

# Thermal-Aware 3D IC Placement Via Transformation

Jason Cong, Guojie Luo, Jie Wei and Yan Zhang

Department of Computer Science  
University of California, Los Angeles  
Los Angeles, CA 90095

Email: { cong, gluo, jwei, zhangyan } @cs.ucla.edu

Abstract - 3D IC technologies can help to improve circuit performance and lower power consumption by reducing wirelength. Also, 3D IC technology can be used to realize heterogeneous system-on-chip design, by integrating different modules together with less interference with each other. In this paper, we propose a novel thermal-aware 3D cell placement approach, named T3Place, based on transforming a 2D placement with good wirelength to a 3D placement, with the objectives of half-perimeter wirelength, through-the-silicon (TS) via number and temperature. T3Place is composed of two steps, transformation from a 2D placement to a 3D placement and the refinement of the resulting 3D placement. We proposed and compared several different transformation techniques, including *local stacking transformation (LST)*, *folding-2*, *folding-4* and *window-based stacking/folding* transformation, and concluded that (i) LST can generate 3D placements with the least wirelength, (ii) the folding-based transformations result in 3D placements with the fewest TS vias, and (iii) the window-based stacking/folding transformations provide good TS via number and wirelength tradeoffs. For example, with four device layers, LST can reduce the wirelength by over 2 $\times$  compared to the initial 2D placement, while window-based stacking/folding can provide over 10 $\times$  variation in terms of the TS via number, thus adaptive to different manufacturing ability for TS via density. Moreover, we proposed a novel *relaxed conflict-net (RCN) graph-based layer assignment* method to further refine the 3D placements. Compared to LST results, thermal-aware RCN graph-based layer assignment algorithm ( $r = 10\%$ ) can further reduce the maximum on-chip temperature by 37%, with only 6% TS via number increase and 8% wirelength increase.

## 1. INTRODUCTION

Three-dimensional (3D) IC technologies can offer the potential to significantly reducing interconnect delays and improving system performance [3]. Furthermore, the shortened wirelength, especially that of the clock net, also lessens the power consumption of the circuit. 3D IC technologies also provide a flexible way to carry out the heterogeneous system-on-chip (SoC) design by integrating disparate technologies, such as memory and logic circuits, radio frequency (RF) and mixed signal components, optoelectronic devices, etc., onto different layers of a 3D IC.

Device layers in a 3D IC are connected using through-the-silicon vias (TS vias). However, TS vias are usually etched or drilled through device layers by special techniques and are costly to fabricate. Large numbers of the TS vias will degrade the yield of the final chip. Also, under the current technologies, TS via pitches are very large compared to the sizes of regular metal wires, usually around 5–10 $\mu\text{m}$ . In 3D IC structures, TS vias are usually placed at the

whitespace between the macro blocks or cells, so the number of TS vias will not only affect the routing resource but also the overall chip or package areas. Therefore, the number of TS vias in the circuit needs to be controlled.

3D fabrication technologies can be roughly divided into three groups according to the level of integration, including (a) packaging-level integration, or 3D multiple-chip module (MCM) [17][21], with TS via pitches larger than 50 $\mu\text{m}$  [19]; (b) block-level integration, usually fabricated by technologies of wafer bonding or silicon epitaxial growth [2], with TS via pitches of about 5 $\mu\text{m}$  [19]; and (c) cell-level integration [16], fabricated by monolithic stacking, with TS via pitches of about 200nm [19]. 3D physical design tools, such as 3D placers, should consider the specific target 3D fabrication technology. For example, under the 3D MCM technology, the vertical interconnect number should be strictly limited.

One of the most critical challenges of 3D IC design is heat dissipation, which has already posed a serious problem even for 2D IC designs [1]. The thermal problem is exacerbated in the 3D cases for mainly two reasons: (1) The vertically stacked multiple layers of active devices causes a rapid increase of power density; (2) The thermal conductivity of the dielectric layers between the device layers is very low compared to silicon and metal. For instance, the thermal conductivity for epoxy,  $k_{\text{epoxy}}$  is 0.05W/mK, while  $k_{\text{silicon}}$  is 150 W/mK and  $k_{\text{copper}}$  is 285 W/mK. Therefore, the thermal issue needs to be considered during every stage of 3D IC designs, including the placement process.

