LP Based White Space Redistribution for Thermal Via Planning and Performance Optimization in 3D ICs*

Xin Li* Yuchun Ma* Xianlong Hong* Sheqin Dong* Jason Cong**
*Department of Computer Science & Technology, Tsinghua University, Beijing 100084, P.R.China
Email: {myc, hxl-dcs}@mail.tsinghua.edu.cn
**Department of Computer Science & Technology, UCLA, USA

Abstract: Thermal issue is a critical challenge in 3D IC circuit design. Incorporating thermal vias into 3D IC is a promising way to mitigate thermal issues by lowering down the thermal resistances between device layers. However, it is usually difficult to get enough space at target regions to insert thermal vias. In this paper, we propose a novel analytical algorithm to re-allocate white space for 3D ICs to facilitate via insertion. Experimental results show that after reallocating whitespaces, thermal vias and total wirelength could be reduced by 14% and by 2%, respectively. It also shows that whitespace distribution with via planning alone will degrade performance by 9% while performance-aware via planning method can reduce thermal via number by 60% and the performance is kept nearly unchanged.

I. INTRODUCTION

Three-dimensional (3D) integration has recently drawn much attention due to its potential for reducing the interconnect delay. However, there are some significant challenges along with its adoption and further development. One extremely important issue in 3D IC design is the thermal problem coming from the higher power density and low thermal conductivity.

During the past a few years, several works on thermal optimization have been proposed including thermal-driven floorplanning[1,2], placement[3,4] and routing[5]. Unfortunately, even with complicated thermal-aware approach to improve heat dissipation, the maximum on-chip temperature is still too high for the circuit to operate properly[1]. Introducing thermal vias into the circuit(Fig.1), in fact, is an efficient way to reduce the chip temperature to a satisfactory level[7]. But it is not free to use thermal vias in 3D designs. On one hand, it is costly to fabricate thermal vias and the common thermal via pitch is very large compared that of regular mental wires. On the other hand, thermal vias are usually inserted into white space between blocks, which may block the routing and cause serious congestion problem. Consequently, the number of thermal vias should be reduced as much as possible while temperature constraints must be satisfied as well.

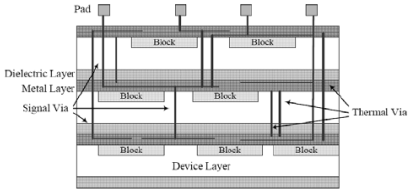
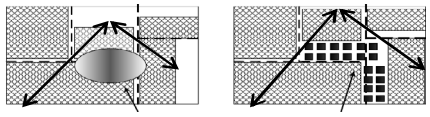


Figure 1 thermal via in 3D IC stack

To optimize the distribution of thermal vias, some previous works[6, 8] formulate and solve the thermal via problem as a post-floorplan procedure to improve heat resource distribution while [9] incorporate thermal via planning into the thermal-driven floorplanning.

*This work is supported by the NSFC 60606007 and Tsinghua Basic Research Fund JC20070021

Nevertheless, the thermal via distribution is determined by the thermal distribution. Generally speaking, the hotter the region is, the more vias are needed. In fact, hot areas are usually occupied by macro blocks or cells and thermal distribution can not easily be improved since thermal vias can not be inserted directly into these regions as shown in Figure 2(a). As a result, the maximal temperature could not be reduced to desired threshold unless the initial floorplan is modified to facilitate via insertion, which would cause degradation on total wirelength and overall packing area. As shown in Figure 2(b), after modifying positions of the blocks, it could increase total wirelength. In addition, it may even worsen performance by increasing delays along critical paths.



(a) no white space (b) white space redistribution

Figure 2 via insertion and wirelength degradation

Facing this case, [6] and [8] just insert thermal vias into nearby white spaces or remote regions, which may need much more thermal vias or even may fail to bring down maximal temperature to desired threshold. In this paper, we propose a novel thermal-aware algorithm to redistribute white space for 3D ICs to favor the thermal via insertion while the performance of the design is simultaneously under control.

