3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three-Dimensional Integrated Circuits

Pingqiang Zhou*, Yuchun Ma*, Zhouyuan Li†,
Robert P. Dick‡, Li Shang§, Hai Zhou‡, Xianlong Hong*, Qiang Zhou*

*CS Department
Tsinghua University
Beijing, 100084, China
zpq05@mails.tsinghua.edu.cn
{myc, hxl-dcs, zhouqiang}@
mail.tsinghua.edu.cn

†Advanced Technology Group
Synopsys, Inc.
Beijing, China
zhuoyuan@synopsys.com

‡EECS Department
Northwestern University
Evanston, IL 60208, U.S.A.
dickrp@northwestern.edu
haizhou@eecs.northwestern.edu

§ECE Department
Queen's University
Kingston, ON K7L 3N6,
Canada
li.shang@queensu.ca

***Abstract*— Thermal issues are a primary concern in the three-dimensional (3D) integrated circuit (IC) design. Temperature, area, and wire length must be simultaneously optimized during 3D floorplanning, significantly increasing optimization complexity. Most existing floorplanners use combinatorial stochastic optimization techniques, hampering performance and scalability when used for 3D floorplanning. In this work, we propose and evaluate a scalable, temperature-aware, force-directed floorplanner called *3D-STAF*. Force-directed techniques, although efficient at reacting to physical information such as temperature gradients, must eventually eliminate overlap. This can cause significant displacement when used for heterogeneous blocks. To smooth the transition from an unconstrained 3D placement to a legalized, layer-assigned floorplan, we propose a three-stage force-directed optimization flow combined with new legalization techniques that eliminate white spaces and block overlapping during multi-layer floorplanning. A temperature-dependent leakage model is used within 3D-STAF to permit optimization based on the feedback loop connecting thermal profile and leakage power consumption. 3D-STAF has good performance that scales well for large problem instances. Compared to recently published 3D floorplanning work, 3D-STAF improves the area by 6%, wire length by 16%, via count by 22%, peak temperature by 6% while running nearly 4× faster on average.**

I. Introduction and Related Work

Three-dimensional (3D) integration, in which multiple integrated circuit (IC) device layers are vertically stacked and attached (see Figure 1), can be used to decrease wire delay, increase integration density, improve performance, and reduce power consumption [1]. Although 3D integration has many potential benefits, it generally increases the difficulty of IC cooling. Worse yet, ongoing reduction in device threshold voltage, channel length, and gate oxide thickness are increasing leakage power. The International Technology Roadmap for Semiconductors [2] predicts that leakage power will account for 50% of the total power in next-generation processors. The super-linear relationship [3] between temperature and leakage power further complicates thermal optimization of deep submicron 3D ICs.

Thermal issues must be considered during every stage of 3D IC design. Floorplanning is a well-studied problem for two-dimensional (2D) IC design. Moving to 3D ICs increases the problem complexity.

1) The design space of 3D IC floorplanning increases exponentially with the number of active layers. Li et al. showed that, given a flooplanning problem with n blocks, the solution space of 3D floorplanning with L layers increases by $n^{L-1}/(L-1)!$ times compared to the 2D case [4].

2) The addition of a temperature constraint or temperature minimization objective complicates optimization, requiring trade-offs among area, wire length, and thermal characteristics.

3) Inter-layer and intra-layer block dependencies affect IC temperature, area efficiency, and wire length. Layer assignment and intra-layer block placement must be jointly solved during 3D floorplanning.

4) In deep-submicron 3D ICs, it is necessary to account for the closed temperature/leakage power feedback loop to accurately estimate or optimize either.

Since the 2D and 3D rectangular packing problems are $\mathcal{NP}$-hard, most floorplanning algorithms are based on stochastic combinatorial optimization techniques such as simulated annealing. Floorplanning algorithms use various kinds of floorplan representations such as Corner Block List [5], Sequence Pair (SP) [6], Bounded Sliceline Grid (BSG) [7], B*-tree [8] and Transitive Closure Graph [9]. To enable the 3D floorplanning, researchers have extended 2D algorithms to represent the floorplans of different layers with an array of 2D representations, such as two layer BSG [10] and four-layer SP [11]. With these representations, blocks are moved inside each layer or swapped between different layers during optimization.

Cong et al. proposed a thermal-driven floorplanning algorithm for 3D ICs [12]. It uses simulated annealing with an integrated compact thermal model. Hung et al. proposed temperature-aware floorplanning for 3D microprocessors [13]. The power consumption of interconnect is considered during floorplanning. Recently, Li et al. developed a hierarchical 3D floorplanning algorithm [4]. All of these solutions use simulated annealing based stochastic optimization techniques, which generally have long run times that scale poorly with problem size. Section IV provides some evidence that the runtime of simulated annealing technique increases super-linearly with problem size.

Recently, Obermeier and Johannes [14] as well as Goplen and Sapatnekar [15] developed force-directed temperature-aware standard-cell placement algorithms. These techniques demonstrate good, scalable performance. Though straightforward layer assignment works for standard cells, heterogeneity in block sizes and shapes complicates the problem. A small change during optimization can cause large displacements in the final legalized packing. Therefore, straightforward layer assignment and intra-layer legalization may result in large white space and area imbalance among layers. This imbalance cannot easily be fixed by post-placement exchanges. Force-directed floorplanning of heterogeneous blocks requires that the transition between global placement and the final packing be smoothed.

This work was supported in part by the NSFC under award 60606007, in part by the Tsinghua Basic Research Fund under award JC20070021, in part by SRC under award 2007-HJ-1593, in part by the NSF under awards CCF-0702761 and CNS-0347941, and in part by NSERC under Discovery Grant #388694-01.