2D placement has been studied extensively and a recent survey is available in [7]. There are several published works targeted on the 3D cell placement problem. Partition-based 3D placement methods [11][10], are proposed to recursively partition the 3D circuit stack to 2D regions and reuse the existing 2D placement engine within each 2D partition. However, neither of the approaches considers the thermal issue during placement. A thermal-driven force-directed 3D placement method [13] was proposed, where the temperature profile is interpreted as thermal forces to guide the cell placement. In their work, a 3D force-directed placement engine is used to place each cell in a true 3D space, with  $z$  position being a real number. Rounding is needed for layer assignment, which may introduce rounding errors.

In this paper, we propose a novel way of generating 3D thermal-aware placement from existing 2D placement results through a two-step procedure: 3D transformation and refinement through layer assignment. For 3D transformation, we propose local stacking transformation, folding-based transformation and the window-based stacking/folding transformation methods. The layer assignment refinement

procedure is based on a relaxed conflict-net (RCN) graph representation. The advantages of our proposed 3D placement transformation include:

- The existing 2D placement core engine can be easily reused. Significant progress has been made on 2D cell placement in recent years. An efficient transformation from a 2D placement to a 3D placement enables us to leverage the existing high-quality 2D placers, such as Kraftwerk [12], MPL6 [4], NTUplace2 [15], which are three best performing placers in the ISPD'06 placement contest.
- A discrete layer assignment algorithm based on graph representation is developed for layer assignment of cells. No rounding for layer assignment is necessary for analytical placement in 3D (as in some previous approach).
- A simple yet effective thermal cost is derived for temperature optimization during layer assignment. No time-consuming thermal profiling is needed during the optimization process.
- Flexible TS via number and wirelength tradeoff is offered by different transformation schemes and the parameter settings in the RCN graph-based layer assignment, which allows our algorithm to be used for different 3D technologies.

The remainder of the paper is organized as follows. In Section 2, we formulate the 3D placement transformation problem. In Section 3, we introduce our 3D placement framework and the resistive thermal model used in T3Place. In Section 4, we propose the 3D placement transformation algorithms, including the local stacking transformation, folding-based transformations, and a window-based stacking/folding transformation. The RCN graph-based layer assignment for placement refinement is proposed in Section 5. The related experimental results are presented in Section 6 to demonstrate the effectiveness of our proposed algorithm. Finally, we conclude the paper and discuss about the future works on thermal-aware 3D cell placement in Section 7.

## 2. PROBLEM FORMULATION

Given a hypergraph  $H = (V, E)$ , device layer number  $K$ , and a 2D placement  $P$ , the thermal-aware 3D placement transformation problem is to assign a position  $(x_i, y_i, z_i)$  to every cell  $v_i$ , so that the total wirelength and the TS via number are minimized, subject to constraints such as non-overlap constraints, temperature constraints, etc.

Under the current 3D fabrication technology, the number of device layers in a 3D IC technology is limited. The length and the width of a 3D chip are usually much larger than the height of the chip. 2D wirelength, including the wirelength at  $x$  and  $y$  directions, still dominates the total wirelength of a 3D design and the total wirelength in the  $z$  direction is negligible. Therefore, in this work, we will use “wirelength” to refer to the 2D wirelength and consider the TS vias for connections in the  $z$  direction as a different optimization objective.

We use the traditional half-perimeter model for wirelength calculation, where the wirelength  $l(e)$  of a net  $e$  is calculated as follows.

$$l(e) = \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j|$$

Because the routing information is unknown during the placement process, TS via number is also estimated through a similar model. The TS via number of a net is calculated as the height of the bounding cube of the cells belong to that net. The TS via number  $v(e)$  of a net  $e$  is calculated as follows.

$$v(e) = \max_{v_i, v_j \in e} |z_i - z_j|$$

## 3. 3D PLACEMENT FRAMEWORK AND RESISTIVE THERMAL MODEL

### 3.1 3D Placement Framework

Figure 1 shows the framework of the proposed 3D placement algorithm. The components with a dashed boundary are the existing tools that we make use of. A 2D wirelength- and/or thermal-driven placer is first used to generate a 2D placement for the target design. The quality of the final 3D placement highly depends on the initial placement. The 2D placement is then transformed into a legalized 3D placement according to the given 3D technology. During the transformation, wirelength, TS via number and temperature are considered. A refinement process through layer reassignment will be carried out after 3D transformation to further reduce the TS via number and bring down the maximum on-chip temperature. Finally, a 2D detailed placer will further refine the placement result for each device layer.

