertion of signal and/or thermal TSV [9], [27]. He et al. [12] consider deadspace redistribution for buffer and signal-TSV insertion. In contrast, we focus on meeting multiple objectives.

In this paper, we make the following contributions.

- We identify the **major deadspace-distribution requirements**, essential for addressing key challenges of 3D-IC integration during early design phases (Section II).
- We develop a **first-of-a-kind multiobjective methodology for deadspace optimization** in 3D ICs, called *MoDo* (Section III). To illustrate the modularity of our approach, we construct a design-flow extension using our proposed algorithms and available design tools.

The remainder of this paper is structured as follows. We first review the major challenges for 3D-IC design, discuss related work and derive the resulting deadspace requirements in Section II. We then motivate multiobjective deadspace optimization. Our optimization methodology *MoDo* is presented in Section III; an experimental investigation is provided in Section IV. Our conclusions on optimizing deadspace for 3D ICs and its benefits are given in Section V.

II. Challenges and Opportunities in 3D-IC Design

Early physical-design phases of 3D-IC integration are driven by floorplanning and TSV placement. Both stages are typically implemented to address key challenges in 3D-IC design reviewed below. These stages are also responsible for regulating the amount and distribution of deadspace.

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

TABLE I
PROPERTIES OF DIFFERENT TSV TYPES AND RELATED DEADSPACE-DISTRIBUTION REQUIREMENTS

Type	Diameter	Alignment	Local Density	Preferred Placement	Deadspace-Distribution Requirements
Signal	$\approx 2-20\mu m$	encouraged for specific applications (e.g., buses crossing multiple dies [25])	low - high	irregular [19], [20]	nonuniform; small - large contiguous regions; rarely aligned
Thermal	$\approx 2-40\mu m$	may be encouraged [7]	low - medium	irregular [8], [9], [27]	nonuniform; small - medium contigous regions; possibly aligned
Power/Ground	$\approx 10-40\mu m$	strongly preferred [7], [13], [14], [16]	low	irregular [11], [13], [14], [16]	nonuniform; small contigous regions; necessarily aligned
Clock	$\approx 2-20\mu m$	may be encouraged [33]	low	irregular [33], [34]	nonuniform; small contigous regions; possibly aligned

An important concern is that the same amount of deadspace can be distributed throughout the die in many different ways. Depending on floorplanning objectives, blocks and thus deadspace are typically distributed to reduce wirelength and facilitate thermal management [9], [27], [35]. However, blocks can be redistributed later to address other design concerns. Such floorplan modifications can, for example, be implemented using the notion of *spatial slacks* [1]. Redistributing blocks and deadspace is essential for placement of different TSV types and related optimization goals.

A. TSV Types and 3D-IC Integration

Table I contrasts properties of different TSV types and outlines resulting requirements for deadspace distribution. For example, power/ground TSVs are preferably aligned to limit electromigration, IR-drop and routing congestion. Signal TSVs may be grouped into TSV islands [15], [20] to enhance fault tolerance where contiguous deadspace is available. TSVs of all types form placement obstacles since they occupy at least the device layer. As illustrated in Figure 1, block-level integration requires TSVs to be placed between blocks. Note that TSVs are not expected to scale as well as transistors [29]. Therefore, TSV overhead must be limited, which favors block-level integration over gate-level integration [20].

Depending on the die-stacking technique, TSVs may require aligned deadspace regions on adjacent dies—we refer to this as the *deadspace-alignment problem*. For back-to-back (B2B) die stacking, this applies to each TSV since they are passing through adjacent substrate layers. For face-to-back (F2B) stacking, alignment should be considered according to Table I.

B. Thermal Management

Unlike 2D designs, 3D designs exhibit higher *packing density* and therefore higher *power density*. Sophisticated thermal management techniques have been developed to address potential problems [30]. Common techniques include (i) thermal-aware block placement to spread high-power blocks and (ii) insertion of thermal TSVs (or recently microfluidic channels) in order to increase the vertical (or horizontal) thermal conductivity of a 3D IC. For example, Zhou et al. [35] propose an force-directed floorplanner using technique (i) while simultaneously optimizing wirelength, area and thermal distribution. Cong et al. [8] propose irregular TSV placement and are able to provide significantly better temperature reduction compared to uniform placement. Their technique is motivated by their following result. The maximal temperature on the whole 3D IC can be minimized if, for each die, the TSV area in any given (2D) bin is proportional to the lumped power consumption of this and all overlapping bins from dies underneath.

The techniques discussed so far tend to overlook other important objectives such as IR-drop in their formulations, and may lead to significant deterioration in these objectives.

C. Power/Ground and Clock Networks

In addition to thermal management, the high packing density of a 3D design also affects power and clock-signal delivery. Power delivery must provide sufficient current to each module and reduce IR-drop, i.e., the DC voltage drop during normal operation. This drop is the dominant cause of power-noise issues in 3D ICs. However, for large stacks, the TSV inductance which impacts transient noise should also be considered [13].

Clock networks must ensure small skew while satisfying slew constraints and minimizing power consumption. These networks are characterized by large capacitive loads and high-frequency switching. This requires a large amount of power, possibly up to 50% of total power consumption [34].

Some recent work (Table I) proposes to use *aligned TSV stacks* which span multiple dies. Such stacks for power/ground (PG) (or clock) TSVs must be carefully coordinated. First, this requires deadspace alignment. Second, these stacks obstruct many enclosed routing tracks—connecting the TSV landing pads requires multiple vias in all metal layers to enable proper power (or clock) delivery.

Prior work [13], [14] suggests that a *distributed topology* for PG TSVs is superior to both single, large TSVs and groups of clustered TSVs. These and other studies (Table I) also favor *irregular TSV placement*, in particular such that regions drawing significant current exhibit a higher TSV density. Irregular placement allows one to reduce TSV count compared to uniform placement. These guidelines are particularly helpful in block-level 3D-IC integration.

For clock-network design, a straightforward approach is to place a single TSV in each die to interconnect the network. However, Zhao et al. [33], [34] show that multiple TSVs help reduce power consumption, wirelength and clock skew.

D. Routing

Note that TSVs obstruct routing in 3D ICs [18]. Accounting for signal, thermal, PG and clock networks and required TSVs poses a major challenge in routing. In this context, Lee and Lim [23] propose a methodology to co-optimize routing, thermal distribution and power-supply noise. However, they ignore clock networks.

