Fixed-outline Thermal-aware 3D Floorplanning

Linfu Xiao†, Subarna Sinha‡, Jingyu Xu‡ and Evangeline F.Y. Young†
† Department of CSE, The Chinese University of Hong Kong, Shatin N.T., Hong Kong
‡ Advanced Technology Group, Synopsys, U.S.A
e-mail: lfxiao@cse.cuhk.edu.hk

Abstract— In this paper, we present a novel algorithm for 3D floorplanning with fixed outline constraints and a particular emphasis on thermal awareness. A computationally efficient thermal model that can be used to guide the thermal-aware floorplanning algorithm to reduce the peak temperature is proposed. We also present a novel white space redistribution algorithm to dissipate hotspot. Thermal through-silicon via (TSV) insertion is performed during the floorplanning process as a means to control the peak temperature. Experimental results are very promising and demonstrate that the proposed floorplanning algorithm has a high success rate at meeting the fixed-outline constraints while effectively limiting the rise in peak temperature.

I. Introduction

As VLSI circuits keep shrinking following Moore's Law, packaging and interconnection technologies are required to keep up. The 3D IC is gaining a lot of interest as a viable solution to help maintain the pace of system demands on scaling, performance, and functionality. The benefits include system-size reduction, performance enhancement due to shorter wirelength, power reduction and the potential for hetero-integration.

A good 3D floorplanner has been identified as being a key component to help realize some of the benefits of 3D integration. Smart algorithms are needed to partition the blocks among the different die layers for minimizing the wirelength while limiting the temperature increase. Consequently, 3D floorplanning has received a lot of interest in the past few years and numerous solutions have been proposed ([1], [2], [3]). These techniques either simultaneously consider all the die layers while determining the optimal location of the blocks or prepartition the blocks among the die layers based on some estimations and then work on each of the die layers separately. Typical objectives considered so far include wirelength, area, peak temperature and/or number of through-silicon vias (TSVs). Existing solutions on thermal management during 3D floorplanning include thermal TSV insertion during or after floorplanning [4], [5] and [7]). The thermal TSVs do not provide any connectivity and are primarily inserted to help with heat dissipation.

A common limitation of the previous methods of 3D floorplanning is that they are focused on area and/or wirelength minimization with or without thermal considerations. This can be a serious limitation as modern floorplanners often have to work with a fixed die size constraint, or with a fixed outline constraint in low-level design of hierarchical floorplanning flow [11]. For such cases, a floorplan with pure area and/or wirelength minimization without any fixed outline constraints may be useless because it may not fit in the given outline. The existing solutions for thermal TSV insertion do not consider the key fact that the thermal TSVs need to connect adjacent silicon layers to effectively dissipate heat and hence need to consider the whitespace restrictions on adjacent layers for a valid solution. Thus, a while space redistribution algorithm is performed to gather as much as possible white spaces at the same location for all layers after floorplan realization. In this paper, we address the above issues with a novel thermal-aware fixed-outline 3D floorplanning algorithm. The thermal-aware solution also considers thermal TSV insertion during floorplanning, thereby increasing the potential for peak temperature reduction.

The paper is organized as follows. Section 2 provides a brief overview of thermal modeling and also presents an approximate thermal model for use during floorplanning. Section 3 discusses in detail the thermal-aware 3D floorplanning. This section also describes the thermal TSV insertion algorithm. Experimental results are given in Section 4. The paper ends with conclusions in Section 5.

II. Fast Temperature Estimation of 3D ICs

In this section, we present a computationally efficient thermal model that has good fidelity with an accurate thermal model for 3D ICs and can be used to guide thermal-aware 3D floorplanning.

The Hotspot thermal model [9] is used as the starting point for the detailed model for 3D ICs. The model used in Hotspot is briefly presented below for the reader's convenience. Figure 1 shows a typical package for 2D ICs, as it is modeled in Hotspot. Heat generated from the active silicon layer is conducted through the silicon die to the Thermal Interface Material (TIM), heat spreader and heat sink, then convectively removed to the ambient air. This is the primary heat transfer path and represents what is captured in the thermal models for 3D ICs as well. A grid-based model is used to determine the temperature profile. Typically, the entire stack is gridded as shown

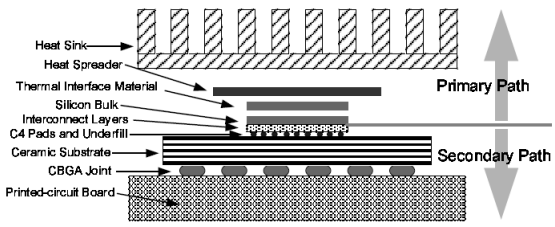


Fig. 1. Heat Flow in 2D ICs

in Figure 2. Each grid cell maps to a node in the thermal circuit with lateral and vertical thermal resistances connecting the nodes. The power dissipated in each silicon grid cell

978-1-4244-5766-3/10/$26.00 ©2010 IEEE

is modeled as a current source connected to the corresponding node. The package-to-air thermal resistance is calculated from specific heat-sink configurations and ambient conditions. The thermal resistances are proportional to the thickness of the material and inversely proportional to the cross-sectional areas across which the heat is being transferred. The temperature at each node in the network is then equivalent to the voltage value calculated at the node. The thermal model for 3D ICs