II. PROBLEM STATEMENT

Thermal-aware whitespace redistribution in 3D IC can be described as: Given a multi-layer packing with a set of n blocks $M=\{M_1, M_2, \cdots, M_n\}$ in K layers, where w_i and h_i specify the dimensions of block M_i, respectively, a set of nets $N=\{N_1, N_2, \cdots, N_m\}$ where N_i, $i=1,2,\cdots, m$ describes the connections between blocks, we need to generate a new layout with the same outline of the original packing and the original relative relations between blocks are remained unchanged. But the whitespace between blocks are redistributed so that: 1) the required temperature can be reached with fewer thermal vias; 2) total wirelength is not enlarged evidently or the performance estimation of the design is not degraded seriously.

To deal with the trade-off between multiple objectives, we first estimate thermal via requirement in each tile by a thermal profile model. Then we formulate thermal via requirement and other optimizations into LP models so that multiple objectives and constraints can be handled simultaneously.

III. OVERVIEW OF THERMAL RESISTANCE MODEL

For temperature profiling, we use the same thermal resistive model as [6]. The 3D circuit stack is divided by a two-dimensional array of tile stacks, as shown in Figure 3(a). Each tile stack is

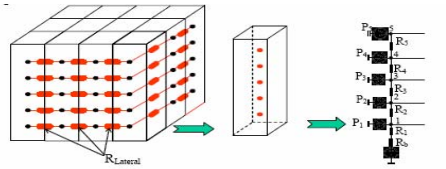
978-1-4244-1922-7/08/$25.00 ©2008 IEEE

composed of several vertically-stacked tiles, as shown in Figure 3(b). These tile stacks are connected by lateral thermal resistances, $R_{lateral}$. Within each tile stack, a thermal resistor R_i is modeled for the i-th device layer, while thermal resistance of the bottom layer and silicon substrate is modeled as R_b as shown in Figure 3(c).

A tile stack is modeled as a resistive network. The isothermal bases of room temperature are modeled as a voltage source. A current source is present at every node in the network to represent the heat sources. The tile stacks are connected by lateral resistances. One can spatially discretize the system and solve the following equation to determine the steady-state thermal profile as a function of power profile.

$$T = PA^{-1} \tag{1}$$

where A is an $N \times N$ sparse thermal conductivity matrix. T and $P(t)$ are $N \times 1$ temperature and power vectors.



(a) Tiles Stack Array (b) Single Tile Stack (c) Tile Stack Analysis

Figure 3 Resistive thermal model for 3D ICs

Cong, et al[6] formulated the thermal via planning as a nonlinear programming (NLP) problem and solved it by solving a sequence of simplified via planning sub-problems in alternating directions. In this paper, we will use the same via planning model to estimate the requirement of via number in each tile and fulfill thermal via insertion.

IV. THERMAL VIA BUDGET FOR WHITESPACE ALLOCATTION

In room-based floorplanning representation such as CBL[14], the whole layout can be partitioned into rooms and blocks are packed within rooms as shown in Figure 2. To insert enough thermal vias to target positions, the white spaces resource should be allocated as needed. In our approach, we transform the required thermal vias of the tiles to *via requirements* of rooms, which will facilitate the LP modeling. We firstly calculate the tile coverage by the room partitions, then we attach via requirement to each room.

Given an n-block set, it divides the chip into at least n rooms and assigns no more than one block to each room. Since the tile structure is constructed to compute the thermal profile and via distribution, we first figure out *ideal via number* for each tile to satisfy temperature threshold. Then we assign *via requirement* to each room according to the tiles it covers. Assume (x_i,y_i) are the coordinates for lower-left corner of block i. As Figure 4 shows, tiles are denoted by shadow rectangles, $area\_budget_{ij}$ is defined as the overlapping area of room i and tile j, which can calculated as:

$$area\_budget_{ij} = \begin{cases} \Delta x * \Delta y & if\ room\ \ i\ \text{covers}\ tile\ j \\ 0 & if\ not\ \text{cover} \end{cases} \tag{2}$$