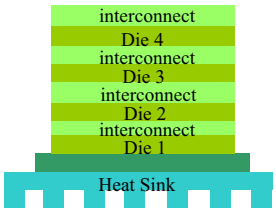
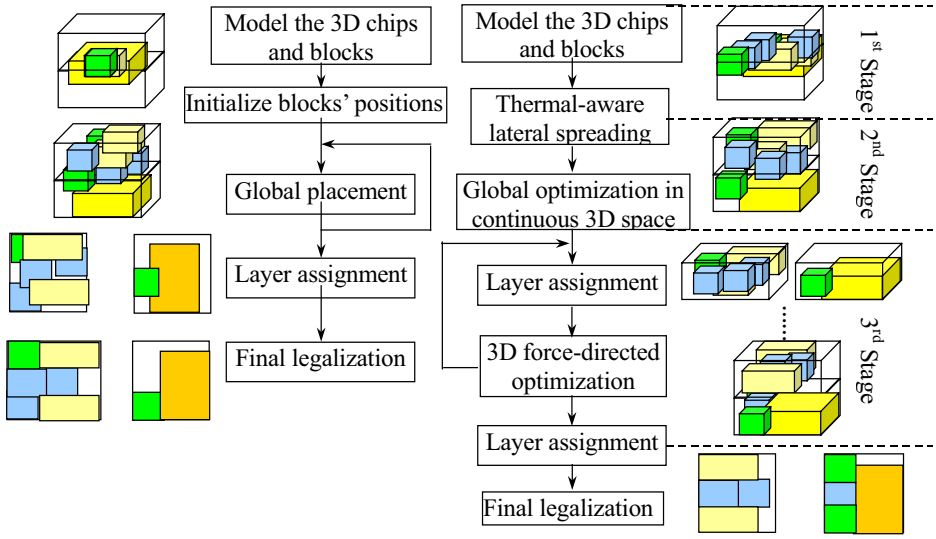


Fig. 1. 3D IC technology.



(a) Traditional 3D force-directed placement flow (b) Adaptive three-stage 3D force-directed floorplanning

Fig. 2. Traditional force-directed flow and adaptive three-stage 3D flow.

II. 3D Temperature-Aware Floorplanning

This section describes the proposed temperature-aware 3D IC floorplanning algorithm. Section II-A gives an overview of the proposed solution. Section II-B defines the problem formally. Section II-C describes the proposed solution in detail.

II.A. Overview of Proposed Technology

Most previous floorplanning approaches using simulated annealing require ten of thousands of iterations to produce solutions of good quality and the number of iterations generally increases super-linearly with problem size. We propose a 3D temperature-aware floorplanning algorithm that optimizes peak temperature, area, wire length, and via count. It runs quickly and scales well with increasing problem size. This algorithm is rooted in force-directed placement techniques and exploits domain-specific information, i.e., thermal gradients. Our flow has two phases: global placement and legalization.

Starting from a placement solution requires translation from continuous space to a discrete, layer-assigned, legalized solution. For blocks with heterogeneous sizes and shapes, small local changes may cause significant changes to the global solution. We use two techniques to solve this problem. First, we propose a three-stage global placement flow to control the distribution adaptively. Overlaps are permitted, but controlled, during the early stages. This eases the legalization stage. Second, we exploit flexibility in block orientations to assist legalization. When we construct the topological relations among blocks, the orientations of blocks are taken into consideration to optimize the displacement. The proposed algorithm has the following main features:

1) We propose and develop 3D-STAF, a temperature-aware floorplanner with run time that scales well with problem size. This is the first work using force-directed techniques for 3D IC floorplanning problems in which block sizes and shapes are highly heterogeneous.

2) This algorithm simultaneously optimized temperature, area, and wire length under a constraint on total via count.

3) The algorithm supports gradual transition from continuous (placement) to discrete (floorplanning) space. The force-directed approach is combined with a macro-block legalization method that jointly optimizes block motions and rotations during legalization and effectively eliminates block overlapping.

4) The proposed floorplanner integrates an iterative power–thermal analysis model that closes the leakage-temperature feedback loop. Though several papers [16], [17] considered the effects caused by leakage power in 2D designs, this paper is the first work to consider temperature-dependent leakage power during 3D physical design.

II.B. Problem Definition

An instance of the 3D temperature-aware floorplanning problem is composed of a set of blocks $\{m_1, m_2, \cdots, m_n\}$. A block m_i is a $W_i \times H_i$ rectangle with area A_i, aspect ratio H_i/W_i, and power density PD_i. Each block is free to rotate. There is fixed number of layers L. Let the tuple (x_i, y_i, l_i) denote the coordinates of the lower-left corner of block m_i, where $1 \leq l_i \leq L$. A 3D floorplan F is an assignment of (x_i, y_i, l_i) for each block m_i such that no two blocks overlap. The objectives of our 3D temperature-aware floorplanning algorithms are to minimize (1) chip peak temperature $T_{\max}$, (2) wire length, and (3) chip area. Chip area is the product of the maximal height and maximal width over all layers. Wire length is the half-perimeter wire length estimation.

II.C. Adaptive Three-Stage Force-Directed Approach

Applying force-directed optimization to problems with conflicting objectives or sub-optimal local minima is difficult. This is especially true for 3D IC floorplanning, in which thermal profiles depend on layer assignment, the distribution of high power density blocks, and the locations of whitespace. The transition from continuous to discrete space may disrupt nearly-optimal continuous-domain results. Therefore, we propose a three-stage global optimization flow that smooths the transition from continuous to discrete space. The three stages are (1) Temperature-Aware Lateral Spreading; (2) Global Optimization in Continuous 3D Space, and (2) Optimization in 2.5D Space with Layer Assignment.

As shown in Figure 2(b), we first spread blocks laterally in the x–y plane to produce an initial layout. Then we optimize the positions of blocks in continuous 3D space. When the sum of overlaps among blocks is reduced to a small fraction of the total area, we start to integrate discrete layer assignment pressures within the force-directed algorithm. After a few iterations, the packing has little remaining overlap and the blocks are evenly distributed among, and within, layers. This contrasts with the traditional force-directed placement flow shown in Figure 2(a). The traditional flow is simpler but is

susceptible to large deviation from optimality during the legalization stage in which overlap is eliminated, especially when used for blocks with heterogeneous sizes and shapes.