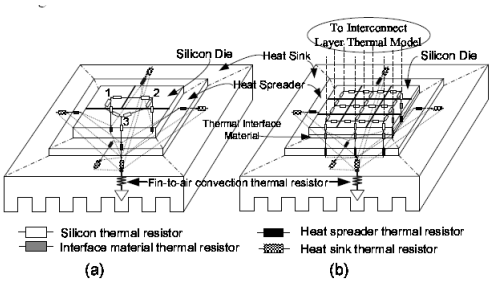


Fig. 2. Hotspot Grid Model

is based on similar principles but needs to consider the impact of the multiple die layers. Figure 3 illustrates how the stack can be extended for a 3-D chip consisting of four die layers, interconnect layers , TIM layer and package layer (where die1 and die2 are in the face-to-face configuration and the rest are in face-to-back configuration [6]). Next, we discuss the various approximations we developed to build a computationally efficient thermal model that can be used by the floorplanning algorithm.

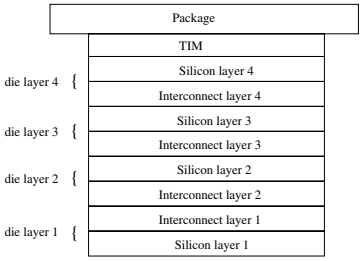


Fig. 3. Layers considered in the Thermal Modeling of 3D ICs

The number of grids is a key factor in determining the speed of the thermal model. In this paper, the grid size is set to be equal to the smallest dimension among the given blocks and was found to provide the right balance between speed and accuracy. Increasing the grid size could result in inaccuracies in the model, whereas reducing the grid size could result in an unnecessary slowdown. The next simplification is based on the empirical observation that the metal and dielectric stack can be replaced by a single thicker dielectric layer. Thus, if the total thickness of the metal (dielectric) layer is t_m (t_d) um, the interconnect stack (i.e., including both metal and dielectric layers) is $t_m + t_d$ um, then a thermal model that has a single dielectric layer of t_d um and has the same interconnect via density[1] as the dielectric layers will introduce negligible error in predicting the peak temperature.

[1]For floorplanning, the interconnect via density value has to be estimated. We used data from existing designs to estimate values for the interconnect via densities.

The model based on the above simplifications is still not fast enough to be used in every iteration of the floorplanning algorithm. The simplified vertical model proposed by Cong et. al. [1] was considered. Figure 4(b)-(c) captures the essence of the simplified vertical model where it is assumed that the heat flows down along each grid column with no lateral heat flow. Thus, only the vertical heat dissipating path is considered. As

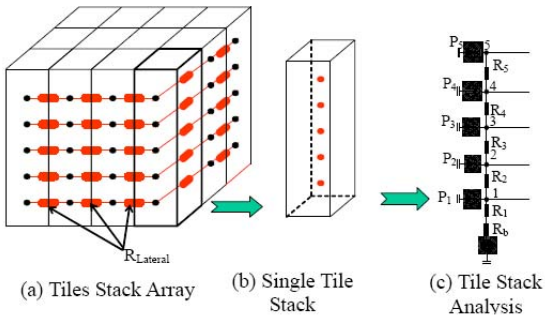


Fig. 4. Simplified Vertical Model

a result of this simplification, the temperature increase can be written in an Elmore-delay-like closed-form formula. Some additional assumptions about the constancy of the thermal resistances during the floorplanning process produces the following formula which is used during the floorplanning steps in [1]:

$$T = \sum_{i=1}^{k} (P_i \sum_{j=1}^{i} R_j),$$

where k is the number of die layers and P_i is the current source at node i and R_j is the (vertical) thermal resistance of layer j. This simplified model while being fast can produce inaccuracies that can result in inferior solutions being picked during the floorplanning process. To address this problem, an interpolation based model is proposed that is still computationally efficient but does not ignore the lateral heat flow.

Algorithm *interpolation* describes the steps used to compute the temperature distribution of the given floorplan. The basic idea is described below. Let the number of grids [2] used for computing the accurate temperature be $n \times n$ and L denote the number of die layers. The **first step** is to compute the temperature profiles using the simplified vertical model for different griddings of the floorplan. Let T_n denote the $n \times n \times L$ temperature matrix when the floorplan is gridded using $n \times n$ grids. This captures the case when no lateral flow is considered between the $n \times n$ grid cells at each layer. Subsequently, the temperature profile is computed using the simplified vertical model when the floorplan is divided into $n/2 \times n/2$ grids, $n/4 \times n/4$ grids and so on. $T_{n/2}(T_{n/4}, T_{n/8} \ldots, T_1)$ implies perfect lateral flow(internal lateral resistance is zero) in adjacent $2 \times 2, (4 \times 4, 8 \times 8, \ldots, n \times n)$ cells of the $n \times n$ grid. In the sequel, we will refer to a grouping of adjacent grid cells of the $n \times n$ grid as a region. The **second step** is to compute the final temperature matrix by suitably interpolating between $T_n, \ldots, T_1$. The key idea here is to choose appropriate coefficient values $\alpha_i^j (\in [0, 1])$ for each interpolation (i denotes the layer number and $j \times j$ denotes the number of grids used in a