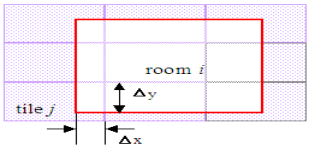


Figure 4 room i covers tile j

$$where: \quad \Delta x = \min(x_j + w_j, x_i + w_i) - \max(x_j, x_i)$$
$$\Delta y = \min(y_j + h_j, y_i + h_i) - \max(y_j, y_i) \tag{3}$$

Assume VN_j is *ideal via number* for tile j, *via_area* denotes the area of a single thermal via. We distribute the *ideal via number* in tile j to those rooms that cover tile j. The exact percentage of via number allocated to room i is proportional to its $area\_budget_{ij}$. The distribution procedure is a two-stage approach. The first stage is to determine the number of vias allocated to the room and the second one will find appropriate boundary of the room to receive these via number. *Via requirement* req_{ij} divided out to room i from tile j is calculated by:

$$req_{ij} = \frac{VN_j * via\_area * area\_budget_{ij}}{\sum_i area\_budget_{ij}} \tag{4}$$

which suggests that the more a room covers the tile, the more *via requirement* from this tile will be allocated to this room.

After *via requirement* is distributed to the room (transform *ideal via number* to *via requirement*), then we must calculate the required area for thermal via insertion in each room. We attach the *area requirement* in each room to the corresponding block. Assume $left\_req_{ij}$, $right\_req_{ij}$, top_req_{ij} and $bottom\_req_{ij}$ are individual *via requirements* distributed to the left, the right, the top and the bottom boundary of room i from tile j, respectively. The actual distribution percentage is proportional to the part of the boundary cut by the tile. As can be seen from Figure 4, the left bottom corner of room i is inside the tile j, thus the *via requirements* are distributed to $left\_req_{ij}$ and $bottom\_req_{ij}$:

$$left\_req_{ij} = req_{ij} * \Delta y / (\Delta x + \Delta y)$$
$$bottom\_req_{ij} = req_{ij} * \Delta x / (\Delta x + \Delta y) \tag{5}$$

If the entire tile j is covered by the room i, we can distribute *via requirement* evenly to the four boundaries. Above requirement values in each tile are calculated sequentially. Assume LA_i, RA_i, TA_i and BA_i are total via requirement attached to the left, the right, the top and the bottom boundary of block i *of* room i, respectively, as can be seen in Figure 5.

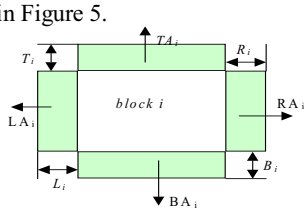


Figure 5 via requirement attached to block i

Actually, these variables can be achieved by:

$$LA_i = \sum_{for\ all\ tile\ j} left\_req_{ij},\ BA_i = \sum_{for\ all\ tile\ j} bottom\_req_{ij}$$
$$TA_i = \sum_{for\ all\ tile\ j} top\_req_{ij},\ RA_i = \sum_{for\ all\ tile\ j} right\_req_{ij} \tag{6}$$

After *via requirements* for each block are obtained, on the assumption that these required area are rectangle-shaped we then take them as our optimization objective to achieve as much white space as possible for via insertion.

V. LP BASED WHITE SPACE REDISTRIBUTION

A. LP constraints for the topological relations

Given an initial floorplan, it is easy to represent the topological relations in linear constraints preventing overlapping of any pair of rectangular modules i and j. In white space redistribution, blocks would deviate from the original positions. Let (x_i,y_i), (x_j,y_j) denote the variable positions of the lower left corners of module i and j. From the existing floorplan using 3D representation such as CBA[1], we can capture the corresponding relative positions of modules, which keep unchanged in the optimization process. To prevent overlapping between original modules i and j, one of the following linear inequalities actually holds:

$$\begin{aligned} x_i + w_i &\le x_j && if\ i\ left\ to\ j \\ x_j + w_j &\le x_i && if\ i\ is\ right\ to\ j \\ y_i + h_i &\le y_j && if\ i\ is\ below\ \ j \\ y_j + h_j &\le y_i && if\ i\ is\ above\ \ j \end{aligned} \tag{7}$$