As indicated in Table I, irregular placement is preferred for all TSV types and requires several nonuniform deadspace regions. Given such a spread-out TSV placement, local routing congestions may be limited due to medium local TSV densities. This particularly applies to block-level integration, where only a limited number of global nets need signal TSVs [31].

E. Research Opportunities in 3D-IC Design and Optimization

While previous work succeeds in addressing individual challenges for 3D-IC integration, a unified approach to address major requirements and provide design-quality analysis remains a key challenge. The closest prior work is presented by Lee and Lim [23]. However,

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705–712, San Jose, CA, November 2012.

they consider only gate-level integration and ignore clock networks. Their deadspace optimization is focused on thermal TSVs only.

In previous subsections, we outlined how prior work in block-level 3D-IC integration has been relying on specific deadspace-distribution characteristics. In case these requirements are not satisfied, several authors propose to redistribute deadspace [9], [12], [20], [25], [26]. However, prior work mainly focuses on single-objective deadspace optimization, which may undermine overall design quality. In contrast, multi-objective optimization offers a greater promise in this context, as confirmed by our experiments (Section IV). Such optimization requires understanding of the impact of different TSV-planning phases on design quality, as well as techniques for multiobjective deadspace optimization.

Recall that multiobjective deadspace optimization seeks to improve block and TSV placement in order to diminish TSV overhead and account for *multiple* design constraints and optimization goals. Such an optimization process can be successfully implemented during early design phases, as described in the remainder of our work.

III. MoDo: A Methodology for Multiobjective Deadspace Optimization

3D-IC design is challenging in many aspects. In particular, deadspace optimization at early design phases is necessary to ensure design closure. In order to enable multiobjective deadspace optimization, we propose a modular methodology which can guide existing 3D-IC design flows and provide feedback to specific design steps. We construct a design-flow extension using our algorithms and available 3D design tools. The approach is modular and can accommodate other tools or stages.

Our proposed design-flow extension is illustrated in Figure 2; it is based on an incremental process aiming for a deadspace-optimized floorplan satisfying multiple design criteria. As is typical in modular 3D-IC design flows, TSV planning can be separated from the floorplanning and/or placement stages. Thus, the *main loop* encapsulating TSV planning and deadspace optimization seeks to (i) determine appropriate TSV sites, likely requiring deadspace redistribution and/or alignment, (ii) place a TSV into or near the site, and (iii) perform deadspace optimization considering (updated) TSV sites. To guide TSV planning, related quality-analysis metrics are evaluated during iterations. After the main loop has converged, overall design quality is evaluated, possibly restarting global optimization (*global loop*). Our algorithms and methodology are presented next.

A. Methodology Configuration

Given a 3D-IC design, we perform the following methodology-configuration steps. **First**, an initial 3D floorplan is obtained (Subsection IV-A). This floorplan provides the inter-die block partitioning and (preliminary) block locations. **Second**, a die ordering to improve the thermal distribution and to minimize the TSV count is performed. Given $|\mathcal{D}|$ dies, we analyse all $|\mathcal{D}|!$ possible die sequences. For each sequence, we estimate the power distribution and the signal-TSV count. The sequence with the lowest cost $\Gamma_{seq} = w_{seq} * \gamma_{P,norm} + (1 - w_{seq}) * \gamma_{TSV,norm}$ is chosen where $\gamma_{P,norm}$ and $\gamma_{TSV,norm}$ denote to $[0,1]$ normalized stack-order-weighted power distribution and TSV count, respectively.

B. Deadspace Optimization

Note that the main loop including TSV planning (Subsection III-C) and deadspace optimization is a key part of MoDo. Thereby, TSV planning seeks to guide deadspace optimization and thus to address the following concerns.

- **Managing deadspace utilization** — regulating the TSV count and determine TSV sites. Given that different TSVs of different types compete for available deadspace, managing the utilization directly impacts design quality.
- **Accounting for deadspace-distribution requirements** (Section II) eases TSV placement. Once TSV sites are determined, they are considered as rectangular blocks, occupying some amount of deadspace. This resource accounting is convenient during subsequent deadspace optimization.
- **Tackling the deadspace-alignment problem**, i.e., aligning deadspace regions to place aligned TSVs. To ease placement of all TSVs, those to be aligned should be considered first.

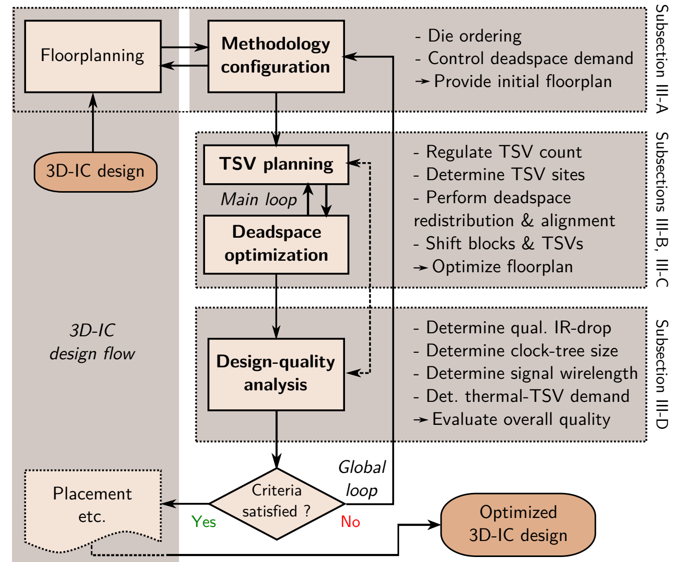

Fig. 2. Main parts of MoDo, that can be integrated in a 3D-IC design flow at early design phases.

Addressing these issues allows us to improve TSV and block placement while exploiting given deadspace. For that purpose, we invoke deadspace redistribution and alignment as well as shifting of blocks and TSVs. We limit deadspace insertion because it can increase area and wirelength overhead [20]. However, when design quality is judged unacceptable, the amount of deadspace must increase to ease deadspace optimization and TSV insertion. By doing so, we intend to reach the desired design quality during global-loop iteration(s).

We consider planned TSV sites as *movable blocks*. This allows us to place TSVs into nearby deadspace in cases where determined sites overlap with design blocks. In fact, we allow design blocks themselves to be shifted as well; this enables deadspace redistribution and alignment, therefore strict TSV-placement requirements can be also satisfied. Note that we have to perform shifting of both blocks and TSVs such that (i) a valid placement can be assured and (ii) the desired design quality is only marginally affected.