II.C.1) Force-Directed Techniques: In this paper, we extend the force-directed approach to handle the 3D temperature-aware macro-block floorplanning. The proposed force-directed algorithm simulates the mechanics problem in which particles are attached to springs and their movement obeys Hooke's law. A system of quadratic equations is built based on connections between blocks [14], [15].

$$c_{ij}\left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right] \tag{1}$$

where c_{ij} is the weight of the connection between the two nodes. If the c_{ij} coefficients are combined into a global net stiffness matrix, $\mathbf{C}$, an objective function can be written for the entire system:

$$\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{C}x + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}y + \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{C}z \tag{2}$$

where x, y, and z are the coordinates of all blocks and points of interest. This objective function can be minimized by solving the following three systems of equations:

$$\mathbf{C}x = f_x, \mathbf{C}y = f_y, \text{ and } \mathbf{C}z = f_z \tag{3}$$

We model the multi-layer design in 3D space by assuming each layer has the same thickness (D) in z-dimension and blocks are modeled as 3D rectangles with uniform z-axis depths. A homogeneous cubic bin structure is overlayed on the 3D space to simplify computation of forces. To trade-off accuracy and computation time, we set the thickness in z-direction of each bin to $D/2$ and the size in x–y space to half the minimum block size. Based on this bin structure, two kinds of forces, Filling Forces and Thermal Forces, are introduced to eliminate overlaps and reduce placement peak temperature.

- **Filling Force (f_x^F, f_y^F, f_z^F):** Filling Force is used to eliminate overlap between blocks and distribute them evenly over the 3D placement region. It drives the placement to remove overlap by pushing blocks away from regions of high density and pulling blocks toward regions of low density in 3D space. We define the bin density as the sum of the block areas covering the bin. Each bin's Filling Force is equal to its bin density. A block receives a Filling Force equal to the sum of the prorated Filling Forces of the bins the cell covers.
- **Thermal Force (f_x^T, f_y^T, f_z^T):** We used the thermal model described in Section III to obtain thermal gradient for a placement. We would like to move blocks (which produce heat) away from regions of high temperature. This goal is achieved by using the thermal gradient to determine directions and magnitudes of the Thermal Forces on blocks.

A weighted sum of these two forces is used to compute an aggregate forces in each direction, yielding the following systems of equations:

$$\begin{aligned}
\mathbf{C}x &= \alpha_x\left[\beta_x f_x^F + (1-\beta_x)\, f_x^T\right], \\
\mathbf{C}y &= \alpha_y\left[\beta_y f_y^F + (1-\beta_y)\, f_y^T\right], \\
\mathbf{C}z &= \alpha_z\left[\beta_z f_z^F + (1-\beta_z)\, f_z^T\right].
\end{aligned} \tag{4}$$

where the α weight parameters control the magnitudes of forces in the above equations, thereby influencing the amount of block displacement per iteration resulting. The β parameters are used to adjust the percentages of contributions between Filling Force and Thermal Force over iterations. In combination, the α and β parameters permit control over the relative importance of wire length minimization, Thermal, and Filling forces. The values of these parameters are experimentally determined, but are general; they do not need to be adjusted to suit individual problem instances.

We used a 3D oct-tree extension of the Barnes-Hut quad-tree for n-body force calculation [18], [19]. Given a 3D packing, blocks and their temperatures are inserted into the oct-tree based on their positions and shapes. The Filling Force and Thermal Force for a given block are calculated by summing the individual forces upon the bins the block occupies in each level of the tree. Forces from a bin and from its nearest neighbors are considered. Large blocks span numerous oct-tree bins. As a consequence, they receive greater forces than small blocks.

II.C.2) Temperature-Aware Lateral Spreading: Instead of spreading blocks from the center of the chip to every corner of the cubic space, we first spread blocks laterally in the x–y plane. This provides a good initial distribution of blocks before vertical spreading begins. In the traditional force-directed approach, all blocks are initially centrally located and are subsequently spread from this point, as shown in Figure 2. Initially, large overlaps result in large Filling Forces. Therefore, the initial optimization is influenced primarily by overlap instead of thermal effects and interconnect length. If such a technique were used, it would be likely for some cool blocks with large areas to be pushed near the heatsink due to overestimated Filling Forces, while the hot blocks are pushed far from the heatsink. This can result in hot-spots. However, if we first restrict the spreading to the x–y plane, the blocks are spread laterally and the temperature-aware spreading will separate hot blocks and gather heavily-communicating blocks on x–y space. The benefits of the initial lateral spreading follow: (1) it can evenly distribute lateral power density; (2) the lateral spreading can give the initial distribution of interconnects since the wire length estimation is 2D x-y space computation; and (3) overlaps are controlled to support subsequent 3D optimization.

II.C.3) Global optimization in continuous 3D space: To permit thermal gradient guided optimization, we first globally optimize the block locations in continuous 3D space. Based on the 3D bin structure, the Filling Forces can be computed to avoid the overlaps between blocks. However, Thermal Forces are computed based on multi-layer IC thermal analysis (see Section III) since the thermal model is based on discrete layer assignment. Therefore, it is necessary to tentatively map the power distribution from continuous 3D space to discrete 2D layers before each analysis run. This is achieved by stochastically mapping blocks to layers. Suppose that the center of block $m_i(x_i, y_i, z_i)$ is between layer q and layer $q-1$, which means $Z_{q-1} < z_i + D/2 < Z_q$, where D is the depth of block along the z-axis. $P(m_i, q)$ is used to denote the probability that block m_i will be assigned to layer q. The probability for a block to be assigned to a layer is related to the vertical distance between the block and the layer. The nearer the block is to the layer, the higher probability that the block will be assigned to this layer. A block can only be assigned to one of its two neighboring layers.

$$P(m_i, q) = (z_i + \frac{1}{2}D - Z_{q-1})/D \tag{5}$$

$$P(m_i, q-1) = (Z_q - (z_i + \frac{1}{2}D))/D \tag{6}$$

If the power density of block m_i is PD_i, the projection of m_i onto layer q is a rectangle defined as follows: $(x_i, y_i, Z_q, x_i + w_i, y_i + h_i, Z_q)$. We define $MPD(m_i, q)$ to be the stochastically-mapped power density for layer q, i.e.,

$$MPD(m_i, q) = PD_i * P(m_i, q) \tag{7}$$

$$MPD(m_i, q-1) = PD_i * P(m_i, q-1) \tag{8}$$

By computing the mapped power densities of all blocks, we can obtain the stochastic power density distribution on each layer. The thermal model in Section III is used to compute the temperature in each bin. The resulting temperature gradients are used to determine the Thermal Forces applied to blocks.