[2]The algorithm presented below requires n to be a power of 2; however, it can be easily extended to work for other values as well

Algorithm 1 interpolation

1: //calculate temperature profiles using different grid size.
2: partition chip into $(n * n)$ grids, calculate temperature profile T_n using simplified vertical model
3: partition chip into $(n/2*n/2)$ grids, calculate temperature profile T' using simplified vertical model, then mapping T' to $n * n$ grids profile $T_{n/2}$.
4: partition chip into $(n/4*n/4)$ grids, calculate temperature profile T' using simplified vertical model, then mapping T' to $n * n$ grids profile $T_{n/4}$.
5: ...
6: partition chip into $(1 * 1)$ grid, calculate temperature profile T' using simplified vertical model, then mapping T' to $n * n$ grids profile T_1.
7: //interpolate gradually.
8: //interpolation between T_n and $T_{n/2}$, which means we consider lateral flow for each 2*2 region.
9: **for** each grid, r denotes the row index, c denotes the column index, i denotes the layer index **do**
10: calculate $\alpha_i^{n/2}$
11: $\hat{T}_1[r][c][i] = \alpha_i^{n/2} * T_n[r][c][i] + (1 - \alpha_i^{n/2}) * T_{n/2}[r][c][i]$
12: **end for**
13: //interpolation between $\hat{T}_1$ and $T_{n/4}$, which means we consider lateral flow for each 4*4 region.
14: **for** each grid **do**
15: calculate $\alpha_i^{n/4}$
16: $\hat{T}_2[r][c][i] = \alpha_i^{n/4} * \hat{T}[r][c][i] + (1 - \alpha_i^{n/4}) * T_{n/4}[r][c][i]$
17: **end for**
18: ...
19: //interpolation between $\hat{T}_{k-1}$ and T_1, which means we consider lateral flow for each $n * n$ region(whole chip).
20: **for** each grid **do**
21: calculate α_i^1
22: $\hat{T}_k[r][c][i] = \alpha_i^1 * \hat{T}_{k-1}[r][c][i] + (1 - \alpha_i^1) * T_1[r][c][i]$
23: **end for**

interpolation step). α_i^j is related to the lateral and vertical heat flow ratio for each grid.

The interpolation is necessary to compute the lateral resistance more accurately without expending too much computational resources. Suppose we are interpolating between two temperature profiles T_n and $T_{n/2}$. During the calculation of T_n, it is assumed that the lateral resistance between adjacent grid nodes is infinite. On the other hand, the calculation of $T_{n/2}$ assumes zero lateral resistance between adjacent nodes (thus, 2×2 adjacent grid cells are grouped to form a single grid cell during the calculation of $T_{n/2}$). The actual lateral resistance is a finite value greater than zero. Thus, the interpolation step attempts to find a balance between the two extreme situations based on heat dissipating paths. In our implementation,

$$\alpha_i^j = Rleff_i^j/(Rleff_i^j + Rveff_i^j),$$

where $Rleff_i^j$ and $Rveff_i^j$ denote the lateral effective resistance and the vertical effective resistance, respectively. The value of α_i^j depends on the number of grids $(j \times j)$ used to partition the floorplan and the number of die layer i. $Rleff_i^j$ and $Rveff_i^j$ are calculated based on the resistance of the heat dissipating paths in the horizontal and vertical directions for a particular grid size, respectively. Multiple interpolations are needed to capture the lateral heat flow through the span of the silicon layer. The proposition below illustrates the calculations for the effective resistances, and then an example is given to demonstrate the algorithm for computing the temperature.

Proposition *In our model (Fig. 3), there are 4 die layers and die layers 1 and 2 are in a face-to-face bonding configuration. Suppose we are interpolating between two temperature profiles T_{2G} and T_G for the blocks on die layer 1, for a given value of G. The lateral (for both directions) and vertical resistances for each grid are given by the following equations:(Fig. 5(a))*