To address both issues, we base our shifting algorithm on the concepts of *constraint graphs (CGs)* [17, Chapter 3], *range constraints* [32] and *spatial slacks* [1] in floorplanning. Representing a floorplan using a CG pair (horizontal and vertical graph) allows us to maintain a valid placement and to handle the relations between block positions efficiently. Spatial slacks describe maximal possible shifting ranges of blocks within the given floorplan outline, whereas range constraints are used to limit shifting within certain regions.

In our incremental flow, we initially generate the CG pair for each die separately, considering placed blocks, and update them during TSV planning. Furthermore, we transform block and TSV coordinates (x, y) into range constraints $[x - \delta, x + \delta], [y - \delta, y + \delta]$, defining different *shifting windows* (Figure 3a). In order to judge the feasibility

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

Fig. 3. Handling of range constraints. (a) Block and TSV boundaries are extended by range constraints, defining shifting windows (dotted rectangles). Note that shifting windows of TSVs originate from their rectangular keep-out zone (KOZ). (b) The related horizontal CG with annotated slacks. During slack determination, window dimensions are considered as limits. Affected slack values are underlined.

of considered sites during TSV planning, i.e., the capabilities for required block shifting, we determine and annotate slacks to the CGs. Note that we determine slacks for both possible shifting directions to account for non-packed floorplans. During slack determination, shifting windows are considered as limits, as illustrated in Figure 3b.

Updating the CGs is trivial if a planned TSV site occurs within deadspace. For cases where the TSV site overlaps one or several existing blocks, the respective slacks have to considered. To furthermore minimize required shifting, TSV insertion takes place next to the nearest border of the overlapped blocks. Our proposed algorithm is outlined in Figure 4.

To enable our proposed shifting flow, we furthermore implement simple algorithms to determine slacks and to transform CGs to floorplans and vice versa [17, Chapter 3]—the shifting algorithm results from transforming the proposed extended CGs to an floorplan. Note that accounting for shifting windows requires to generate non-packed floorplans. Therefore, additional CG edges have to be inserted such that blocks and TSVs are limited to their windows.

Note that coordinates of placed blocks and TSVs represent their original locations, determined by initial floorplanning and TSV planning, respectively. This prevents blocks and TSVs from being shifted too far from their intended locations. Defining appropriate values for $\pm\delta$, i.e., sizing the shifting window, allows us to limit the impact of shifting on design quality. In order to support aligned TSVs placed on several dies, we set their locations to identical coordinates and set $\delta = 0\mu m$.

C. TSV Planning

We order TSV types as follows to facilitate deadspace optimization: 1) PG TSVs, 2) clock TSVs, 3) signal TSVs, and 4) thermal TSVs. The rationale for this ordering is discussed next. **First**, PG TSVs should be aligned throughout the 3D-IC stack and are thus given priority. Clock TSVs may also be aligned, but not necessarily throughout the whole stack. **Second**, critical PG and clock networks should be planned early. **Third**, signal-TSV planning adheres some degree of freedom for site determination (Subsection III-C3); previously placed TSVs are not expected to significantly obstruct it. **Fourth**, all placed TSVs may be exploited as thermal TSVs and thus facilitate thermal management. Nevertheless, additional thermal TSVs may be warranted in practice (Subsection III-C4).

For each TSV-planning phase, our site-accounting allows for the initial value Γ_γ^0 of the corresponding cost γ (e.g., maximal IR-drop $\Gamma_{\gamma_{IR}}^0$) to be reduced to $\Gamma_\gamma^{opt} = w_\gamma^{opt} * \Gamma_\gamma^0$. In the following subsections, we describe our techniques for TSV planning in detail.

```
UPDATE_CG(TSV toInsert, FP floorplan)
 1  // determine overlapped blocks/TSVs
 2  Block[] overlap = DETERMINE_OVERLAP(toInsert, floorplan)
 3  // trivial case, insert into deadspace
 4  if (overlap == NULL)
 5      // consider existing nodes; e.g., toInsert lies between A, B:
 6      // adapt edge A→B to A→toInsert, add toInsert→B
 7      ADAPT_CG_LOCALSEARCH(toInsert, floorplan)
 8      // insert edge to retain non-packed location;
 9      // parameters: source, target, weight
10      INSERT_NEW_EDGE(globalSrc, toInsert, toInsert.lowerLeft)
11      // update slacks of affected blocks/TSVs
12      UPDATE_SLACKS_LOCALSEARCH(toInsert)
13      // feedback (for TSV planning); insertion is feasible
14      return TRUE
15  // nontrivial case, some blocks overlap; CG adaption for each block
16  else
17      foreach Block b = overlap[b']
18          // determine the nearest border of block/TSV overlap;
19          // consider available slacks in both directions
20          Border insertBorder = DETERMINE_BORDER(toInsert, b)
21          // insufficient slack; insertion is infeasible
22          if (insertBorder == NULL)
23              return FALSE
24          else
25              // sufficient slack; update CG according to insertBorder
26              // and the relative ordering of toInsert and b
27              ADAPT_CG_BORDER(toInsert, b, insertBorder)
28              UPDATE_SLACKS_LOCALSEARCH(b)
29      return TRUE
```

Fig. 4. Our algorithm to update the CGs during TSV planning. Note that invoked procedures consider the graph type, i.e., horizontal or vertical CG.

1) Power/Ground-TSV Planning: Placing irregularly distributed TSVs stacks in high-power regions is most useful for limiting IR-drop and TSV count (Subsection II-C). We consider PG-grid structures, illustrated in Figure 5a. (Structural properties are described in Subsection IV-A2). In order to determine PG-TSV sites, i.e., grid nodes, previous work mostly considers modified nodal analysis of a resistive equivalent circuit. However, scaling such network analysis to large designs, resulting in possibly millions of network nodes, is difficult. We therefore propose a simplified diagnostic—the *qualitative IR-drop distribution*. Our approach is based on the following observations, made while performing SPICE-based simulations for the IR-drop on abstracted, resistive PG grids (Figure 5b). (A simulation result is illustrated in Figure 5c.) **First**, we observe that aligned PG TSVs influence IR-drop in the circumference of that TSV site on all dies. **Second**, the IR-drop caused by modules is distributed less evenly than the drop caused by TSVs and grid wires. **Third**, the total IR-drop distribution can be interpreted as the superposition of separate distributions originating from power-consuming modules. Each distribution can be described using an exponential function while considering nearby power consumption and PG TSVs.