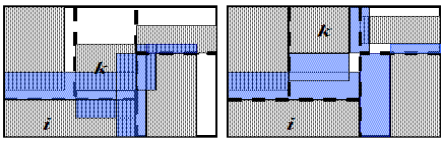
Meanwhile, assume W and H are width and height of the original packing, respectively, to keep the initial packing area, additional inequalities for any module i are needed as follows:

$$x_i \ge 0, \quad y_i \ge 0, \quad x_i + w_i \le W, \quad y_i + h_i \le H \tag{8}$$

B. Optimization with thermal via planning

After we obtain *via requirement* of each block, we can establish the objective for white space optimization by distributing the white space to meet *via requirement* as much as possible. As shown in

section IV, *via requirements* are attached to the boundaries of blocks. White spaces between blocks can also be partitioned into rectangles around blocks. As Figure 6 shows, the shadow rectangles represent *via requirement* of each block and we could redistribute the rectangular white spaces to cover these shadow regions. After the optimization, most of *via requirement* are satisfied, except TA_j of block i and RA_k of block k due to the fixed-outline constraints.



(a) via requirement (b) results of optimization

Figure 6 optimization with via planning

Let variables la_i, ra_i, ta_i and ba_i represent the actual rectangular white spaces finally allocated to the left, the right, the top and the bottom boundaries of block i, To meet the thermal via requirement as much as possible, the objective then can be written as:

$$\min \sum_{i \in M} \beta_i \{ (LA_i - la_i) + (RA_i - ra_i) + (TA_i - ta_i) + (BA_i - ba_i) \} \quad (9)$$

where β_i is the weight factor for block i. The hotter blocks will be assigned with higher weight values.

Intuitively, above objective is to optimize area-difference which is generally quadratic. But if we consider the following equations:

$$\begin{aligned} LA_i &= L_i * h_i, \quad RA_i = R_i * h_i, \\ TA_i &= T_i * w_i, \quad BA_i = B_i * w_i \\ la_i &= l_i * h_i, \quad ra_i = r_i * h_i \\ ta_i &= t_i * w_i, \quad ba_i = b_i * w_i \end{aligned} \quad (10)$$

where L_i, R_i, T_i and B_i are heights(widths) of LA_i, RA_i, TA_i and BA_i, respectively, as Figure 5 shows, while l_i, r_i, t_i and b_i are corresponding actual heights(widths) satisfied finally. Then the initial objective can be rewritten as:

$$\min \sum_{i \in M} \beta_i \{ h_i(L_i - l_i) + h_i(R_i - r_i) + w_i(T_i - t_i) + w_i(B_i - b_i) \} \quad (11)$$

It is a linear objective, since l_i, r_i, t_i and b_i are variables while all the others are known constants.

Take the white space around blocks into consideration, the relative relation constraints in equations (7) should be revised to:

$$\begin{aligned} x_i + w_i + r_i &\le x_j - l_j \quad \text{if } i \text{ left to } j \\ x_j + w_j + r_j &\le x_i - l_i \quad \text{if } i \text{ is right to } j \\ y_i + h_i + t_i &\le y_j - b_j \quad \text{if } i \text{ is below } j \\ y_j + h_j + t_j &\le y_i - b_i \quad \text{if } i \text{ is above } j \end{aligned} \quad (12)$$

And the fixed-outline constraints should also be re-written as:

$$x_i \ge l_i, \quad y_i \ge b_i, \quad x_i + w_i + r_i \le W, \quad y_i + h_i + t_i \le H \quad (13)$$

C. Optimization with wirelength(HPWL)

Here we model HPWL optimization with LP formulation as follows. Assume the pins are at the center of the corresponding blocks and each net j has four variables x^j_{max}, x^j_{min}, y^j_{max}, and y^j_{min} representing its four boundary edges of the bounding box, (x^i_{pin}, y^i_{pin}) represents the pin location of block i. Since pins are in the bounding box, following LP constraints can be achieved:

$$\begin{aligned} x^i_{pin} &\ge x^j_{\min}, \quad x^i_{pin} \le x^j_{\max} \\ y^i_{pin} &\ge y^j_{\min}, \quad y^i_{pin} \le y^j_{\max} \\ x^i_{pin} &= x_i + w_i / 2, \; y^i_{pin} = y_i + h_i / 2 \end{aligned} \quad (14)$$

Hence, the total wirelength HPWL could be minimized by:

$$\min \sum_{j \in N} x^j_{\max} - x^j_{\max} + y^j_{\max} - y^j_{\max} \quad (15)$$

Where N is set of nets of the floorplan.