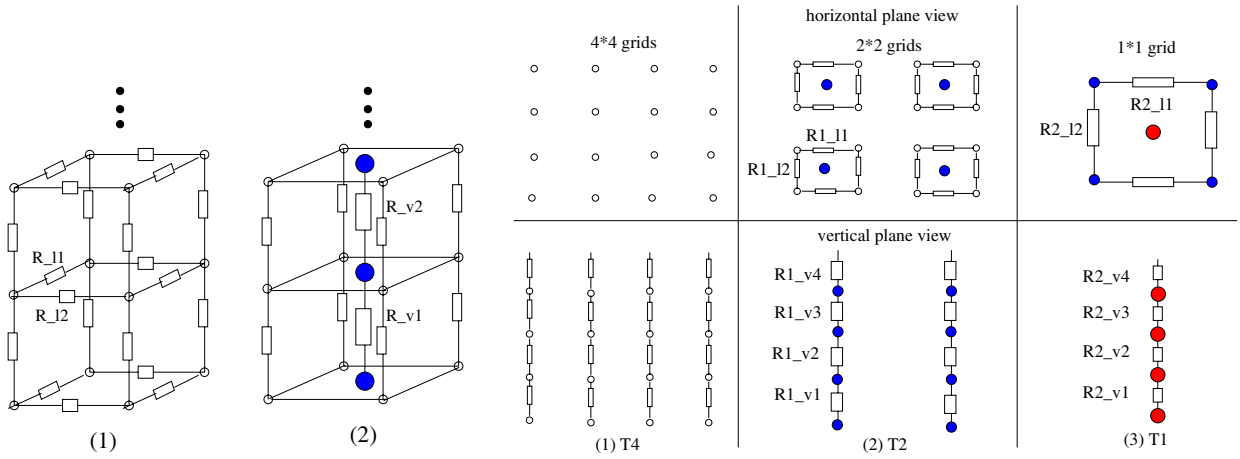
$R_{l1} = (W/2G)/(C_{si} * T_{si} * H/2G) = W/(C_{si} * T_{si} * H)$.
$R_{l2} = (H/2G)/(C_{si} * T_{si} * W/2G) = H/(C_{si} * T_{si} * W)$.
$R_{v1} = T_{si}/(C_{si} * W/G * H/G) + T_{di}/(C_{di} * d_1 * W/G * H/G) + T_{di}/(C_{di} * d_2 * W/G * H/G)$.
$R_{v2} = T_{si}/(C_{si} * W/G * H/G) + T_{di}/(C_{di} * d_3 * W/G * H/G)$.
$R_{v3} = T_{si}/(C_{si} * W/G * H/G) + T_{di}/(C_{di} * d_4 * W/G * H/G)$.
$R_{v4} = T_{si}/(C_{si} * W/G * H/G) + T_{TIM}/(C_{TIM} * W/G * H/G)$.

$W(H)$ denotes the width(height) of the floorplan area, $C_{si}(T_{si})$, $C_{di}(T_{di})$, $C_{TIM}(T_{TIM})$ denote the conductivity(thickness) of the silicon layer, dielectric layer and TIM layer respectively. G denotes the the number of grids. d_1, d_2, d_3 and d_4 denotes the via density value for each grid in dielectric layer 1, 2, 3 and 4. Each node in (Fig. 5(a).(1)) has two lateral heat flow paths (through both R_{l1} and R_{l2}), so we choose $Rleff$ equals to its two adjacent lateral resistors in parallel $R_{l1}||R_{l2}$. The vertical heat flow path is through all the vertical resistors in the 2×2 region of the $2G \times 2G$ grid (Fig. 5(a).(2)); hence $Rveff$ is computed by treating all the corresponding vertical resistors in series $(R_{v1} + R_{v2} + R_{v3} + R_{v4})$. (Note that all the nodes in 2×2 region correspond the same $Rveff$ since their vertical heat flow path is the same).

Example 1 *Figure 5(b) gives an example of the temperature calculation for the case when the floorplan is partitioned into 4×4 grids. First, T_4, T_2 and T_1 are calculated using the simplified vertical model when the floorplan is gridded using $4 \times 4, 2 \times 2$ and 1×1 grids, respectively. Then, $\hat{T}$ is computed by interpolating between T_4 and T_2 at each grid location (second column in figure). For each grid, using the above equation, $Rleff$ equals to $R1_{l1}||R1_{l2}$, while $Rveff$ equals to $R1_{v1} + R1_{v2} + R1_{v3} + R1_{v4}$. α_1^2 (for layer 1) is calculated from the above values for $Rleff$ and $Rveff$. The final temperature profile T is computed by interpolating between $\hat{T}$ and T_1. In this case, $Rleff$ equals to the $R2_{l1}||R2_{l2}$ and the effective vertical resistance $Rveff$ equals to $R2_{v1} + R2_{v2} + R2_{v3} + R2_{v4}$ (referring the third column).*

Figure 6 shows an example which compares the temperatures predicted by the accurate model, the simplified vertical model and the proposed interpolation model for 30 different floorplans. The correlation between the accurate model and the simplified vertical model is 0.82, while the correlation between the accurate model and the proposed interpolation model is 0.97. It is clear qualitatively that the interpolation model provides more accurate results when compared to the simplified vertical model, because it captures more accuracy of the lateral flow. This is also true for the case when a quantitative comparison is done. It should be noted that both the simplified vertical model and the interpolation model provide the temperature of the blocks relative[3] to the heat sink and hence the numeri-

[3]The package is a very good heat conductor as the material is copper. Therefore, it will introduce significant errors if the lateral heat flow among the grids in the package is ignored. A reasonable approximation for the package is to treat it as an isothermal object. Thus, the temperature difference between the package and the ambient is $\delta(T) = P_{total} * R_{package}$. P_{total} is the power summation of all the blocks and $R_{package}$ is the effective resistance of package. These two values are independent of the floorplan, so the temperature difference between package and ambient is also fixed for different floorplan in