Applying these observations, we determine the qualitative IR-drop distribution as follows. **First**, we construct 2D grids similar to the 3D-IC PG grids. **Second**, we sum up power consumption of modules which overlap in the 3D-IC stack, and assign normalized values $P(n)$ to related nodes n in the 2D grids. **Third,** given such 2D lumped-power grids, we determine for each node n (with no TSV assigned yet) four *power-spreading factors*, one for each cardinal direction. Each factor a_{left}, a_{right}, a_{top} and a_{bottom} is calculated as $a = -\ln(a_{min})/d_{max}$ where a_{min} is the given minimal IR-drop factor (Table II) to be reached at distance d_{max} away from n; d_{max} is determined by following the respective grid direction until the nearest TSV (or die boundary) is reached. **Fourth**, we determine the superposition of power spreading on all nodes, representing the qualitative IR-drop IR'. For a particular node n, we consider

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

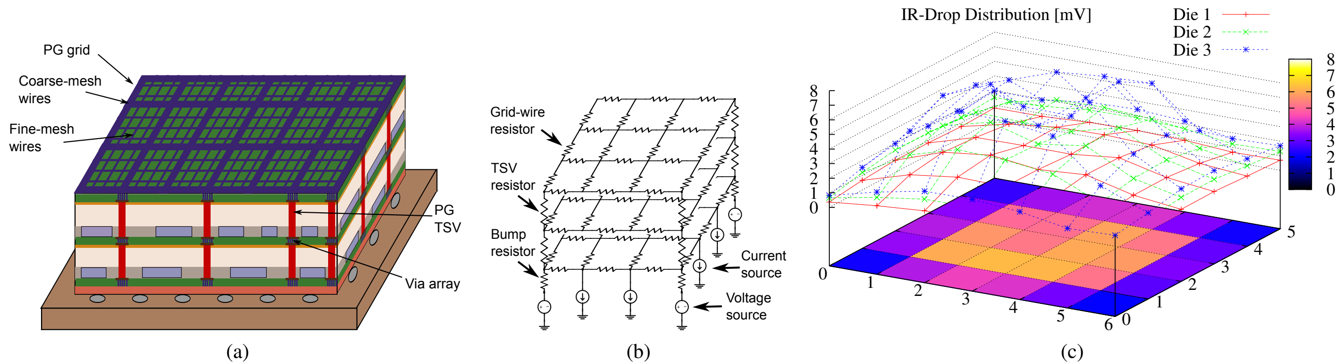

Fig. 5. PG grids. (a) Grid structures in a 3D IC; wires are only illustrated on the uppermost die. Depending on power-distribution requirements, TSVs may be required only for some grid nodes. Note that PG TSVs are aligned and connected through metal layers using via arrays. (b) Resistive grid model. Current sources are connected to each node (only some illustrated), representing power consumption of modules. PG TSVs and grid wires are represented by resistors for IR-drop modelling. (c) SPICE simulation of the IR-drop for benchmark *n300*. Power TSVs placed on the outer grid ring connect the separate dies, resulting in locally reduced IR-drop. Note the additional local minima in the center region, resulting from further TSVs.

itself and other nodes $n' \neq n$ in the same quadrant and determine $IR'(n) = s_{TSV} * \left(P(n) + \sum_{n',a} P(n') * \exp(-a * \text{dist}(n, n')) \right)$ where s_{TSV} is a scaling factor (Table II) applied for nodes with a TSV assigned. Note that only the two relevant factors a are considered, i.e., the ones pointing towards n.

Employing the diagnostic of qualitative IR-drop distribution, we perform PG-TSV planning as follows. **First**, we consider the largest-value node as TSV site. **Second**, we perform deadspace optimization. **Third**, we redetermine the qualitative IR-drop distribution. In cases where desired cost $\Gamma^{opt}_{\gamma_{IR}}$, i.e., reduction of initially largest IR-drop, is not reached, we continue with the first step.

2) Clock-TSV Planning: Using multiple clock TSVs helps to reduce power consumption due to wirelength reduction; a single TSV enforces large global trees on each die, whereas multiple TSVs enable several smaller local trees [34] (Figure 6). To facilitate clock-tree synthesis and appropriate TSV count, we propose the following TSV-planning algorithm. **First**, for each die (except the uppermost) *k-means++ clustering* [3] of clock sinks is performed in order to determine TSV sites and accomplish TSV assignment. The cluster count k is stepwise increased until desired cost $\Gamma^{opt}_{\gamma_{CP}}$ can be reached, that is the reduction of initially estimated wirelength using only one cluster. The cost term is defined as $\Gamma_{\gamma_{CP}}(k) = \sum_{c \in C} \max\left(\text{dist}(c.center, sink \in c)\right) * |sink \in c|$, that is the sum over all cluster of the maximal distance between the cluster center and any assigned sink, multiplied by the sink-cluster-assignment count. We also refer to this term as *weighted clock-tree size*. Its purpose is to model the expected change in wirelength of balanced clock-trees during clustering. Clustering cannot account for clock-

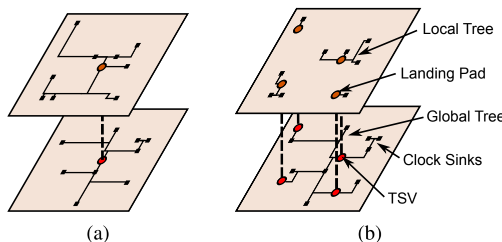

Fig. 6. Different 3D-IC clock-tree structures and their impact on wirelength. The clock source is assumed to be connected to the lower die, TSVs interconnect the separate trees, which then interconnect clock sinks. (a) Using single TSVs enforces large global trees on each die. (b) In contrast, using multiple TSVs enables several local trees, reducing the total wirelength.

network parameters such as clock skew, but subsequent (obstacle-aware) clock-tree synthesis optimizes them via buffer insertion and clock-tree tuning [21], [22]. If required, clock sinks may be even reassigned to TSVs or swap assignments with signal TSVs. **Second**, deadspace optimization is performed using determined cluster center as TSV sites. Thereby, the shifting windows are initially defined as $\delta_C = 0\mu m$ to fix the cluster centers. In case of infeasible TSV insertion, the value is adapted (Table II).