During this stage, we extend the spreading metric based on violation measure in the Barns-Hut quad-tree [18]. This metric is used to determine when 3D block spreading in continuous space is adequate. Our experiment results indicate that the global optimization process proceed to final layer assignment and legalization stage when approximately 5%–10% overlap remains.

II.C.4) Optimization in 2.5D Space With Layer Assignment: After optimizing the placement in continuous 3D space, blocks must be assigned to discrete IC layers. However, straightforward discrete layer assignment may be suboptimal. Therefore, instead of treating layer assignment as a separate post-processing stage, we integrate placement and layer assignment to permit optimization in *2.5D space*, i.e., we introduce an intermediate representation sitting between 3D and 2D (see Figure 2).

In the above approach, each block is modeled as a 3D rectangle that can be moved freely in continuous 3D space. Layer assignment moves blocks from continuous space to discrete space, forcing each block to occupy exactly one IC layer. The force-directed approach tries to gradually distribute the blocks evenly in space. Initially, the blocks are still far from their final positions. Therefore, direct layer assignment would disrupt the convergence of the optimization algorithm. As time proceeds, the blocks begin to approach their final positions and are assigned to discrete layers.

Layer assignment is based on block positions on the z-axis, derived from the current placement obtained by the force-directed approach. Assume there are n blocks $\{m_1, m_2, \cdots, m_n\}$ that should be placed on L layers. To evenly distribute blocks on each layer, we set the area thresholds for layers to $\{AT^1, AT^2, \cdots, AT^L\}$. The current total occupancies of each layer are represented as $\{Ao^1, Ao^2, \cdots, Ao^L\}$. Equal area thresholds are used for each layer.

$$AT_i = a \sum_{i=1}^{n} A_i / L \tag{9}$$

where a is a weighting coefficient ($a \approx 1$). If a block m_i is near or crosses layer p in the initial packing, we assume this block would be assigned to layer $p-1$, p, or $p+1$. Blocks may not be moved across layers. Suppose the projection of block m_i on a certain layer p is R_i^p. We define a term ζ_i^p to be *potential overlap ratio* of R_i^p. If the current occupancy of layer p exceeds the threshold, we punish the assignment of block m_i to layer p by defining ζ_i^p to be 1. Otherwise, we set the potential overlap ratio to the total overlap ratio of R_i^p with other blocks on layer p. We define the total overlapped area in region R_i^p to be $Total\_Over_i^p$.

$$\zeta_i^p = \begin{cases} 1 & \text{if } Ao^p + A_i > AT^p, \\ Total\_Over_i^p / A_i & \text{if } Ao^p + A_i \leq AT^p. \end{cases} \tag{10}$$

We set the overlap threshold for layer p to OT^p. To balance the thermal distribution, layer assignment starts from the layer closest to the heatsink, attempting to assign blocks as low as possible considering overlaps, and proceed upward. To control the number of vias, we dynamically compute the number of vias implied by a block layer assignment decision. Layer assignment attempts to honor the via constraint. However, due to sequential layer assignment, it does not guarantee that the constraint will be met.

Figure 3 illustrates the process of layer assignment for three blocks. Since layer assignment disturbs the positions of blocks, there are

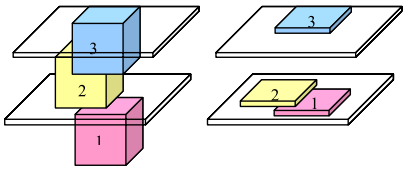


Fig. 3. Layer assignment.

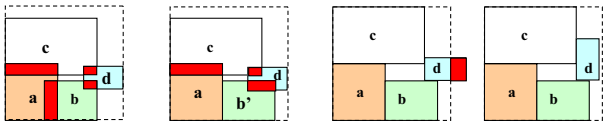


Fig. 4. Legalization process.

mismatches between optimization in the continuous and discrete spaces. As shown in Figure 3, in continuous space, block 2 overlaps block 3. After layer assignment, blocks 2 and 3 are assigned to different layers, removing the overlap. Blocks 1 and 2 are assigned to the same layer resulting in overlap. One must be careful to prevent the transition from continuous space to discrete multi-layer space from generating severe distortions to the tentative floorplan. Attempting to correct such distortions via post processing based on the random exchange of blocks among layers does not efficiently produce a high-quality corrected layer assignment. In the proposed algorithm, this problem is circumvented by smoothing the transitions between continuous space and discrete multi-layer design space by integrating layer assignment with the force-directed approach.

In this stage, we perform the layer assignment according to the current placement. Since all blocks are assigned to layers, the thermal model can be used to compute temperature gradients and the resulting Thermal Force vectors. Filling Forces are also computed. The changes in positions caused by layer assignment lead the subsequent force-directed iterations to adjust the placement. The optimization process ultimately converges to the final multi-layer floorplan.

II.C.5) Macro-Block Legalization: After the global placement described in the previous sections, we arrive at a multi-layer packing solution with little residual overlap. To obtain a feasible placement, our legalization strategy perturbs the solution slightly to produce an overlap-free packing while attempting to maintain the original topological relationships among blocks.

Few of the deterministic legalization methods consider the orientation of blocks. However, since the sizes and shapes of the blocks vary, considering only block motion without rotation may result in large displacements, disrupting a solution that originally had good area, interconnect, and thermal properties. The rotation of blocks can permit legalization with less displacement. We propose a macro-block legalization method that optimizes block orientations.

Our legalization problem definition is similar to that in past work [20], [21]: construct the topological relations between overlapping blocks so that the displacements of blocks are minimized. However, unlike previous work, we optimize block orientations as well as topological relations.

We first sort blocks according to their positions from the lower-left to upper-right corners of the IC to get a rough topological sequence. As shown in Figure 4, block a is before block b in the sequence and they overlap with each other. Each block has 4 orientations. We must determine whether block b should be right of or above block a and choose the best orientation. To evaluate the displacement caused by these possible topological relations, we define the term EVd. Suppose block b is moved to the right to eliminate overlap with block a. This movement generates new overlaps with block d. The evaluation of the

displacement includes the movement of block b and the new overlap generated by this movement. Suppose block b is moved to b'. Given that $TOPO$ is the topological relation between b' and a (either to the right of or above),

$$EVd(a, b, O_b, TOPO) = Dist(b, b') + \psi(b', O_b) \quad (11)$$

Where $Dist(b, b')$ is the distance between the center of b and the center of b', $\psi(b', O_b)$ is the total area of the overlap with other blocks if we move block b to b' orientation is O_b.