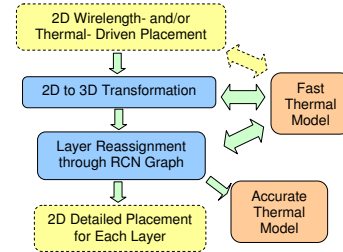


Figure 1 T3Place Framework

### 3.2 Resistive Thermal Model

Since the clock cycle of a modern chip is orders of magnitude smaller than the timing constant for thermal conduction, we need to consider only the steady-state temperature. Also, we only consider the heat generated by transistor switches, so the cells are treated as the only heat sources with constant given power densities. Since a heat sink is usually attached to the substrate, the bottom side of the tile stack is isothermal of constant temperature. Because the chip is usually packaged in thermally insulated materials, the four side walls and top of the chip are treated as adiabatic.

In this work, we use the thermal resistive model proposed in [18]. Compared with the accurate simulation tool, the error of the resistive network model is smaller than 2% [18]. A tile structure is imposed on the circuit stack, as shown in Figure 2(a). Each tile stack contains an array of tiles, one from each device layer, as shown in Figure 2(b). A tile stack is modeled as a resistive chain as shown in Figure 2(c). The tile stacks are connected by lateral resistances. A voltage source is used for the isothermal base of heat sink temperature, and current sources are present in every tile to represent heat sources.

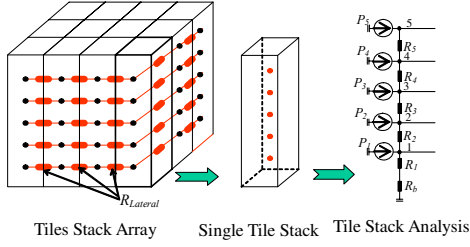


Figure 2 Resistive Thermal Model

## 4. TRANSFORMATIONS FOR 3D PLACEMENT

In this section, we will propose the transformation schemes from a 2D placement to a 3D placement. Initial optimized 2D placement is done on a chip  $K$  times larger than one layer of the 3D chip, where  $K$  is the number of device layers. Given this 2D placement with short wirelength, *local stacking transformation* can achieve even shorter wirelength for the same circuit under 3D IC technology. We will also present two folding-based transformation schemes, *folding-2* and *folding-4*, which can generate 3D placement with very low TS via number. Moreover, TS via number and wirelength tradeoffs can be achieved by the window-based stacking/folding. All these transformation methods can guarantee wirelength reduction over the initial 2D placements.

### 4.1 Local Stacking Transformation

Local stacking transformation (LST) consists of two steps, stacking and legalization, as shown in Figure 3. The stacking step shrinks the chip uniformly but does not shrink cell areas so that cells are stacked in a region  $K$  times smaller and remain the original relative locations. The legalization step minimizes maximum on-chip temperature and TS via number through the position assignment of cells. The result of LST is a legalized 3D placement.

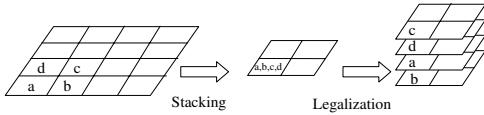


Figure 3 Local Stacking Transformation

#### 4.1.1 Stacking

For  $K$  device layer designs, if the original 2D placement is of size  $S$ , then the 3D cell area of each layer is  $S/K$ . During the stacking step, we first shrink the width and length of the original placement by ratio of  $\sqrt{K}$ , so that the chip region can maintain the original chip aspect ratio. Cell locations  $(x_i, y_i)$  for cell  $i$  are also transformed to new locations  $(x_i', y_i')$ , where  $x_i' = x_i / \sqrt{K}$ , and  $y_i' = y_i / \sqrt{K}$ .

After such a transformation, the initial 2D placement is turned into a 2D placement of size  $S/K$  and has an average cell density of  $K$ , which later will be distributed to  $K$  layers in the legalization step (next subsection). As shown in Figure 3, a group of neighboring cells are pulled together at the same location after the stacking process.