3) Signal-TSV Planning: We perform signal-TSV planning as follows. **First**, we determine for each net n its *projected net bounding box* bb^p_n, which encircles pins on all related dies. **Second**, we determine area and available deadspace covered by bb^p_n on each related die separately. For each die d, related nets (with pins on d) are then sorted in the ascending order of area and deadspace, thus prioritizing (partial) nets with small boxes and little available deadspace. **Third**, starting with the lowermost die of the stack, a TSV site is planned within deadspace of bb^p_n using a local search for each (sorted) net on each die. If the search fails due to insufficient deadspace, sites are placed into nearby deadspace such that the distance to a related net pin on the same die is minimal. The search accounts for dense packing of multiple, grouped TSVs. This allows to reduce keep-out zones (KOZs) without increasing stress-induced impact on logic blocks [28]. Note that for nets spanning more than two dies, TSV planning has to be performed on all but the uppermost die connected by the net.

Note that we define no cost term for signal-TSV planning, since their count is minimized by die ordering. However, we evaluate the impact of TSV packing on estimated wirelength and routing utilization (Subsection III-D) in our experiments.

4) Thermal-TSV Planning: Recall that die ordering (Subsection III-A) and aligned PG TSVs facilitate thermal management [7]. Nevertheless, we consider the insertion of additional thermal TSVs to further decrease maximal temperature. We leverage findings by Cong et al. [8] (Subsection II-B) for our approach. **Initially**, we construct 2D lumped-power grids (Subsection III-C1) for all ordered subsets $\{d_1\}, \{d_1, d_2\}, \ldots, \{d_1, \ldots, d_{|\mathcal{D}|}\}$ of accordingly gridded dies where d_1 denotes the bottom die. The following steps are then performed independently for each lumped-power grid g and its uppermost die d_{top}. **First**, we determine the TSV count $T_{curr}(b)$ for each bin b in d_{top}. **Second**, we determine the ratio $r = \sum_b T_{curr}(b) / \sum_b lp(b)$ of d_{top}'s total TSV count and g's total lumped power. **Third**, we determine the *desired TSV count* $T_{des}(b) = \lfloor 0.5 + r * lp(b) \rfloor$ for each b in d_{top}. **Fourth**, we plan sites for b if $T_{curr}(b) < T_{des}(b)$ using a local search, as proposed for

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

signal-TSV planning. Since this last step may impact r, we repeat all enumerated steps until cost $\Gamma_{\gamma_T}^{opt}$ can be reached or no further TSV can be inserted for any b due to $T_{curr}(b) = T_{des}(b)$ or lacking deadspace. The initial cost is defined as $\Gamma_{\gamma_T}^{0} = \sum_b T_{des}(b) - T_{curr}(b)$.

D. Design-Quality Analysis

Our methodology provisions for frequent estimation of design quality. We estimate quality during TSV planning and deadspace optimization to guide this incremental process appropriately. However, we also seek to evaluate the overall design quality after finishing the main loop and possibly reconfigure MoDo and start over with deadspace optimization if design costs are not sufficiently reduced. For example, in cases where our flow fails to reduce cost $\Gamma_{\gamma_{IR}}^{0}$ to $\Gamma_{\gamma_{IR}}^{opt} = w_{\gamma_{IR}}^{opt} * \Gamma_{\gamma_{IR}}^{0}$, design quality in terms of IR-drop reduction is not ensured. Thus, additional PG-TSV sites are required, and the floorplanner is reconfigured to increase deadspace. Other TSV-related cost terms are covered in Subsection III-C.

We estimate signal wirelength using the half-perimeter wirelength (HPWL) metric as follows. For each net n, its bounding box bb_n, encircling pins of related blocks and possibly a TSV, is determined on each related die d separately. Note that we also consider the TSV on the die below d if applicable, thus account for routing to the landing pads. The resulting HPWL is denoted as $\mathrm{HPWL}(bb_n, d)$. The overall wirelength estimate is then calculated as $WL = \sum_n \left(\sum_d \mathrm{HPWL}(bb_n, d) + h_d * (\max(d_n) - \min(d_n))\right)$, where h_d refers to the die thickness, $\max(d_n)$ to the uppermost die of respective net n, and $\min(d_n)$ to its lowermost die. In order to estimate the signal-routing utilization, we construct separate routing grids for each die using tiles with dimensions according to signal-TSV dimensions (Subsection IV-A2). Each (partial) net is assumed to be routed in L-shaped wires on the related grid(s); wire segments ws are mapped to the tiles rt they cover. The average utilization is then determined as $u = \sum_d \sum_{rt} |ws(rt)| * |rt|^{-1} * |d|^{-1}$.

IV. Experimental Validation

A. Configuration

1) Methodology Configuration: Parameters introduced in Section III are summarized in Table II along with their values. Initial 3D floorplans are obtained using an academic tool [35] which accounts for wirelength, area and thermal distribution. The tool is configured such that all three objectives are equally weighed.

2) 3D-IC Configuration and Benchmarks: We consider F2B stacking and via-first TSVs (Figure 1) with a diameter of $4\mu m$ and a square KOZ with dimensions of $8\mu m \times 8\mu m$. PG-TSVs are larger, with a diameter of $8\mu m$ and a KOZ of $12\mu m \times 12\mu m$. Signal and thermal TSVs are grouped as *TSV islands*, reducing individual KOZs to $6\mu m \times 6\mu m$. Dies are thinned down to $40\mu m$. Metal layers are $4\mu m$ and bonding layers are $2\mu m$ thick. Power and ground grids are offset by $12\mu m$. Note that die boundaries are extended by $24\mu m$ to enable PG-TSV rings. Coarse PG-grid wires are $8\mu m$ wide, $0.8\mu m$ thick, and their pitch is $80\mu m$; this pitch also applies to PG TSVs.

Experiments are conducted using representative GSRC benchmarks with the following modifications. External pins are represented by package bumps, thus nets linked to such pins must connect to the lowermost die. Each block is assumed to have multiple spread-out clock sinks, one placed at the block's center and four placed in the corners. Net pins are placed at the related block's center.

3) Experimental Configuration: Our experiments using MoDo validate its capabilities for multiobjective deadspace optimization. We independently decrease the different cost factors w_{γ}^{opt} in steps of 10%, in the range from 90% to 40%. Experiments sweep through the parameter space; best results are reported in Table III. Estimated cost reduction determined by final design-quality analysis is reported as $\downarrow$. This reduction typically correlates to $100\% - w_{\gamma}^{opt}$.