We incrementally construct the constraint graph by choosing the best topological relationships while minimizing the resulting displacement. This was inspired by work by Moffitt et al. [20]. In some cases, this generates whitespace along the boundary of the moved block. As shown in Figure 4, directly moving block d to the right without rotation results in a significant increase in whitespace along the right boundary. Therefore, we set an approximate bounding box for the packing and the partial areas of the blocks falling outside the box are also treated as the overlaps. The area of the packing is controlled by the evaluation of the displacement. The resulting constraint graph is used to produce a legalized packing.

III. Thermal and Leakage Models

In this section, we introduce the 3D thermal analysis technique and temperature-dependent leakage power model used in 3D-STAF.

III.A. Introduction to 3D IC Thermal Analysis Problem

Thermal analysis is the simulation of heat transfer through heterogeneous material among heat producers (e.g., transistors) and heat consumers (e.g., heat-sinks attached to an IC). Modeling thermal conduction is analogous to modeling electrical conduction, with thermal conductivity corresponding to electrical conductivity, power dissipation corresponding to electrical current, and temperature corresponding to voltage [22]. One can spatially discretize the system being analyzed and solve the following equation to determine the steady-state thermal profile as a function of power profile.

$$T = P\mathbf{A}^{-1} \quad (12)$$

where $\mathbf{A}$ is an $[N \times N]$ sparse thermal conductivity matrix. T and $P(t)$ are $[N \times 1]$ temperature and power vectors. This model treats each thermal element as isothermal, i.e., all points within the element have the same temperature. Therefore, a large N may be required for accurate thermal analysis, making direct solvers for T unacceptable slow for use during each iteration of thermal aware floorplanning.

In a 3D ICs, heat may be dissipated in multiple IC power layers. In addition the material structure of the stacked IC contains alternating layers of silicon (with a thermal conductivity, k, of approximately 150 W/mK) and polyimide epoxy (0.05 W/mK), as well as a copper heat-sink (285 W/mK). This structure fits within the general model specified by Equation 12. However, the heterogeneous 3D structure complicates the thermal model, further increasing the importance of using an efficient analysis technique.

III.B. Thermal Analysis

Zhan and Sapatnekar [23] proposed a steady-state thermal analysis method based on the Green's function formalism that was accelerated by using discrete cosine transforms and a look-up table [23]. Li et al. proposed a full-chip steady-state thermal analysis method [24]. In this work, matrix operations are handled using the multi-grid method. However, although the advantages of heterogeneous element discretization is noted, in their work, no systematic adaptation method is provided. 3D-STAF uses an extended version of a spatially-adaptive 3D multi-layer chip-package thermal analysis software package [25] in the inner loop of floorplanning. This software provides feedback guiding placement moves during optimization.

III.C. Leakage Power Model

Leakage current can be modeled as follows [3]:

$$I_{sub} = A_s \frac{W}{L} {v_T}^2 e^{\frac{(V_{GS}-V_{TH})}{n v_T}} \left(1 - e^{\frac{-V_{DS}}{v_T}}\right) \quad (13)$$

where

- A_s is a technology-dependent constant,
- V_{TH} is the threshold voltage,
- L and W are the device effective channel length and width,
- V_{GS} is the gate-to-source voltage,
- n is the subthreshold swing coefficient for the transistor,
- V_{DS} is the drain-to-source voltage, and
- v_T is the thermal voltage.

For V_{DS} significantly greater than v_T, we can simplify to

$$I_{sub} = A_s \frac{Wk^2}{Lq^2} T^2 e^{\frac{q(V_{GS}-V_{TH})}{nkT}} \quad (14)$$

where k is Boltzmann's constant, T is temperature, q is the elementary charge, and n is the subthreshold swing coefficient. Let

$$K1 = A_s \frac{Wk^2}{Lq^2} \text{ and} \quad (15)$$

$$K2 = \frac{q(V_{GS} - V_{TH})}{nk} \quad (16)$$

Then,

$$I_{sub} = K1 T^2 e^{\frac{K2}{T}} \quad (17)$$

The thermal profile can be obtained by iteratively conducting thermal analysis and leakage power estimation until convergence. This usually requires only a few iterations.

IV. Experimental Results

We implemented 3D-STAF, the proposed temperature-aware 3D floorplanner, in C++ on Linux. In this section, we present the results produced by 3D-STAF on a number of floorplanning/placement problems. Section IV-A presents the impact of each phase within the 3D-STAF optimization flow. Section IV-B compares a version of 3D-STAF that optimizes only wire length and area with a version of CBA [12] with the same optimization objectives. Section IV-C compares versions of 3D-STAF and CBA with thermal effect considered. Section IV-D shows the impact of considering the interdependence between temperature and leakage power consumption during the temperature-aware floorplanning flow.

All experiments were performed on a workstation with 3.0 GHz CPU and 4 GB physical memory. We tested our algorithm on MCNC benchmarks and GSRC benchmarks [12]. The number in each benchmark's name indicates the number of blocks, i.e., these numbers correspond to problem instance size. We assume the pads are located at the center of the chip; this permitted comparison with past work. Four device layers are used for all circuits.