#### 4.1.2 Tetris-Style Legalization

After stacking, we sort all the cells by their x-coordinates in the increasing order. Starting from the leftmost cell, we

determine the location of cells one by one in a way similar to the method used in 2D placement legalization [14]. Each time we consider the left most legal position of all rows at all layers. We pick one position by minimizing the relocation cost  $R$ .

$$R = \alpha \cdot d + \beta \cdot v + \gamma \cdot t \quad (1)$$

where  $d$  is the cell displacement from LST result,  $v$  is the TS via number and  $t$  is the thermal cost. Coefficients  $\alpha, \beta, \gamma$  are predetermined weights.  $d$  is related to the  $(x, y)$  locations of the cells,  $v$  and  $t$  are related to the layers of the cells.

#### 4.1.3 Thermal Optimization

In this work, temperature optimization is considered through the layer assignment of the cells. As mentioned in Section 3.2, under the current 3D IC technologies, the heat sink(s) are usually attached at the bottom (and/or top) side(s) of the 3D IC stack, with other boundaries being adiabatic. So the main dominant heat flow within the 3D IC stack is vertical towards the heat sink. Our study shows that the  $z$  location of a cell will have a larger influence on the final temperature than the  $x/y$  location of the cell [9]. However, the lateral heat flow can be considered if the initial 2D placement is thermal-aware, so that hot cells will be evenly distributed to avoid hot spots.

The full resistive thermal model mentioned in Section 3.2 is used for the final temperature verification. During the inner loops of the optimization process, a much simpler and faster thermal model [9] is used for the temperature optimization to speed up the placement process. Each tile stack is viewed as an independent thermal resistive chain, as shown in Figure 2. The maximum temperature of such a tile stack then can be written as follows.

$$T = \sum_{i=1}^k (R_i \sum_{j=1}^k P_j) + R_b \sum_{i=1}^k P_i = \sum_{i=1}^k P_i (\sum_{j=1}^i R_j + R_b) \quad (2)$$

Besides the fast speed, such a simple close-form equation can also provide a direct guide to thermal-aware cell layer assignment. Equation (2) tells us that the maximum temperature of a tile stack is the weighted sum of the power number at each layer, while the weight of each layer is the sum of the resistances below that layer. Device layers that are closer to the heat sink will have smaller weights.

The thermal cost  $t_{i,j}$  of assigning cell  $j$  to layer  $i$  in Equation (1) can be written as

$$t_{i,j} = P_j (\sum_{k=1}^i R_k + R_b)$$

This thermal cost of layer assignment is also used both in Equation (1) and during placement refinement, which will be presented in Section 5.

## 4.2 Transformation through Folding

LST achieves short wirelength by stacking the neighboring cells together. However, a great number of TS vias will be generated when the cells of local nets are put on top of one another. If the target 3D IC technology allows only limited TS via density, we need to use the transformations that generate fewer TS vias.

Folding-based transformation is to fold the original 2D placement like a piece of paper without cutting off any parts of the placement. The distance between any two cells will not

increase and the total wirelength is *guaranteed* to decrease. TS vias are only introduced to the nets crossing the folding lines (shown as the dashed lines in Figure 4). With an initial 2D placement of minimized wirelength, the number of such long nets should be fairly small, which implies that the connections between the folded regions should be limited, resulting much fewer TS vias (compared to that of the LST transformation, where many dense local connections cross different device layers). Figure 4 (a) shows one way of folding, named folding-2, by folding once at both x and y directions. Figure 4 (b) shows another way of folding, named folding-4, by folding twice at both x and y directions. The folding results are legalized 3D placements, so no legalization step is necessary.

After folding-based transformations, only the lengths of the global nets that go across the folding lines (dotted lines in Figure 4) get reduced. Therefore, folding-based transformations can not achieve as much wirelength reduction as LST. Furthermore, if we want to maintain the original aspect ratio of the chip, folding-based transformations are limited to even numbers of device layers.

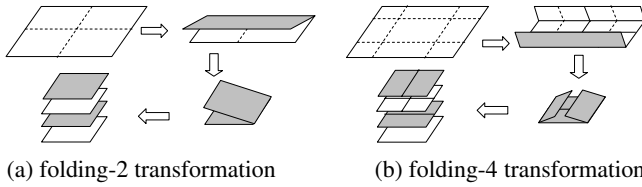


Figure 4 Two Folding-based Transformation Schemes

#### 4.3 Window-based Stacking/Folding

As stated above (and will be shown in Section 6), LST achieves greatest wirelength reduction at the expense of large amount of TS vias, while folding results in much smaller TS via number but longer wirelength and possibly high via density along the folding lines.