TABLE II
Parameters along with their values

Metric	Meaning	Value
w_{seq}	Cost factor for layer ordering	0.5
a_{min}	Minimal power-spreading factor (qualitative IR-drop distribution)	0.01
s_{TSV}	Power-scaling factor for nodes with TSVs (qualitative IR-drop distribution)	0.5
$w_{\gamma_{IR}}^{opt}$	Cost factor for IR-drop optimization	input value
$\Gamma_{\gamma_{IR}}^{0}$	IR-drop optimization cost term: init. largest qualitative IR-drop	depends on design
$w_{\gamma_{CP}}^{opt}$	Cost factor for clock-power optimization	input value
$\Gamma_{\gamma_{CP}}^{0}$	Clock-power optimization cost term: weighted clock-tree size (single TSV)	depends on design
$w_{\gamma_T}^{opt}$	Cost factor for thermal optimization	input value
$\Gamma_{\gamma_T}^{0}$	Thermal optimization cost term: initial thermal-TSV demand	depends on design
δ_{PG}	Shifting window for PG TSVs	$0\mu m$
δ_C	Shifting window for clock TSVs	$0/50\mu m$
δ_S	Shifting window for signal TSVs	relates to net bounding box bb_n
δ_T	Shifting window for thermal TSVs	relates to grid bin b
δ_b	Shifting window for blocks	$100\mu m$

Results are compared to the following settings, subsequently referred to as baseline cases. For IR-drop optimization, PG-TSVs are only placed on the rings. For clock-power optimization, single clusters define global TSV sites on each die. For signal-TSV planning, TSVs are not packed into groups. For thermal optimization, additional thermal TSVs are not considered for any die and (redundant) PG TSVs are not placed in the uppermost die.

If deadspace is insufficient to reach the desired cost reductions, our methodology requests the floorplanner to increase deadspace. Our experiments swept the range from 10% to 60% in 10% steps.

B. Results

Experimental results in Table III suggest several observations. **First**, our methodology enables a tangible increase of deadspace utilization; in all experiments, most of deadspace finds good use, with $< 5\%$ deadspace left in some cases. **Second**, multiple deadspace-distribution requirements can be satisfied during early chip-planning phases. However, the prospects for optimizing the deadspace distribution depend on initial floorplans. A large amount of available deadspace may be insufficient *per se* because the relative block ordering and thus available slacks are also important. **Third**, we note that the deadspace-alignment problem can be successfully addressed within our methodology by sizing shifting windows to $\delta = 0$. **Fourth**, the die count impacts optimization results. The best results are typically obtained for three-die integration. Using four dies (and greater total deadspace) may not be justified; different optimization steps require more TSVs to maintain quality, thus increasing overhead and cost while decreasing deadspace-optimization chances. Considering two dies typically results in decreased slacks, thus also limits the space for optimization. **Fifth**, the dimensions of shifting windows influence deadspace optimization. We observe that increasing δ_C above $50\mu m$ is counterproductive in terms of weighted clock-tree size reduction. Furthermore, the initial value of $\delta_b = 50\mu m$ resulted in worse cost reductions, mainly for IR-drop reduction. However, increasing the window dimension above $100\mu m$ was not beneficial either.

Based on experimental results, we also made the following general observations on 3D-IC integration of the GSRC benchmarks. **First**, the signal-wirelength reduction due to TSV packing scales with the amount of interconnect, as expected. Interestingly, the average signal-routing utilization is reduced for TSV-packing setups; this is possibly due to the increased flexibility for TSV-group insertion and resulting

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

TABLE III
MODO RESULTS FOR GSRC BENCHMARKS

Metric	2 Dies			3 Dies			4 Dies		
	n100	*n200*	*n300*	*n100*	*n200*	*n300*	*n100*	*n200*	*n300*
Init. largest qualitative IR-drop	0.2605	0.2492	0.4822	0.2333	0.2502	0.4108	0.2376	0.2196	0.4037
Qualitative IR-drop ↓ (%)	40.23	30.85	30.21	34.27	51.64	45.46	37.98	38.03	42.30
IR-drop red. det. by SPICE sim. (%)	52.49	41.57	39.80	39.45	49.44	55.15	32.50	31.88	44.12
Init. weighted clock-tree size* (μm)	139231	265480	477488	103802	234309	377711	92468.6	179463	332338
Weighted clock-tree size ↓ (%)	31.85	30.39	30.74	23.03	42.50	41.27	29.18	43.07	37.53
Init. thermal-TSV demand*	307	486	539	569	887	1037	850	1617	1779
Thermal-TSV demand ↓ (%)	21.02	51.83	61.39	30.37	77.14	22.58	23.92	14.84	16.50
Est. signal WL (μm)	135588	298400	407762	146690	344503	459203	164400	442147	599411
Est. signal-WL red. (%)	8.06	16.71	7.24	11.40	19.24	n.a.***	11.29	n.a.***	n.a.***
Est. signal-routing util. incr. (%)	-8.35	-17.48	-8.41	-12.60	-20.77	n.a.***	-13.59	n.a.***	n.a.***
PG-TSV count*	92	92	110	108	115	131	128	109	166
Clock-TSV count*	6	8	8	6	12	16	9	18	12
Thermal-TSV count*	222	305	307	189	393	186	202	148	142
Signal-TSV count*	464	935	1123	838	1714	2022	1235	2455	2879
Die area (μm^2)	206528	206540	262656	124906	166036	187740	112700	104940	154298
Inital min. deadspace** (%)	55.65	56.97	46.95	51.61	64.60	50.61	59.32	57.36	54.69
Final min. deadspace** (%)	33.79	21.04	15.32	19.85	16.92	8.71	26.40	3.14	4.04

*Refers to the total value considering all dies.
** Refers to the minimal value of all dies. Extended boundary regions for PG-TSV rings are included; TSV KOZs are excluded.
*** Signal-TSV planning without considering packing was infeasible for the given outline.

small offsets for TSV groups. However, the estimated wirelength increases notably with die count, which favors integration using only two dies. This increase is mainly due to longer interconnects passing multiple dies (notably caused by nets connecting to external pins), which undermines wirelength reduction by shorter inter-die routes. Depending on die thickness, inserting multiple TSVs guided by tree construction may reduce wirelength [19]. However, this would increase TSV count notably and thus cost as well. **Second**, weighted clock-tree sizes decrease with increasing die count. Smaller sizes indicate lower power consumption, thus considering more dies is beneficial for clock-power optimization of separate dies. **Third**, our proposed IR-drop optimization is effective when using two or three dies but slightly limited in case of four dies. Also, the initially largest qualitative IR-drop decreases with die count in some cases. Both observations are possibly due to closer packing of blocks. Figure 7 illustrates the qualitative IR-drop distribution of the benchmark *n300* integrated on three dies. (Compare to Figure 5c for the corresponding SPICE simulation.) **Fourth**, the thermal-TSV demand increases with die count as expected, due to closer packing and stacking of blocks. Similar to IR-drop optimization, considering four dies is not appropriate for thermal optimization. In summary, these observations suggest that a limited die count helps to maintain design quality.