IV.A. Optimization Process in 3D-STAF

This goal of this section is to illustrate the relative importance and impact of each stage of the optimization flow used in 3D-STAF. Later sections will focus on comparing solution quality with past work. To illustrate the operation of proposed three-stage optimization flow, we report the percentage of the overlapping block area (*Ovlp*), bounding box area, wire length, and runtime of each floorplanning stage (see Table I). We fixed the packing region in the global placement process. Blocks are spread on the x–y plane in the first stage and then spread

TABLE I
EFFECTS OF OPTIMIZATION STAGES

Circuit	Layer No	Stage1 + Stage2				Stage3				Legalization			
		Ovlp (%)	Area (mm^2)	HPWL (mm)	Time (s)	Ovlp (%)	Area (mm^2)	HPWL (mm)	Time (s)	Ovlp (%)	Area (mm^2)	HPWL (mm)	Time (s)
ami33	1	48.9	30.3	9.8	34	2.9	30.3	20.5 (+109%)	26	0.0	39.1 (+29%)	21.1 (+115%)	0.10
	2	16.5				0.8				0.0			
	3	16.9				0.2				0.0			
	4	25.3				0.6				0.0			
ami49	1	29.7	974.7	258.1	34	3.4	974.7	467.0 (+81%)	31	0.0	1392.8 (+43%)	411.2 (+59%)	1.22
	2	12.4				1.6				0.0			
	3	11.0				0.2				0.0			
	4	13.4				1.6				0.0			
n100	1	30.7	4.9	21.9	36	0.7	4.9	90.1 (+311%)	35	0.0	5.3 (+10%)	90.9 (+315%)	3.21
	2	20.8				0.1				0.0			
	3	19.8				0.1				0.0			
	4	13.4				1.6				0.0			
n200	1	8.5	4.6	76.8	113	0.61	4.6	155.2 (+102%)	287	0.0	6.1 (+33%)	164.5 (+114%)	6.7
	2	8.4				0.07				0.0			
	3	7.3				0.07				0.0			
	4	7.6				0.55				0.0			
n300	1	6.2	7.5	109.1	126	0.55	7.5	216.7 (+99%)	308	0.0	8.8 (+17%)	224.6 (+106%)	7.8
	2	3.6				0.04				0.0			
	3	7.2				0.04				0.0			
	4	6.6				0.45				0.0			
Aggregate		17.8			69	0.77	+0.0%	+140%	137	0.0	+26%	+142%	3.8

TABLE II
COMPARISON FOR AREA AND WIRE LENGTH OPTIMIZATION

Circuit	CBA				3D-STAF: NT			
	Area (mm^2)	HPWL (mm)	Via (count)	Time (s)	Area (mm^2)	HPWL (mm)	Via (count)	Time (s)
ami33	35.3	22.5	93	23	37.9	22.0	122	52
ami49	1490.0	446.8	179	86	1349.1	437.5	227	57
n100	5.29	100.5	955	313	5.9	91.3	828	68
n200	5.77	210.3	2093	1994	5.9	168.6	1729	397
n300	8.90	315.0	2326	3480	9.7	237.9	1554	392
Aggregate relative to CBA					+4%	-12%	-1%	-31%

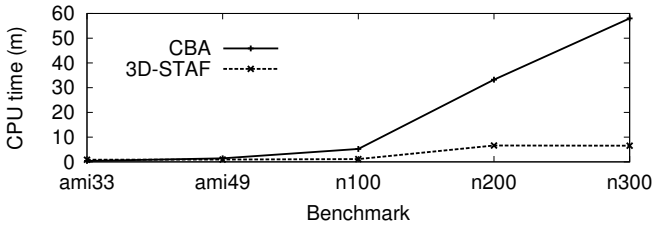


Fig. 5. Comparison of runtime for area and wire length optimization.

in 3D space in the second stage. At the end of the second stage, the blocks are assigned to discrete layers. There are mismatches between optimization in continuous and discrete vertical space. Though the overlaps in continuous space are bounded by constraining the overlap to 5%, they increase after layer assignment. The third stage smooths transitions between continuous space and discrete multi-layer space by integrating layer assignment with global optimization. As shown in the Table I, the overlaps on each layer are generally below 0.8% after the third stage, easing legalization. At this point, interconnect lengths are within approximately 5% of the final results. For ami49, some reduction of wire length is possible after legalization because improved orientations are determined during legalization. These results suggest that gradual transition from continuous 3D space to discrete, legalized prevents layer assignment from causing large displacements in block positions. The three-stage process can therefore support force-directed optimization by avoiding degradation to solution quality during legalization.

IV.B. Impact of Wire Length and Area Optimization

This section compares the quality of results produced by 3D-STAF and CBA [12] when both are configured to optimize area and wire length but not peak temperature. We consider temperature optimization in the following section. 3D-STAF has a number of force control parameters that vary by optimization stage. These parameters are controlled by the α and β values explained in Section II-C.1. They have different values during each stage of optimization. 3D-STAF relies on adaptation that is controlled by these empirically-determined parameters. However, these values are general across a wide range of problem instances: the same parameters were used for all benchmarks whenever comparing with past work.

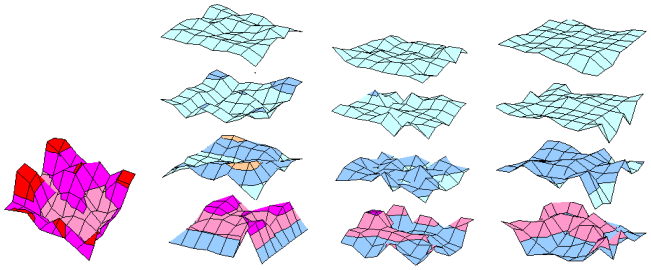
As shown in Table II, compared with CBA, 3D-STAF degrades area by 4%, improves wirelength by 12%, has little impact on via count, and completes execution in 69% of the time required by CBA, when averaged over all benchmarks.

Figure 5 illustrates the relationship between problem size and runtime for CBA and 3D-STAF. 3D-STAF is able to solve large problem instances with sublinear increase in runtime. In summary, when optimizing area and wire length, 3D-STAF produces results that are comparable with state-of-the-art related work in significantly less run time and, more importantly, a slower rate of growth in runtime with increasing problem size.

IV.C. Temperature Optimization With Fixed Leakage Power

This section compares 3D-STAF with CBA when optimizing area, wire length, via count, and temperature using fixed leakage power consumption. To permit direct comparison with CBA [12], we used the same dynamic power values and thermal parameters. Although these specific parameters can lead to high temperatures for some benchmarks, it would be possible to reduce these temperatures in real ICs by using more efficient cooling technologies. We assume that leakage power is a fraction (about 10%) of dynamic power and add this fraction to obtain the total power dissipated by each block. The next section examines the impact of considering temperature-dependent leakage power.