(a) Example for Effective Resistance (b) Example for Interpolation Model

Fig. 5. Examples

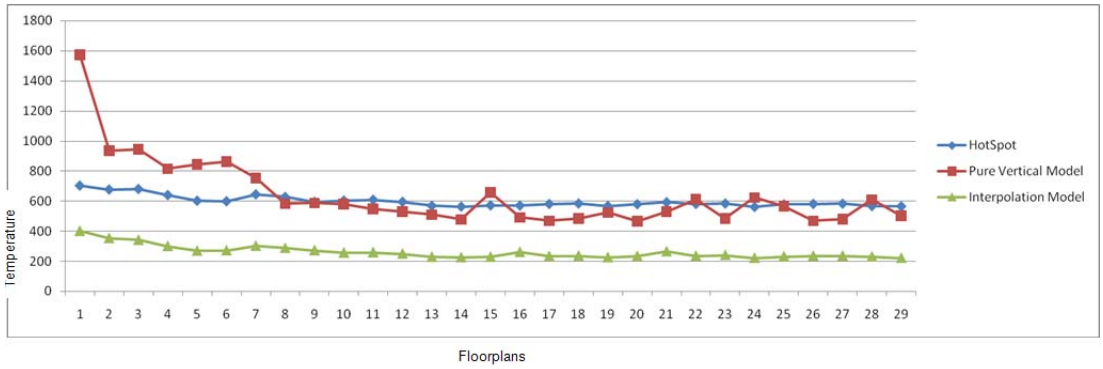


Fig. 6. Correlation between accurate model, interpolation model and simplified vertical model

cal values don't match exactly. The key idea is to capture the trends rather than the absolute number.

III. 3D Floorplanning with Fixed Outline and Thermal Constraints

Here, we present the thermal-aware 3D fixed outline floorplanning algorithm. The above-mentioned floorplanning problem can be formulated as follows. Let $B = \{b_i | 1 \leq i \leq n\}$ be a set of hard blocks, and each block b_i has width and height of w_i and h_i, respectively. Each block is free to rotate and/or flip. Let W and H denote the desired width and height of the 3D IC[4]. Our objective is to find a coordinate (x_i, y_i, l_i) for the lower-left corner of each block b_i, such that $0 \leq x_i \leq W - w_i$, $0 \leq y_i \leq H - h_i$, $1 \leq l_i \leq L$ and no two blocks overlap. In addition, the wirelength and the peak temperature needs to be minimized. We say a floorplan is successful if all the fixed outline constraints are satisfied.

this assumption.

[4]In our experiments, we adopt the technique proposed by [11] to determine the width and height, given the allowable white space in the floorplan. Assume the sum of the area of all the block is A, the number of die layers of the 3D IC is L, the maximum allowable fraction of the white space is ϵ and the given aspect ratio is γ. Then, the width W and height H of the 3D IC can be expressed as follows:

$$W = ((1+\epsilon)A\gamma/L)^{1/2}; H = ((1+\epsilon)A/\gamma L)^{1/2}.$$

The proposed floorplanning algorithm is a simulated annealing based algorithm that seeks to minimize the wirelength and the peak temperature while meeting the fixed outline constraints. The simulated annealing is performed in two phases: during the first phase, the primary focus is on meeting the fixed outline constraints and minimizing the wirelength, while the second phase emphasizes minimizing the peak temperature while maintaining the fixed outline constraints. An array of sequence pairs (one sequence pair for each die layer) is used to represent the position of blocks during the simulated annealing steps. The perturbation scheme includes both inter-layer and intra-layer moves. A more in-depth presentation of the 2-phase approach is presented in the next section.

A. Proposed 2-Phase Algorithm

Before getting into the details of the 2-Phase algorithm, we discuss why a single simulated annealing run with an appropriate cost function does not yield satisfactory results. An obvious solution to the above fixed-outline thermal-aware 3D floorplanning problem could be to extend the cost function used in a simulated annealing based fixed-outline 2D floorplanning solution (such as Parquet [8]) to consider 3D-specific fixed outline constraints and wirelength and also include temperature in the cost function. In that case, the cost function would be written as follows:

$$cost = \alpha * narea + \beta * nAR + \gamma * nwl + \eta * c_T,$$

where $narea$, nAR, nwl and c_T are the normalized area, aspect ratio, wire length and peak temperature (normalized to the maximum original peak temperature) values, respectively. The combination of $narea$ and nAR account for the fixed-outline constraints[5]. The values of the coefficients $\alpha, \beta, \gamma, \eta$ are all less than 1. The first three terms in the cost function are the same ones used for fixed-outline 2D floorplanning in Parquet [8]. Thus, adding a temperature term to the cost function would seem like a natural extension for thermal-aware fixed outline 3D floorplanning. Our experimental results presented in a later section show that such a cost function does not necessarily result in good packing results while producing low peak temperature values. The above observation holds even if the temperature component is ignored during the early phases of simulated annealing and only used later in the annealing steps. This is especially true for the examples with high aspect ratio and low white space, where it is not possible to get a good η value where both the peak temperature is significantly reduced and the packing is successful. In the sequel, we will refer to the above algorithm as the *1-phase* algorithm.

To address the above concerns, we propose a *2-phase* algorithm. The basic idea of the *2-phase* algorithm is that the simulated annealing process is applied twice. During the **first phase** (henceforth, referred to as 3D-1), the primary focus is on optimizing the area, aspect ratio and wire length only. The cost function is set up as $cost = \alpha * narea + \beta * nAR + \gamma * nwl$ as in Parquet. The best solution at the end of this phase is usually successful in meeting the fixed outline constraints with a small wirelength value. This solution is used as the starting point of the second phase annealing. In the **second phase** (henceforth, referred to as 3D-2), the primary focus is on peak temperature optimization. The solution is penalized only if it does not meet the fixed outline constraints. The cost function in this case is given as

$$cost = max(0, (Width - W)) + max(0, (Height - H)) + \eta * c_T.$$

The annealing schedule for the two phases is shown in Figure 7. The initial annealing temperature for the second phase is set to significantly lower than the initial annealing temperature for the first phase. This is done to avoid completely losing the results obtained at the end of 3D-1.