To validate our qualitative IR-drop distribution, we perform SPICE simulations of the PG grids and planned PG TSVs. The resistance of PG TSVs is calculated as $R_{TSV} \approx 16 m\Omega$, considering the electrical resistivity of copper $\rho_{Cu} = 0.02[\Omega \mu m]$ and TSV properties (Subsection IV-A2). Grid-wire resistances are calculated in a similar way. A voltage source supplying $1V$ is assumed to be connected to grid nodes of the lowermost die with assigned PG TSVs. Simulation results for reduction of initially largest IR-drop are given in Table III. We observe that our qualitative IR-drop distribution tends to underestimate simulated IR-drop reduction by on average 7.5% for integration using two or three dies, and to overestimate it by on average 3.3% for four-die integration. Thus, our proposed diagnostic is able to predict IR-drop with acceptable limitations of accuracy.

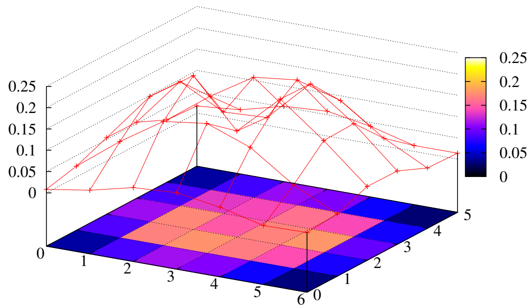

Fig. 7. Qualitative IR-drop distribution (power grid) of benchmark *n300*.

To validate our thermal-TSV planning algorithm, we perform finite element analysis (FEA) of the 3D-IC stacks using the open-source tools *SALOME* and *Elmer*.[2] Considered dimensions result from the 3D-IC configuration (Subsection IV-A2); for thermal conductivity $\lambda = \left[\frac{W}{m*K}\right]$, the following values are assumed. For the substrate (silicon) $\lambda_{Si} = 130$, for the TSVs (copper) $\lambda_{Cu} = 395$, and for the combination of bonding layer and metal layers $\lambda_{BEOL} = 66$. The heatsink atop has a heat transfer coefficient $h = 0.1 \left[\frac{W}{m^2*K}\right]$ and an ambient temperature $T = 300[K]$. Performing FEA after deadspace optimization, we observe that maximal-temperature reduction does not scale well with thermal-TSV increase, as expected. Temperature reductions are below 4% while comparing optimized layouts to baseline layouts. However, considering the initially optimized thermal distribution (Subsection IV-A1) and the increase of vertical thermal conductivity due to previously placed TSVs, this reduction appears reasonable. Furthermore, we note that the cost for additional thermal TSVs is limited; the ratio of thermal TSVs to all TSVs is below 17% on average. Figure 8 illustrates an FEA plot of the benchmark *n100*.

Figure 9 illustrates the floorplan of *n300* integrated on three dies.

V. CONCLUSION

Our work addresses the multiobjective optimization of deadspace, a critical resource for 3D-IC integration. Deadspace is limited and highly contested because it is required for several design tasks during early chip planning, such as TSV planning. To facilitate these tasks, we present a multiobjective optimization methodology called MoDo. It is motivated by the need for a unified approach to handle key challenges of block-level 3D-IC integration. We initially review these challenges and identify related deadspace-distribution requirements. We observe that these different requirements should be simultaneously satisfied to improve design quality. To do so, we develop a design-flow extension which incorporates algorithmic optimization for TSV planning, deadspace optimization, as well as design-quality

[2]SALOME generates finite-element meshes to facilitate heat-transfer modelling in Elmer. See additional details at http://www.salome-platform.org/ or http://www.csc.fi/english/pages/elmer, respectively.

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.

evaluation. Experimental results show that our methodology can simultaneously optimize interconnect, maximal temperature, estimated IR-drop and clock-tree size by improving deadspace distribution. We also observe that greater die count leads to greater TSV overhead and may undermine design quality. This suggests limiting the die count for block-level 3D-IC integration. Future work may consider transient IR-drop and related decap planning during deadspace optimization.

Acknowledgments

We are thankful for the reviewers' thoughtful comments. The work of J. Knechtel and M. Thiele was supported by the German Research Foundation under project 1401/1. The work of I. L. Markov was supported by the National Science Foundation.

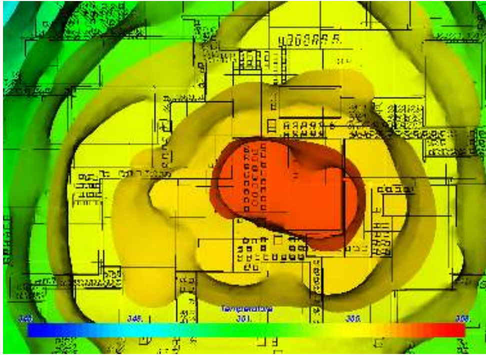

Fig. 8. The impact of TSV placement on heat conduction. Illustrated are thermal isosurfaces for the two-die integrated benchmark *n100*; the viewpoint is below the die stack. Small vertical blocks represent TSVs, design blocks are illustrated as horizontal blocks. Note that grouped TSVs next to the hotspot (red, centered region) limit the horizontal heat spreading due to desirable increased vertical conduction towards the heatsink atop (below in this view).

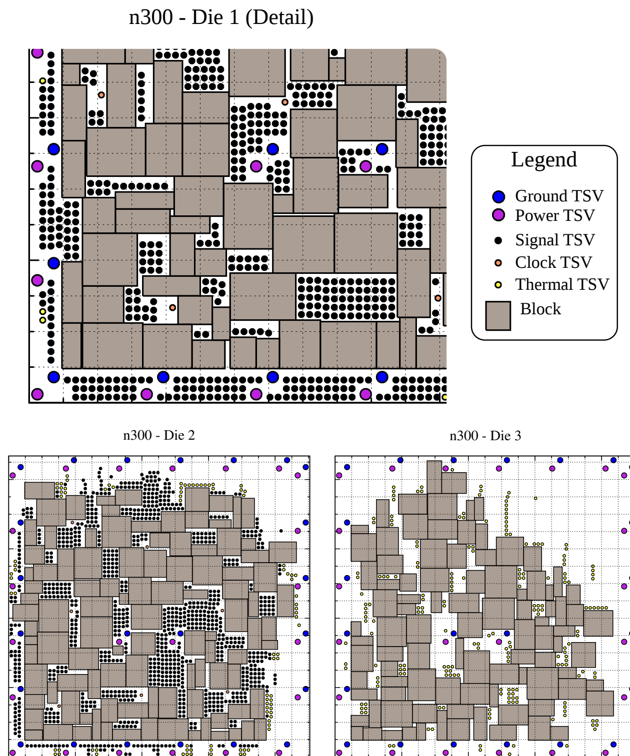

Fig. 9. Final floorplan of the benchmark *n300* after performing MoDo. The bottom die (Die 1) is shown in detail.