An ideal 3D placement should have short wirelength with TS via density satisfying what the vertical interconnection technology can support. Moreover, we prefer even TS via density for routability reason. Therefore, we propose a window-based stacking/folding method for better TS via density control.

In this method, we first divide the 2D placement into  $N \times N$  windows. Then we apply stacking or folding in every window. Each window can use different stacking/folding orders. Figure 5 shows the cases for  $N=2$ . The circuit is divided into  $2 \times 2$  windows (shown in solid lines). Each window is again divided into four squares (shown in dotted lines). The number in each square indicates the layer number of that square after stacking/folding. The four-layer placements of each window are packed to form the final 3D placement.

Wirelength reduction is due to the following reasons: the wirelength of the nets inside the same square is preserved; the wirelength of nets inside the same window is most likely reduced due to the effect of stacking/folding; and wirelength of nets that cross the different windows is reduced. Therefore the overall wirelength quality is improved.

Meanwhile, the TS vias are distributed evenly among different windows and can be reduced by choosing proper layer assignments. TS vias are introduced by the nets that cross the boundaries between neighboring squares with

different layer numbers, and we call such boundary between two neighboring squares a *transition*. Fewer transitions result in fewer TS vias. Intra-window transitions cannot be reduced because we need to distribute intra-window squares to different layers, so we focus on reducing inter-window transitions. Since the sequential layer assignment in Figure 5(a) creates lots of transitions, we use another layer assignment as in Figure 5(b), called symmetric assignment, to reduce the amount of inter-window transitions to zero. So this layer assignment generates the smallest TS via number, while the wirelength is similar.

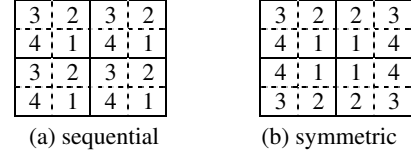


Figure 5  $2 \times 2$  windows with different layer assignments

The wirelength vs. TS via number tradeoffs can be controlled by the number of windows.

## 5. REFINEMENT: RCN GRAPH-BASED LAYER REASSIGNMENT

During the 3D transformations proposed in the previous Section, layer assignment of cells is based on simple heuristics. To further reduce the TS via number and the temperature, we propose a novel layer assignment algorithm to re-assign the cell layers.

### 5.1 Conflict-Net Graph

We extended a metal wire layer assignment algorithm proposed in [6] for cell layer assignment in 3D placement. For a given legalized 3D placement, a conflict-net (CN) graph is created, as shown in Figure 6, where both the cells and the vias are nodes. One via node is assigned for each net. There are two types of edges, net edges and conflicting edges. Within each net, all cells are connected to the via node by net edges in star mode. A conflict edge is created between cells that overlap with each other if they are placed in the same layer.

We want to find a layer assignment of each cell node in the graph so that the total cost, including edge costs and node costs are minimized. Cost  $\theta$  is assigned for all net edges. If two cells connecting by a conflicting edge are assigned to the same layer, the cost of the conflicting edge is set to *infinity*, otherwise, the cost is set to  $\theta$ . The cost of a via node is the height of that via, which represents the total TS via number in that net. The heights of the vias are determined by the layers of the cells connecting with them. The cost of a cell node  $v_j$  is the thermal cost  $t_{ij}$  of assigning  $v_j$  to layer  $i$ . The cost of a path is the sum of the edge costs and the node costs along that path.

The resulting graph is a directional acyclic graph. A dynamic programming optimization method can be used to find the optimal solution for each induced sub-tree of the graph in linear time. An algorithm that constructs a sequence of maximal induced sub-tree from the CN graph is then used to cover a large portion of the original graph. It turns out that average node number of the induced sub-trees can be as many as 40%-50% of the total nodes in the graph. After the iterative optimization of the sub-trees, we can achieve a globally