In a later section, we will present experimental results to demonstrate that the *2-phase* algorithm can consistently achieve superior packing, wirelength and temperature results.

B. Thermal TSV Assignment

During 3D-2, besides including temperature in the cost function, thermal TSV insertion is considered in order to reduce the peak temperature. The key idea is that opportunistic thermal TSV insertion will provide improved conduits for heat flow (equivalent to lowering the vertical resistance) and result in lower peak temperatures for the die layers further away from the heat sink. However, it is important that the constraints imposed by thermal TSVs are satisfied to achieve a realistic solution. As mentioned earlier, it is necessary to reserve whitespace on the adjacent die layer closer to the heat-sink to insert a thermal TSV and thereby ensure that a high thermal conducting path exists between the silicon layers to effectively dissipate the heat. Our thermal TSV assignment process is designed with the above constraint in mind. The assignment is outlined below.

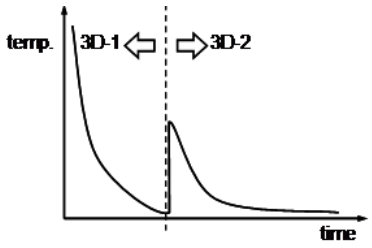


Fig. 7. Plot of Temperature Versus Time.

1. Re-distribute the whitespace with due consideration to the high temperature blocks (hot blocks).
2. Starting from the die layer k farthest away from the heat sink,
   (a) Compute the overlap area between the white space region in layer k and the white space in its adjacent layer.
   (b) Set the TSV density value of dielectric layer sandwiched in between the silicon layers of the corresponding die layers to a pre-determined value. In our experiments, we set the value to 1% to limit congestion.

The basic method for incorporating TSV assignment during floorplanning is as follows. For each iteration of simulated annealing when the impact of thermal TSV assignment needs to be considered, the peak temperature is computed assuming the thermal TSVs are inserted as described above. The floorplan is then evaluated using the cost function for 3D-2. It should be noted that the intermediate thermal TSV assignments are not saved. The idea is to select floorplans during simulated annealing that are more amenable to thermal TSV assignment around the hot blocks and hence lower peak temperatures. Finally, thermal TSV assignment is performed on the floorplan obtained at the end of the simulated annealing process; this serves as the final result.

Experimental data presented in a later section illustrates that considering thermal TSV assignment during the floorplanning process results in greater peak temperature reductions. However, considering it all through 3D-2 can result in very high runtimes. We have found, based on experiments, that restricting thermal TSV assignment to the later stages of 3D-2 provides the best compromise in terms of solution quality and runtime.

B.1 White Space Redistribution (WSR)

The WSR algorithm (outlined in Algorithm 2) seeks to assign more white spaces around the hot blocks and maximize the overlap whitespace area between adjacent die layers, while keeping the topology fixed. The key principle of the WSR algorithm is based on the observation that the packing after evaluation for the sequence-pair representation is lower-left

[5]The area, aspect ratio and wirelength calculation are now adjusted to reflect the respective values for 3D floorplanning. In addition, the concept of spatial slack and slack based moves introduced in [11] were also extended to apply for 3D floorplanning. One major difference from Parquet is we apply slack-based moves at high temperatures and restrict to slack-based swaps at low temperature. This has improved our success rate over Parquet even for the 2D case [12].

compact. This implies there could be a lot of white spaces wasted in the upper-right corner under the fixed-outline packing strategy. Due to fixed outline constraints, the bounding

Algorithm 2 WSR

Require: Fixed-outline width, height, sequence pair, hot blocks array(array length is N)

1: Find the lower-left packing coordinates (x_1, y_1) for all the blocks. Bounding width(height) is W(H).
2: Find the upper-right packing coordinates (x_2, y_2) for all the blocks.
3: If required width(height) RW(RH) $>$ W(H), adjust the coordinates (x_2, y_2).
$x_2 = x_2 + (RW - W)$
$y_2 = y_2 + (RH - H)$
4: for each block A
Compute the summation of total temperature of the hot blocks T_{total},
Compute the summation of temperature T_r for hot blocks located on the right side of it,
$\alpha_x = T_r/T_{total}$
$x_a = x_1 * \alpha_x + x_2 * (1 - \alpha_x)$;
Compute the summation of temperature T_t for hot blocks located above it,
$\alpha_y = T_t/T_{total}$
$y_a = y_1 * \alpha_y + y_2 * (1 - \alpha_y)$;