To minimize peak temperature, layer lateral power density should be nearly uniform and blocks with higher power densities should be



(a) stage 1 (b) stage 2 (c) stage 3 (d) legalization

Fig. 6. Power distribution during each optimization stage for *n100*.

TABLE III
COMPARISON FOR TEMPERATURE OPTIMIZATION

Circuit	CBA					3D-STAF				
	Area (mm^2)	HPWL (mm)	Via (count)	Temp (°C)	Time (s)	Area (mm^2)	HPWL (mm)	Via (count)	Temp (°C)	Time (s)
ami33	43.2	23.9	119	212.4	486	41.5	24.2	116	201.3	227
ami49	1672.6	516.4	251	225.1	620	1539.4	457.3	208	230.2	336
n100	6.6	122.9	1145	172.7	4535	6.6	91.5	753	156.8	341
n200	6.6	203.7	2217	174.7	6724	6.2	167.8	1356	164.6	643
n300	10.4	324.9	2563	190.8	18475	9.3	236.7	2173	168.2	1394
Aggregate relative to CBA						-6%	-16%	-12%	-6%	-75%

separated from each other and placed near the heat sink. 3D-STAF spreads the blocks gradually based on the thermal gradient. Figure 6 shows the power distribution in each optimization step for test case *n100*. In the lateral spreading stage, block depth is ignored but blocks are spread in the x–y plane. In the successive global packing stage, blocks are spread to 3D space and hot blocks are pushed toward the heatsink. At the end of Stage 2, the power densities on planes are not evenly distributed due to block overlaps. In Stage 3, overlaps are well controlled, balancing power density. With the final legalization, power densities are fairly even on each layer and the high power density blocks are near the heatsink.

Table III compares 3D-STAF with CBA [12]. We run temperature-aware CBA and determine the thermal profile using our thermal model. On average, the peak temperature obtained by 3D-STAF is 6% cooler than CBA and 3D-STAF outperforms CBA by 6% and 16% in area and wirelength. CBA increases area by about 20% to improve the thermal profile. 3D-STAF adaptively guides block motion based on thermal profile. Figure 8 shows a four-layer packing with the corresponding power distribution and thermal profile. The blocks with the high power density are assigned to the bottom layer to reduce peak temperature.

As shown in Figure 7, in addition to producing better results, 3D-STAF requires less CPU time than CBA. For test case *n300* with 300 blocks, CBA needs more than 5 hours to get a good solution. 3D-STAF produces a superior result in 25 minutes. More importantly, CPU time increases slowly with problem size for 3D-STAF.

The results in this section indicate that, in comparison with a state-of-the-art temperature-aware 3D floorplanning technique, 3D-STAF better optimizes multiple costs including temperature and that its run time scales well with problem size.

IV.D. Impact of Temperature-Dependent Leakage Power Model

In this section, we evaluate the impact of modeling the interdependence between leakage power consumption and temperature during 3D temperature-aware floorplanning. Higher temperatures increase leakage power, which in turn further increases temperature. In deep submicron processes, this effect can have a significant impact on the final temperature profile.

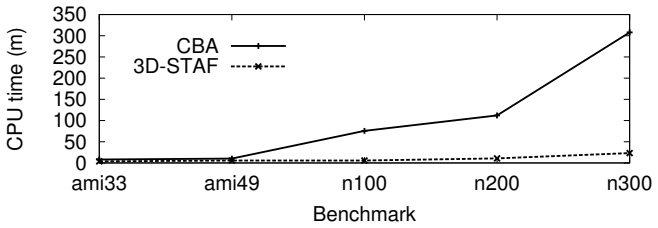


Fig. 7. Comparison of runtime for temperature, area, via count, and wire length optimization.

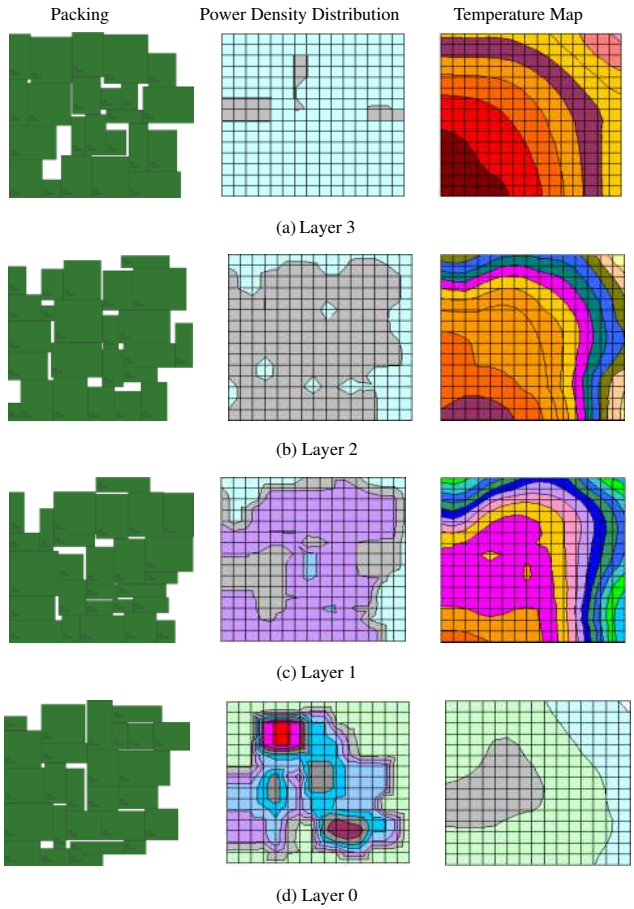


Fig. 8. Four-layer packing and thermal distribution for $n100$.

We set leakage power to 10% of total power at 300 K and use the temperature-dependent leakage model described in Section III-C.

Table IV shows the results produced by 3D-STAF when the interdependence of temperature and leakage are neglected (in the 3D-STAF columns, left) and considered (in the 3D-STAF-TDLP columns, right). *Temp* shows the peak temperature estimated when the dependence of leakage on temperature is ignored. *Temp (feedback)* shows the temperature when the temperature–leakage feedback loop is considered.