Table 1 Benchmark characteristics and Wirelength Comparison of T3Place and 2D mPL5 [5]

circuit	cell #	net #	2D mPL5	LST (r=10%)		LST (r=20%)		Folding-2		Folding-4		LST (8x8 win)	
				WL	via #	WL	via #	WL	via #	WL	via #	WL	via #
ibm01	12282	11507	5.19E+06	2.52E+06	18519	2.68E+06	14102	4.61E+06	1671	4.55E+06	2476	3.53E+06	6688
ibm03	22207	21621	1.37E+07	6.62E+06	30434	7.29E+06	21406	1.14E+07	4125	1.11E+07	5909	8.36E+06	12318
ibm04	26633	26163	1.67E+07	8.45E+06	37414	9.20E+06	26871	1.55E+07	2940	1.43E+07	6388	1.10E+07	15315
ibm06	32185	33354	2.20E+07	1.10E+07	50139	1.52E+07	32939	2.02E+07	4116	1.83E+07	9077	1.44E+07	19315
ibm07	45135	44394	3.73E+07	1.83E+07	65093	2.07E+07	44715	3.18E+07	5932	3.10E+07	8755	2.37E+07	25021
ibm08	50977	47944	3.94E+07	1.98E+07	70317	2.13E+07	49844	3.48E+07	5801	3.28E+07	10181	2.56E+07	25205
ibm09	51746	50393	3.46E+07	1.72E+07	72787	1.95E+07	50755	3.19E+07	4540	2.93E+07	8257	2.34E+07	23836
ibm13	81508	83806	6.58E+07	3.24E+07	121135	3.60E+07	85103	6.03E+07	7696	5.85E+07	13071	4.50E+07	42568
ibm15	158244	161196	1.65E+08	8.26E+07	246509	9.11E+07	176018	1.45E+08	15128	1.38E+08	23662	1.14E+08	72956
ibm18	210323	200565	2.43E+08	1.26E+08	297771	1.34E+08	208564	2.24E+08	12077	2.08E+08	28287	1.74E+08	83380
Avg.			1.00	0.50	1.00	0.56	0.71	0.89	0.08	0.85	0.14	0.67	0.36

optimized solution. Please refer to [6] for the detailed algorithm to solve the layer assignment problem with CN graph.

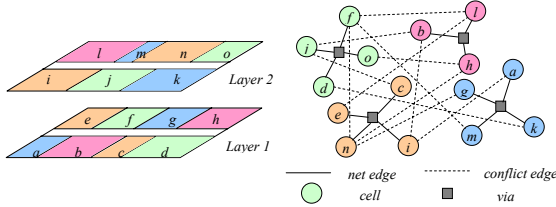


Figure 6 Relaxed Conflict-Net Graph

5.2 Relaxed Non-Overlap Constraint

To further reduce the TS via number and the maximum on-chip temperature, the non-overlap constraints can be relaxed so that a small amount of overlap  $r$  is allowed in exchange for more freedom in layer reassignment of the cells.

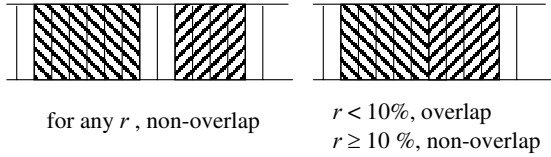


Figure 7 Relaxation of Non-overlap Constraint

The relaxed non-overlap is defined as follows.

$$overlap(i, j) = \begin{cases} false, & \text{if } \frac{o(i, j)}{s(i) + s(j)} \leq r, \\ true, & \text{if } \frac{o(i, j)}{s(i) + s(j)} > r \end{cases}$$

where  $o(i, j)$  is the area of the overlapped region of cell  $v_i$  and cell  $v_j$ ,  $s(i)$  is the area of cell  $i$ .  $r$  is a positive real number between 0 and 0.5. This is illustrated in Figure 7

However, with the relaxed non-overlap constraint, the layer assignment result is no longer a legalized 3D placement and another round of legalization is needed to eliminate the overlap.

6. EXPERIMENTAL RESULTS

We implemented T3Place, which features local stacking 3D transformation and RCN graph-based layer reassignment. We also implemented two folding-based 3D transformation methods and window-based stacking/folding methods. In our experiments, we use MPL5 [5] and its detailed placer [8] to generate the initial legalized wirelength-driven 2D global

placement results. After 3D transformation and refinement, we will apply the 2D detailed placer on each of the device layers to further reduce wirelength. The wirelength and TS via numbers are computed based on the models discussed in Section 2 without performing routing.