width and height calculated by lower-left packing scheme and upper-right packing scheme must be the same. In the algorithm, after step 3 we get two sets of coordinates for all the blocks. Suppose (x_1, y_1) and (x_2, y_2) are the coordinates after the lower-left packing and upper-right packing of a given block A. The final location of A is computed as a linear combination of the two coordinates after taking into account the "repelling" force from the hot blocks. The coefficient used in the equation is defined to be $f(x) = T_r/T_{total}$. Here, T_{total} denotes the summation of the total temperature of the hot blocks (in our case, this is the set of blocks whose temperature is within 20% of the peak temperature), while T_r denotes the summation of temperature for hot blocks located at the right side of A. Thus the larger the number of hot blocks at the right side of A, the greater is the "repelling" force on A trying to push it leftward. Similarly, $f(y) = T_t/T_{total}$, where T_t denotes the summation of temperature for hot blocks located above A. The adjustment process ensures no overlap between the blocks after they are moved. It should be noted that the movement amount for a block on each layer is determined by the relative positions of the hot blocks on all the layers. This uniform treatment of the blocks in different layers is done to increase the overlap amount between the whitespace of adjacent layers. The fact that a non-overlapping floorplan is always guaranteed after applying this algorithm is addressed by the following lemma.

Lemma: Suppose $f(x)$ is a non-increasing function, and (x_1, y_1) and (x_2, y_2) are the coordinates for the lower-left packing and upper-right packing. It is guaranteed that there will be no overlapping between the blocks after the application of the WSR algorithm.

Proof: For any two block A, B, suppose B is on the right of A, $x_1(A) + width(A) \leq x_1(B), x_2(A) + width(A) \leq x_2(B)$.
$x_A = x_1(A) * f(A) + x_2(A) * (1 - f(A))$;
$x_B = x_1(B) * f(B) + x_2(B) * (1 - f(B))$;
$x_B - x_A = x_1(B) * f(B) + x_2(B) * (1 - f(B)) - x_1(A) * f(A) - x_2(A) * (1 - f(A))$
$\geq (x_1(A) + width(A)) * f(B) + (x_2(A) + width(A)) * (1 - f(B)) - x_1(A) * f(A) - x_2(A) * (1 - f(A))$
$= (x_1(A) - x_2(A)) * (f(B) - f(A)) + width(A)$
$\geq width(A)$;

So, block B is still on the right of A after adjustment step. The proof for the y direction is similar and it is skipped here.

IV. Experiments

In this section, we present results to validate the algorithms we presented earlier. The algorithms were implemented by extending the Parquet software package [8]. The results presented represent an average over ten runs. For temperature estimation and thermal-driven floorplanning, we randomly assign a power density between 10^5 (W/m^2) and 10^7 (W/m^2) to each block [1]. The thermal-driven algorithms use the proposed interpolation model during the algorithms. The final evaluation is done with the accurate model (i.e. the model based on Hotspot). As far as we know, we are the first work to handle thermal issure in fixed outline 3D floorplan problem, therefore only the result of our approach are shown.

Table I compares the fixed outline wirelength driven 3D floorplanning without thermal considerations with the proposed 2-phase 3D floorplanning described in Section IIIA. Four die layers are assumed for all the circuits. The results shows that the proposed algorithm can achieve very high success rates for attaining the fixed outline constraints while obtaining a significant reduction in peak temperature values. Even though, the wirelength is larger compared to the case with no thermal consideration, the wirelength numbers are still significantly smaller than the 2D case[6], thereby demonstrating the benefits of moving to more die layers. The runtime penalty for including temperature consideration in the floorplanning algorithm is about $4X$.

Table II compares the proposed 2-phase 3D floorplanning with a 1-phase algorithm where temperature, wirelength and fixed outline constraints are simultaneously considered. It is easy to see that the 1-phase algorithm is not very successful at attaining both objectives: satisfying the fixed outline constraints and achieving low peak temperature values. The table shows the results of the 1-phase algorithm for different values of η (the coefficient for the temperature in the cost function). With a low value of η, the success rate is good but the peak temperatures are high. On the other hand, with a high value of η (this is the same value used for the temperature in the 2-phase approach), the peak temperatures are lower but the success rates are significantly impacted. The 2-phase algorithm, on the other hand, is very proficient at attaining high success rates while minimizing the peak temperature and the wirelength.

Table III demonstrates the value of considering TSV assignment during the simulated annealing process over performing TSV assignment only at the end. It should be noted that both cases provide an improvement over the original case with thermal-aware floorplanning but no TSV insertion. The reduction in peak temperature is the most when TSV insertion is an integral part of the floorplanning process. The runtime penalty for this scenario is about $3X$. Additional data (details omitted due to lack of space) indicates that the peak temperature increase with 4 die layers can be significantly controlled with thermal-aware floorplanning and thermal TSV insertion. The final floorplan is still hotter than the 2D case and but significantly cooler than the 3D case with 2 die layers and no thermal awareness. Advances in cooling targeted toward 3D ICs should

[6]The detailed data had to be omitted due to lack of space. The wirelength reduction when compared to a fixed-outline 2D floorplan with wirelength minimization is about 44%.