As shown in Table IV, neglecting leakage–temperature interdependence results in approximately 32% error in peak temperature estimation. More importantly, using temperature-dependent leakage power estimation during optimization allows a 20% reduction in peak temperature without degrading other costs such as area, wire length, and via count. However, considering this effect increases floorplanning run time by 56%. This is mainly due to the increase

TABLE IV
IMPACT OF CONSIDERING DEPENDENCE OF LEAKAGE POWER ON TEMPERATURE

Circuit	3D-STAF						3D-STAF-TDLP				
	Area (mm^2)	HPWL (mm)	Via (count)	Temp (°C)	Temp (feedback) (°C)	Time (s)	Area (mm^2)	HPWL (mm)	Via (count)	Temp (feedback) (°C)	Time (s)
ami33	39.6	23.1	108	204.4	284.5	280	39.8	22.4	103	236.7	356
ami49	1557.2	483.9	216	227.2	308.4	352	1436.7	476.4	217	257.8	419
n100	6.5	89.4	709	158.0	247.9	355	6.3	90.5	718	205.8	613
n200	6.1	164.1	1272	161.3	259.7	697	6.1	166.1	1235	197.6	1675
n300	9.4	228.3	2204	164.9	270.4	1482	9.5	224.5	2204	218.7	1820
Aggregate				-32%			-2%	-1%	-1%	-19%	+56%

in thermal analysis run time imposed by iteratively considering leakage–temperature feedback until convergence. We conclude that temperature-aware floorplanners should consider this effect.

V. CONCLUSION

In this paper, we propose the first force-directed solution to 3D heterogeneous macro-block temperature-optimized floorplanning problem. By integrating layer assignment with the global optimization process, our proposed flow can smooth transition from continuous vertical space to discrete IC layers. The proposed legalization methods optimize the orientations of blocks during the legalization process. The closed feedback loop between temperature and leakage power consumption is appropriately modeled. Experimental results indicate that 3D-STAF produces better results than a state-of-the-art technique in area, wire length, temperature, and via count while requiring substantially less run time.

REFERENCES

[1] B. Black, D. W. Nelson, C. Webb, and N. Samra., "3D processing technology and its impact on IA32 microprocessors," in *Proc. Int. Conf. Computer Design*, Oct. 2004, pp. 316–318.

[2] "International Technology Roadmap for Semiconductors," 2006, http://public.itrs.net.

[3] A. Chandrakasan, W. Bowhill, and F. Fox, *Design of High-Performance Microprocessor Circuits*. IEEE Press, 2001.

[4] Z. Li, X. Hong, Q. Zhou, Y. Cai, J. Bian, H. H.Yang, V. Pitchumani, and C.-K. Cheng, "Hierarchical 3-D floorplanning algorithm for wirelength optimization," *IEEE Trans. Circuits and Systems I*, 2007, to appear.

[5] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu, "Corner block list: An effective and efficient topological representation of non-slicing floorplan," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2000, pp. 8–12.

[6] H. Murata, K. Fijiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 12, pp. 1518–1524, Dec. 1996.

[7] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitanid, "Module packing based on the BSG-structure and IC layout applications," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 6, pp. 519–530, June 1998.

[8] G.-M. Wu, S.-W. Wu, Y.-W. Chang, and Y.-C. Chang, "B*-trees: A new representation for non-slicing floorplans," in *Proc. Design Automation Conf.*, June 2000, pp. 458–463.

[9] J.-M. Lin and Y.-W. Chang, "TCG: a transitive closure graph based representation for non-slicing floorplans," in *Proc. Design Automation Conf.*, June 2001, pp. 764–769.

[10] Y. Deng and W. P. Maly, "Interconnect characteristics of 2.5D system integration scheme," in *Proc. Int. Symp. Physical Design*, Apr. 2001, pp. 341–345.

[11] P. H. Shiu, R. Ravichandran, S. Easwar, and S. K. Lim, "Multi-layer floorplanning for reliable system-on-package," in *Proc. Int. Symp. Circuits & Systems*, May 2004, pp. 69–72.

[12] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2004, pp. 306–313.

[13] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "Interconnect and thermal-aware floorplanning for 3D microprocessors," in *Proc. Int. Symp. Quality of Electronic Design*, Mar. 2006, pp. 98–104.

[14] B. Obermeier and F. Johannes, "Temperature aware global placement," in *Proc. Asia & South Pacific Design Automation Conf.*, Jan. 2004, pp. 143–148.

[15] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2003, pp. 86–89.

[16] W. Huang, E. Humenay, K. Skadron, and M. R. Stan, "The need for a full-chip and package thermal model for thermally optimized IC designs," in *Proc. Int. Symp. Low Power Electronics & Design*, Aug. 2005, pp. 245–250.

[17] A. Gupta, N. D. Dutt, F. J. Kurdahi, K. S. Khouri, and M. S. Abadir, "LEAF: A system level leakage-aware floorplanner for SoCs," in *Proc. Asia & South Pacific Design Automation Conf.*, Jan. 2007, pp. 274–279.

[18] K. Vorwerk, A. Kennings, and A. Vannelli, "Engineering details of a stable analytic placer," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2004, pp. 573–580.

[19] J. Barnes and P. Hut, "A hierarchical o(n log n) force-calculation algorithm," *Nature*, vol. 324, Dec. 1986.

[20] M. D. Moffitt, A. N. Ng, I. L. Markov, and M. E. Pollack, "Constraint-driven floorplan repair," in *Proc. Design Automation Conf.*, June 2006, pp. 1103–1108.

[21] J. Cong and M. Xie, "A robust detailed placement for mixed-size IC designs," in *Proc. Asia & South Pacific Design Automation Conf.*, Jan. 2006, pp. 188–194.

[22] G. S. Ohm, "The Galvanic circuit investigated mathematically," 1827.

[23] Y. Zhan and S. S. Sapatnekar, "A high efficiency full-chip thermal simulation algorithm," in *Proc. Int. Conf. Computer-Aided Design*, Oct. 2005.

[24] P. Li, L. T. Pileggi, M. Ashghi, and R. Chandra, "Efficient full-chip thermal modeling and analysis," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2004, pp. 319–326.

[25] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated Space and Time Adaptive Chip-Package Thermal Analysis," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Jan. 2007.