Our experiments are performed on 2.4GHz Pentium IV machines with Red Hat enterprise 3.0. We test T3Place using the popular 2D standard cell placement benchmark set, IBM-PLACE from [20]. A four device layer 3D IC structure is assumed for all test cases.

In Table 1, we first compared the results of the wirelength-driven T3Place with the 2D placement results, as shown in. Compared with 2D mPL5, T3Place ( $r = 10\%$ ,  $20\%$ ) can reduce the wirelength by about 2x with four device layers. Moreover, we compared LST transformation with folding-2 and folding-4 and the results are also shown. For folding-based transformations, no placement refinement is done because the TS via number is already very small. Compared to LST, folding-2 can reduce the TS via number by over 12.5x but with only 11% wirelength reduction over the 2D placement. With more folding lines, folding-4 can achieve 15% wirelength reduction over the 2D placement with 7x TS via reduction compared to LST. At last, results of 8x8 window-based stacking demonstrate that our method is adaptive to different manufacturing ability for TS-via density.

We also compared the thermal-aware T3Place with T3Place with no temperature optimization. The results are shown in Table 2. In both schemes, the LST overlap allowance  $r$  is set to 10%. We report the difference of the maximum on-chip temperature and the heat sink temperature. Compared to the wirelength-driven only T3Placer, the thermal-aware T3Placer can reduce the maximum on-chip temperature by 37% on average with 6% more TS vias and 8% longer wirelength.

Table 2 Thermal-Aware T3Place Results

circuit	LST, r = 10%		LST, r = 10%, w/ temp optimization	
	Temp. (°C)	WL	via #	Temp. (°C)
ibm01	276.5	2.81E+06	19020	159.8
ibm03	196.7	7.13E+06	31780	121.6
ibm04	159.6	9.11E+06	40219	96.0
ibm06	160.4	1.23E+07	50576	103.5
ibm07	107.5	2.01E+07	69111	66.4
ibm08	97.7	2.05E+07	75397	63.2
ibm09	96.1	1.94E+07	78102	60.6
ibm13	249.3	3.47E+07	127520	156.2
ibm15	136.5	8.58E+07	260681	90.1
ibm18	89.4	1.31E+08	332012	58.7
Avg.	1.0	0.54	1.06	0.63

Furthermore, we compared the wirelength-driven T3Place with a recently published 3D cell placement algorithm [13], as shown in Table 3. To compare with their results, we also scaled the chip to  $2cm \times 2cm$  and set the height of the TS vias as  $20 \mu m$  as suggested by the authors of [13]. Compared to that reported by [13], our T3Place shows significant wirelength reduction (over  $5\times$ ) and we have been in contact with the authors of [13] to share the results and experimental setting. Because we use different thermal model for 3D IC, the temperature values are not comparable.

Table 3 Comparisons with Existing 3D Placement [13]

circuits	Scaled WL	via #	Total WL under metric of [13]	Total WL in [13]
ibm01	13.3	18519	13.7	63.8
ibm03	29.6	30434	30.2	115.9
ibm04	32.5	37414	33.2	144.5
ibm06	43.3	50139	44.3	183.2
ibm07	53.1	65093	54.4	277.7
ibm08	54.2	70317	55.6	278.9
ibm09	46.3	72787	47.7	252.5

## 7. CONCLUSIONS

In this paper, we propose a novel 3D placement approach through transformation from 2D placement results. We compare different transformation approaches, including LST, folding-2, folding-4 and window-based stacking/folding. We also propose a novel 3D placement refinement through RCN graph-based layer assignment, which can reduce TS via number and temperature significantly.

For the future work at this direction, the heat dissipation effect of the TS vias should be considered during placement. Also, we should considering more accurate wirelength and TS via estimation based on global routing rather than the simple models stated in Section 2. Although the wirelength and TS via estimation models in Section 2 are very efficient to guide the 3D placement engine, they have the problem that both estimations may not able to be realized simultaneously for some complex multi-terminal nets, as achieving the minimum wirelength routing may require to use multiple TS vias. Finally, whitespace allocation and congestion should also be considered in placement optimization to generate routable 3D placement results.

## ACKNOWLEDGEMENTS

This research is partially supported by Gigascale Silicon Research Center, and the National Science Foundation under CCF 0430077. The authors wish to thank Dr. Marek Turowski and Mr. Patrick Wilkerson from CFD Research Corporation (CFDRC) for providing the compact resistive network thermal model, Prof. Sapatnekar and Mr. Brent Goplen for providing their 3D placement wirelength data for comparison.