References

[1] S. N. Adya and I. L. Markov. Fixed-outline floorplanning: enabling hierarchical design. *IEEE TVLSI*, 11(6):1120–1135, 2003.

[2] S. N. Adya, I. L. Markov, and P. G. Villarrubia. On whitespace and stability in physical synthesis. *Integration*, 39(4):340–362, 2006.

[3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. *Proc. SODA*, pp. 1027–1035, 2007.

[4] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proc. IEEE*, 89(5):602–633, 2001.

[5] S. Borkar. 3D integration for energy efficient system design. *Proc. DAC*, pp. 214–219, 2011.

[6] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Hierarchical whitespace allocation in top-down placement. *IEEE TCAD*, 22(11):716–724, 2003.

[7] H.-T. Chen, H.-L. Ling, Z.-C. Wang, and T. T. Hwang. A new architecture for power network in 3D IC. *Proc. DATE*, pp. 1–6, 2011.

[8] J. Cong, G. Luo, and Y. Shi. Thermal-aware cell and through-silicon-via co-placement for 3D ICs. *Proc. DAC*, pp. 670–675, 2011.

[9] J. Cong, J. Wei, and Y. Zhang. A thermal-driven floorplanning algorithm for 3D ICs. *Proc. ICCAD*, pp. 306–313, 2004.

[10] X. Dong et al. Fabrication cost analysis and cost-aware design space exploration for 3-D ICs. *IEEE TCAD*, 29(12):1959–1972, 2010.

[11] P. Falkenstern, Y. Xie, Y.-W. Chang, and Y. Wang. Three-dimensional integrated circuits (3D IC) floorplan and power/ground network co-synthesis. *Proc. ASPDAC*, pp. 169–174, 2010.

[12] X. He, S. Dong, Y. Ma, and X. Hong. Simultaneous buffer and interlayer via planning for 3D floorplanning. *Proc. ISQED*, pp. 740–745, 2009.

[13] M. B. Healy and S. K. Lim. Power delivery system architecture for many-tier 3D systems. *Proc. ECTC*, pp. 1682–1688, 2010.

[14] M. B. Healy and S. K. Lim. Power-supply-network design in 3D integrated systems. *Proc. ISQED*, pp. 223–228, 2011.

[15] A.-C. Hsieh et al. TSV redundancy: Architecture and design issues in 3D IC. *Proc. DATE*, pp. 166–171, 2010.

[16] M. Jung and S. K. Lim. A study of IR-drop noise issues in 3D ICs with through-silicon-vias. *Proc. 3DIC*, pp. 1–7, 2010.

[17] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 2011.

[18] D. H. Kim, S. Mukhopadhyay, and S. K. Lim. Through-silicon-via aware interconnect prediction and optimization for 3D stacked ICs. *Proc. SLIP*, pp. 85–92, 2009.

[19] D. H. Kim, R. O. Topaloglu, and S. K. Lim. Block-level 3D IC design with through-silicon-via planning. *Proc. ASPDAC*, pp. 335–340, 2012.

[20] J. Knechtel, I. L. Markov, and J. Lienig. Assembling 2-D blocks into 3-D chips. *IEEE TCAD*, 31(2):228–241, 2012.

[21] D.-J. Lee, M.-C. Kim, and I. L. Markov. Low-power clock trees for CPUs. *Proc. ICCAD*, pp. 444–451, 2010.

[22] D.-J. Lee and I. L. Markov. Obstacle-aware clock-tree shaping during placement. *IEEE TCAD*, 31(2):205–216, 2012.

[23] Y.-J. Lee and S. K. Lim. Co-optimization and analysis of signal, power, and thermal interconnects in 3-D ICs. *TCAD*, 30(11):1635–1648, 2011.

[24] D. L. Lewis and H.-H. S. Lee. Test strategies for 3D die stacked integrated circuits. Proc. DATE 3D Workshop, 2009.

[25] F. Li et al. Design and management of 3D chip multiprocessors using network-in-memory. *Proc. ISCA*, pp. 130–141, 2006.

[26] X. Li et al. LP based white space redistribution for thermal via planning and performance optimization in 3D ICs. *ASPDAC*, pp. 209–212, 2008.

[27] Z. Li et al. Integrating dynamic thermal via planning with 3D floorplanning algorithm. *Proc. ISPD*, pp. 178–185, 2006.

[28] K. H. Lu et al. Thermo-mechanical reliability of 3-D ICs containing through silicon vias. *Proc. ECTC*, pp. 630–634, 2009.

[29] V. S. Nandakumar and M. Marek-Sadowska. Layout effects in fine-grain 3-D integrated regular microprocessor blocks. *DAC*, pp. 639–644, 2011.

[30] S. S. Sapatnekar. Addressing thermal and power delivery bottlenecks in 3D circuits. *Proc. ASPDAC*, pp. 423–428, 2009.

[31] D. Sylvester and K. Keutzer. A global wiring paradigm for deep submicron design. *IEEE TCAD*, 19(2):242–252, 2000.

[32] E. F. Y. Young, C. C. N. Chu, and M. L. Ho. Placement constraints in floorplan design. *IEEE TVLSI*, 12(7):735–745, 2004.

[33] X. Zhao and S. K. Lim. Power and slew-aware clock network design for through-silicon-via (TSV) based 3D ICs. *ASPDAC*, pp. 175–180, 2010.

[34] X. Zhao, J. Minz, and S. K. Lim. Low-power and reliable clock network design for through-silicon via (TSV) based 3D ICs. *IEEE TCPMT*, 1(2):247–259, 2011.

[35] P. Zhou et al. 3D-STAF: scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits. *Proc. ICCAD*, pp. 590–597, 2007.

© IEEE 2012. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Proceedings IEEE/ACM Int. Conference on Computer-Aided Design (ICCAD), pp. 705-712, San Jose, CA, November 2012.