D. LP formulation of performance in micro achitecture

In micro-architecture design, there are many different sub-systems which separate paths into groups, such as bypass wires between the intALUs. Unlike the wirelength computation, where the wirelength of each net is summed up, in timing analysis, only the delay of the critical path, which is the longest path in a group, will be counted. The performance of the architecture can be evaluated by the weighted sum of number of cycles along the paths. Though the whitespace redistribution does not affect the packing area, blocks' movement in the rooms may change the distance between blocks so the delays on nets will be modified accordingly. Assume K is the delay that signal travels by unit wirelength and Φ is the clock period. (x^i_{pin}, y^i_{pin}) represents the location of pin, lat_i is intrinsic delay of block i. Assume T_p, $cycle_g$ are the delay on path p, number of cycles on the critical path in group g. So that we have:

$$T_p = \sum_{(i,j) \in p} (|x^i_{pin} - x^j_{pin}| + |y^i_{pin} - y^j_{pin}|) * K + \sum_{k \in p} lat_k, \quad cycle_g = \left\lceil \max_{p \in g}(T) / \Phi \right\rceil \quad (16)$$

where (i,j) is a edge in path p. In [17,18], performance estimation is formulated as a linear function:

$$\sum_g c_g * cycle_g \quad (17)$$

Where c_g is the degradation factor for g. Formula (16) implies that for each path p in group g, following inequality is active:

$$\begin{aligned} & T_p \le \max_{p \in g}(T) \le \Phi * cycle_g \\ \Rightarrow & \sum_{(i,j) \in p} K * (|x^i_{pin} - x^j_{pin}| + |y^i_{pin} - y^j_{pin}|) + \sum_{k \in p} lat_k \le \Phi * cycle_g \end{aligned} \quad (18)$$

To model absolute term $|\bullet|$ in LP, we introduce two extra variables vx_{ij} and vy_{ij}, then add four equivalent inequalities:

$$\begin{aligned} x^i_{pin} - x^j_{pin} &\le vx_{ij}, \; -x^i_{pin} + x^j_{pin} \le vx_{ij} \\ y^i_{pin} - y^j_{pin} &\le vy_{ij}, \; -y^i_{pin} + y^j_{pin} \le vy_{ij} \end{aligned} \quad (19)$$

then performance-driven LP objective can be written as:

$$\begin{aligned} \min \quad & \sum_g c_g * cycle_g \\ s.t. \quad & \sum_{(i,j) \in p} K * (vx_{ij} + vy_{ij}) + \sum_{k \in p} lat_k \le \Phi * cycle_g \end{aligned} \quad (20)$$

E. Simultanous optimization of multiple objectives

Together with previous LPs, now we can establish our thermal-aware and performance-driven white space redistribution approach. Multiple objectives, i.e., thermal via planning(TV), total wirelength(WL) and performance(P), can be considered at the same time as follows:

$$\min \quad \alpha * TV + \beta * WL + \lambda * P \quad (21)$$

VI. EXPERIMENTAL RESULTS

We have implemented our white space redistribution approach in C++ programming language, and all experiments are performed on a 1.6GHz PC/Intel. The LP solver we used is [10]. The tile array in each layer is set as 30×30 and the initial temperature profiles are generated by a fast resistance simulator from [6]. All initial floorplans are generated from a SA-based thermal-driven floorplanning algorithm[1,13]. We apply the thermal via planning algorithm m-ADVP from [6] to calculate *ideal via number* in each tile and to insert the thermal vias. M-ADVP has no explicit white space reallocation for thermal via insertion, which just insert thermal vias into nearby white spaces or remote regions if no enough white spaces exist in the target regions. The required temperature threshold is set to be 77^oC.