## REFERENCES

[1] K. Banerjee, "Thermal Effects in Deep Submicron VLSI Interconnects," *IEEE Int. Symposium on Quality Electronic Design*, 2000.  
 [2] K. Banerjee, S. J. Souri, P. Kapur, K. C. Saraswat, "3D ICs: A Novel Chip Design for Improving Deep Submicron Interconnect

Performance and Systems-on-chip Integration," *Proc. IEEE, Special Issue on Interconnects*, pp.602-633, May 2001.  
 [3] B. Black, D. W. Nelson, C. Webb, and N. Samra, "3D Processing Technology and its Impact on IA32 Microprocessors," *Proc. of ICCD*, pp.316-318, 2004.  
 [4] T. Chan, J. Cong, J. Shinnerl, K. Sze and M. Xie, "mPL6: enhanced multilevel mixed-size placement," *Proc. Int. Symposium on Physical Design*, pp. 212-214, 2006.  
 [5] T. Chan, J. Cong, and K. Sze, "Multilevel Generalized Force-Directed Method for Circuit Placement," *Proc. Int. Symposium on Physical Design*, San Francisco, CA, April, 2005.  
 [6] C.-C. Chang and J. Cong, "An Efficient Approach to Multilayer Layer Assignment with An Application to Via Minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.18, no. 5, p.608-620, May 1999.  
 [7] J. Cong, T. Kong, J. Shinnerl, M. Xie, and X. Yuan, "Large Scale Circuit Placement," *ACM Transaction on Design Automation of Electronic Systems*, vol. 10, no. 2, pp. 389-430, April 2005.  
 [8] J. Cong and M. Xie, "A robust detailed placement for mixed-size IC designs," *Proc. Asia South Pacific Design Automation Conference*, pp. 188-194, 2006.  
 [9] J. Cong and Y. Zhang, "Thermal Via Planning for 3-D ICs," *Proc. Intl. Conf. on Computer-Aided Design*, pp. 745-752, 2005.  
 [10] S. Das, A. Chandrakasan, and R. Reif, "Design Tools for 3-D Integrated Circuits," *Proc. Asia South Pacific Design Automation Conference*, 2003.  
 [11] Y. Deng, W. Maly, "Interconnect Characteristics of 2.5-D System Integration Scheme," *Proc. Int. Symposium on Physical Design*, pp. 171-175, 2001.  
 [12] H. Eisenmann and F.M. Johannes, "Generic Global Placement and Floorplanning," *Proc. Design Automation Conference*, pp. 269-274, 1998.  
 [13] B. Goplen and S. S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs using a Force Directed Approach," *Proc. Int. Conf. on Computer-Aided Design*, pp. 86 - 89, 2003.  
 [14] D. Hill, "Method and System for High Speed Detailed Placement of Cells within an Integrated Circuit Design," *US Patent 6370673*, 2001.  
 [15] Z.-W. Jiang, T.-C. Chen, T.-C. Hsu, H.-C. Chen and Y.-W. Chang, "NTUPlace2: A Hybrid Placer Using Partitioning and Analytical Techniques," *Proc. Int. Symposium on Physical Design*, pp. 215-217, 2006.  
 [16] T. H. Lee, "A Vertical Leap for Microchips," *Scientific American*, 2002  
 [17] Y. K. Tsui, S. W. R. Lee, J. S. Wu, J. K. Kim and M. M. F. Yuen, "Three-Dimensional Packaging for Multi-Chip Module with Through-the-Silicon Via Hole," *Electronics Packaging Technology*, 5th Conference, pp 1-7, 2003.  
 [18] P. Wilkerson, A. Raman, and M. Turowski, "Fast, Automated Thermal Simulation of Three-Dimensional Integrated Circuits," *ITherm 2004*, Las Vegas, Nevada, 1-4 June 2004.  
 [19] S. Wong, "3-D Integrated Circuit with Unlimited Upward Extendibility," *talk at DARPA meeting*, Feb, 2005.  
 [20] <http://er.cs.ucla.edu/benchmarks/ibm-place/>  
 [21] [http://www.irvine-sensors.com/r\\_and\\_d.html](http://www.irvine-sensors.com/r_and_d.html)