TABLE I
RESULTS FOR NO TEMP VERSUS PROPOSED 2-PHASE ALGORITHM

			No Temp. Consideration			Proposed 2-phase Algorithm		
Circuit	A.R.	WS	Succ. Rate	HPWL	Peak Temp.	Succ. Rate	HPWL	Peak Temp.
n100	1	10	100	181608	453.374	100	201252	286.337
n100	1.5	10	100	179672	482.192	100	201928	287.679
n100	2	10	100	186167	494.907	90	207032	284.848
n100	2	15	100	177220	490.446	100	207838	241.638
n200	1	10	100	327389	447.251	100	368415	311.592
n200	1.5	10	100	326248	438.316	100	376818	292.612
n200	2	15	90	328851	434.189	90	387580	312.756
Avg	-	-	-	1.0X	1.0X	-	1.14X	0.62X

TABLE II
RESULTS FOR PROPOSED 2-PHASE VERSUS 1-PHASE

			Proposed 2-phase Algorithm			1-phase Algorithm (high η)			1-phase Algorithm (low η)		
Circuit	A.R.	WS	Succ. Rate	HPWL	Peak Temp.	Succ. Rate	HPWL	Peak Temp.	Succ. Rate	HPWL	Peak Temp.
n100	1	10	100	201252	286.337	90	188327	275.624	100	182237	404.896
n100	1.5	10	100	201928	287.679	90	188021	275.624	100	184121	409.422
n100	2	10	90	207032	284.848	20	211813	240.86	90	190667	361.774
n100	2	15	100	207838	241.638	100	183542	262.634	100	179870	383.319
n200	1	10	100	368415	311.592	100	338670	309.6	100	327605	402.916
n200	1.5	10	100	376818	292.612	50	371612	280.613	100	325917	382.759
n200	2	15	90	387580	312.756	50	369631	278.866	40	360507	339.552

TABLE III
RESULTS FOR TSV ASSIGNMENT

			No TSV Assignment			TSV Assignment during SA			TSV Assignment only at the end		
Circuit	A.R.	WS	Succ. Rate	HPWL	Peak Temp.	Succ. Rate	HPWL	Peak Temp.	Succ. Rate	HPWL	Peak Temp.
n100	1	10	100	201252	286.337	100	203408	227.84	100	202448	266.073
n100	1.5	10	100	201928	287.679	100	206102	259.817	100	203505	268.282
n100	2	10	90	207032	284.848	100	207750	241.208	90	208537	257.537
n100	2	15	100	207838	241.368	100	211837	196.453	100	209839	222.829
n200	1	10	100	368415	311.592	100	380961	271.824	100	370457	289.217
n200	1.5	10	100	376818	292.612	100	380787	249.314	100	378891	266.282
n200	2	15	90	387580	312.756	90	389621	278.527	90	391264	273.262
Avg.	-	-	-	-	1.0X	-	-	0.85X	-	-	0.92X

be able to further control the temperature increase with increasing die layers. Thus, overall there is a net benefit in increasing the number of die layers, both in terms of wirelength and die footprint.

V. CONCLUSIONS

The paper presented a novel algorithm for thermal-aware 3D floorplanning with fixed-outline constraints. An approximate thermal model that could be used during 3D floorplanning was proposed. A novel algorithm for fixed-outline thermal-aware floorplanning for 3D ICs was also presented. The algorithm also considered thermal TSV assignment as an integral part of the floorplanning algorithm. Results are very promising and show that we are able to significantly reduce the peak temperatures of 3D ICs (average reduction of 46.7% compared to the case with no thermal optimization), while retaining the benefits of wirelength reduction and satisfying the fixed outline constraints.

REFERENCES

[1] J. Cong, J. Wei, and Y. Zhang, "A Thermal-driven Floorplanning Algorithm for 3-D ICs," in *Proc. Intl. Conf. on Computer-Aided Design*, 2000, pp. 306–313.

[2] Z.Li, X. Hong, Q. Zhou, and et al., "Hierarchical 3D Floorplanning Algorithm for Wirelength Optimization," in *IEEE Transcations on Circuits and Systems*, 2006, pp. 2637–2646.

[3] Z.Li and et al., "3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three Dimensional Circuits," in *Proc. Intl. Conf. on Computer-Aided Design*, 2007.

[4] E. Wong and S.K.Lim, "3d Floorplanning with Thermal Vias," in *Proc. Asia and South Pacific Design Automation Conference*, 2007.

[5] Z.Li, X. Hong, Q. Zhou, and et al., "Integrating Dynamic Thermal Via Planning with 3D Floorplanning Algorithm," in *Proc. of International Symposium on Physical Design*, 2006.

[6] A. W. Topol, Jr. D. C. La Tulipe, L. Shi, and et al., "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50, no. 4/5, pp. 491–506, July/September 2006.

[7] X. Li, Y. Ma, X. Hong, and et al., "LP Based White Space Redistribution for Thermal Via Planning and Performance Optimization in 3D ICs", in *Proc. Asia and South Pacific Design Automation Conference*, 2008.

[8] Parquet software package, University of Michigan.

[9] Hotspot software package, University of Virginia.

[10] W. Huang, M. Stan, K. Skandron and et al. "Compact Thermal Modeling for Temperature-Aware Design", in Proc. of Design Automation Conference, 2004.

[11] S. Adya and I. Markov. "Fixed-Outline Floorplanning: Enabling Hierarchical Design", in TVLSI, 2002.

[12] M. Tsai and S. Sinha. "Summer Intern Report", 2007.