A. Whitespace redistribution for thermal via planning

Firstly we optimize the white space and total wirelength simultaneously on some GSRC benchmarks. The power dissipations of each block are assigned to the same value with [6] (ranging from $10^5 W/m^2$ to $10^7 W/m^2$). For each benchmark, we generated two different initial floorplans with different max temperatures. Table 1 shows the experimental results of our approach. The cpu column is runtime of the whitespace redistribution procedure. As can be seen, for the same temperature constraints, our algorithm can reduce both thermal vias by 14% and total wirelength by 2% compared to the original packing results. As an illustration, figure 7 provides the floorplans before and after white space redistribution for thermal via insertion.

In addition, Figure 8 displays the thermal-via distribution before and after optimization for n200. On the one hand, thermal vias after optimization are distributed more evenly than original floorplan, and some areas where thermal vias cannot be inserted previously now can contain many vias. On the other hand, without the white space re-allocation, due to the limited white space resources around the hotspots, large amount of thermal vias might be gathered to the places which have much white space. After the white space redistribution, previous congested regions are relieved to have fewer thermal vias.

B. Performance driven whitespace redistribution

Here we optimize white space for thermal via planning and design performance simultaneously on Alpha 21264 microprocessor. Table 2 is the experimental result of our approach. Chip performance is estimated by *bips(billion instructions per seconds)*. It shows that whitespace distribution with only via planning will degrade performance by about 9% while performance-aware method can reduce thermal via number by 60% but the performance is remained nearly unchanged. The runtime of all the algorithms are less than 1 second since the number of blocks in this case, which has about 20 blocks, is much less than that in GSRC benchmarks. Note for the designs with high frequency such as the case with 3G frequency in Table II, the original maximal temperature is so high that we set the temperature threshold to be 97°C(370 in absolute temperature). In some cases, without the white space redistribution, the required temperature can not be reached. But with the post-optimization to distribute the whitespace, the required temperature can be satisfied and the number of total thermal vias is reduced significantly as well.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a LP based thermal-aware white space redistribution algorithm in 3D ICs. Based on explicit thermal profile, the thermal via requirements are estimated, then attached to blocks and finally solved by our LP approach. We have also shown that performance can be efficiently model in this approach. Experimental results show that after reallocating whitespaces, the amount of thermal vias and total wirelength could be reduced by 14% and 2%, respectively. In addition, to show the trade-off between via planning and performance, we take Alpha 21264 microprocessor as a design driver, it shows that whitespace distribution with only via planning will degrade performance by about 9% while performance-aware method can reduce thermal number by 60% while the performance is remained similar to the original design. Future works may be focused on the more accurate estimation and attachment of via requirements.

REFERENCES

[1] J.Cong, J.Wei, and Y.Zhang, "A Thermal-Driven Floorplanning Algorithm for 3D ICs", in Procceedings of ICCAD, 2004

[2] P.H.Shiu, R.Ravichandran, S.Easwar, and S.K.Lim, Multi-layer Floorplanning for Reliable Sytem-on-Package," ISCAS, 2004

[3] B. Goplen and S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs using a Force Directed Approach", Proc. of ICCAD, Nov. 2003

[4] S. Das, et al. "Performance, and Computer Aided Design of Three Dimensional Integrated Circuits", Proc.of ISPD, 2004

[5] J. Cong and Y. Zhang, "Thermal-Driven Multilevel Routing for 3-D ICs," Proc. of ASPDAC, pp 121-126, Jan.2005

[6] J. Cong and Y. Zhang, "Thermal Via Planning for 3D ICs", Proc. of IEEE/ACM ICCAD, Nov.2005

[7] T.Y. Chiang, S.J. Souri, C.O. Chui, and K.C. Saraswat, "Thermal Analysis of Heterogeneous 3D ICs with Various Integration Scenarios," Tech. Dig. IEDM, pp. 681-684, 2001

[8] B. Goplen and S. Sapatnekar. "Thermal Via Placement in 3D ICs," Proc. of ISPD, pp 167-174, Apr. 2005

[9] Z.Li, X.Hong, et. al, "Efficient thermal via planning approach and its application in 3D floorplanning," IEEE Trans. Computer-Aided Design, 2007

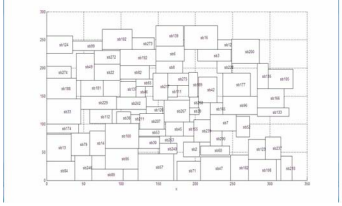
[10] http://gnuwin32.sourceforge.net/packages/glpk.h

[11] P. Wilkerson, et al. Compact Thermal Modeling Analysis for 3D Integrated Circuits. 11th International Conference Mixed Design of Integrated Circuits and Systems, Szczecin, Poland, June. 2004

[12] M. Ekpanyapong, et al., "Profile-guided Microarchitectural Floorplanning for Deep Submicron Processor Design," in Proc. ACM/IEEE DAC, pp. 634–639, Jun. 2004.

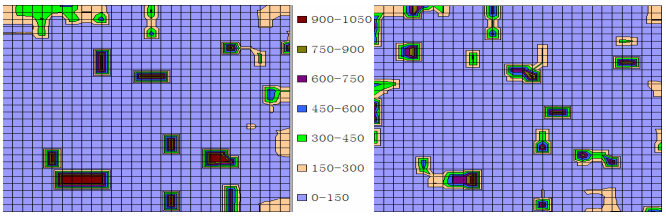
[13] Yuchun Ma, et al., "Micro-architecture Pipelining Optimization with Throughput-Aware Floorplanning", Proc. of ASPDAC, 2007

[14] X. Hong, S. Dong, et al., "Corner Block List: An effective and Efficient Topological Representation of Non-Slicing Floorplan", in proc. of ICCAD, pp.8-12, Nov., 2000.



(a)no white space redistribution (b) white space redistribution

Figure 7 white space redistribution of top layer of n100



(a) no white space redistribution (b) after white space redistribution

Figure 8 thermal via distribution of top layer of n200

Table 1: Results of White Space Redistribution for Thermal Via Planning

		m-ADVP only			m-ADVP + white space redistribution			
	Init T(°C)	T_{max}(°C)	T-via#	WL(um)	T_{max}(°C)	T-via#	WL(um)	Cpu(s)
n100	330.52	76.84	18151	73283	77.05	16353	70211	3.9
n100	406.06	84.42	16700	73639	76.97	12699	68713	3.8
n200	374.78	76.92	20423	159020	77.07	19035	162830	15.6
n200	378.80	76.91	20205	156146	76.94	16028	159670	16.4
n300	405.77	76.89	23214	243682	77.13	17363	243682	33.7
n300	405.70	77.15	23259	268837	76.92	23154	246811	32.8
Avg.			1	1		0.86	0.98	

Table 2: Results of White Space Redistribution with Performance-Optimization and Thermal Via Planning

examples				No white space redistribution			With white space redistribution					
							T-via optimization only			T-via + performance optimization		
Layer#	Freq.(GHz)	T_{max}(°C)	bips	T_{max}(°C)	bips	T-via#	T_{max}(°C)	bips	T-via#	T_{max}(°C)	bips	T-via#
3	1	161.57	0.97	77.11	0.97	12332	77.12	0.87	6207	77.01	0.97	8248
3	2	296.16	1.80	104.80	1.80	246407	77.96	1.80	59136	77.01	1.96	63904
4	1	227.08	0.82	77.01	0.82	18319	76.98	0.78	10592	76.86	0.82	10229
4	1	204.70	0.97	77.10	0.97	20957	76.80	0.82	7490	76.97	0.89	7876
4	2	383.42	1.80	103.72	1.80	323890	77.07	1.77	68155	77.08	1.93	84409
4	3	560.12	2.29	141.73	2.29	524949	97.29	1.81	158720	97.36	2.15	203194
Avg.			1		1	1		0.91	0.35		1.01	0.4