Leakage-Aware Performance-Driven TSV-Planning Based on Network flow Algorithm in 3D ICs*

Kan Wang[1], Sheqin Dong[1], Yuchun Ma[1], Goto Satoshi[2], Jason Cong[3]
[1]Dept. of Computer Science and Technology, Tsinghua University, Beijing, China, 100084
[2]Information Production and Systems, Waseda University, Kitakyushu, Japan, 808-0135
[3]Dept. of Computer Science, UCLA, Los Angeles, CA, 90095
[1]E-mail: wangkan09@mails.tsinghua.edu.cn

Abstract

3D IC has become an attractive technology as it decreases interconnection distance and improves performance. However, it is faced with heat dissipation and temperature problem seriously. The high temperature will increase the interconnection delay, and lead to degradation of performance. Through-silicon-via (TSV) has been shown as an effective way to optimize heat distribution. However, the distribution of TSVs will potentially influence the interconnection delay. In this paper, we propose a performance-driven 3D TSV-planning (3D-PTSP) algorithm, which can generate good TSV distribution, to improve temperature. The thermal effects on critical path delay are analyzed with leakage power-temperature-delay dependence considered. A priority based TSV redistribution algorithm and network flow based signal via allocation algorithm help to improve both TSV number and critical path delay without increasing temperature. Experimental results show that the proposed method can improve total via number by 8.9% and reduce critical path delay by 15.8%.

Keywords

Leakage power, Critical path delay, Leakage power-temperature-delay dependence, Signal TSV allocation

1. Introduction

Compared to 2D circuit, 3D ICs [1] are faced with heat dissipation and temperature problem seriously, which has become a bottleneck in 3D circuit design. Many thermal-driven 3D floorplanning and placement methods, such as [3-8], are proposed to solve the thermal bottleneck. Unfortunately, even with complicated thermal-aware approaches, the maximum on-chip temperature is still too high for the circuit to operate properly [9].

Then a new thermal cooling technology, TTSV(Thermal-Through-Silicon-Via), which is one of the efficient ways to reduce the chip temperature to a satisfactory level, arises as times call for [2,13]. Before long, many TSV-based floorplanning or placement methods [10-12] were proposed to improve thermal in 3D ICs.

TSVs not only effect in heat dissipation, but also play a role of vertical interconnection among layers, referred as [1]STSV (Signal TSV). Making full use of STSVs can greatly reduce the total wire length of interconnection. Due to these effects, more TSVs are wished to reduce thermal and interconnection distance. However, as the common pitch of TSV is also very large, TSVs will occupy the dead-space and create routing congestion. What's more, manufacturing them is very expensive. As a result, it is quite necessary to optimize the total via number.

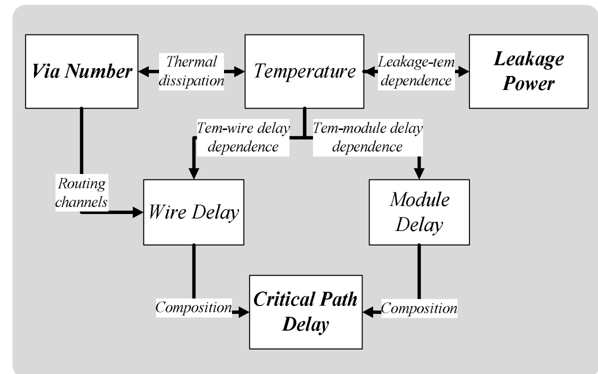


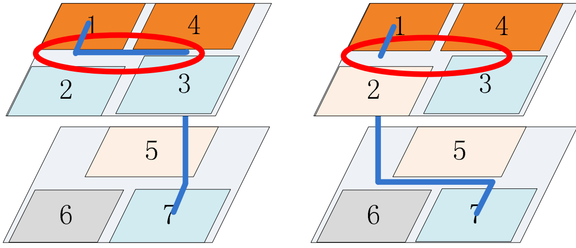
Figure 1: Relationship between leakage, temperature, via number and critical path delay

Under the circumstances, many thermal aware TSV-planning and routing technologies were proposed [9, 13-16], aiming to reduce total via number and total wire length. However, there are two main disadvantages in these researches. First, none of them consider the impact of leakage power on TSV distribution and critical paths. The work in [5] showed that with continuous shrinking of minimal feature size, leakage power has been already non-negligible. As the main composition of static power, leakage power can increase temperature greatly, by nearly 40% in current design [19]. The increased temperature can result in larger module delay and interconnection delay according to the dependence of temperature-module delay and temperature-interconnection delay [17, 21]. At the same time, the increased temperature would lead to more via number and various via distribution, which will also influence the interconnection delay. Without thinking of these relationships together (as shown in Figure 1), TSV-planning and routing results will be misestimated seriously.

Another disadvantage is that all of previous work paid no attention to temperature- interconnection delay dependence and just tried to optimize the total wire length instead of critical path delay. As is well known, path delay includes module delay and interconnection delay. To optimize critical path delay, both of them require to be optimized. Besides, the temperature-interconnection delay dependence, which is closely related to interconnection delay [21], is also necessary to be considered in the routing stage. As shown in

* This work is supported by NSFC 61176022 and the international cooperation project of MOST 2011DFA60290

Figure 2, Module 1 and 4 are two hot modules, and dead-space nearby has higher temperature. There are two possible routes to connect from module 1 to module 7: one routing along the higher die (a) and one along the lower die (b). Considering temperature-delay dependence, delays of the two paths can be different in spite of the same wire lengths. Without considering these, the routing will not be able to optimize the critical path delay.



(a) Routing in higher layer (b) Routing in the lower layer

Figure 2: Example of temperature aware routing

With continuous shrinking of minimal feature size, interconnect delay has become more and more and can be more than 50% of the total delay under 45nm [20]. A good TSV distribution can play a role in interconnection delay optimization. However, as a result of the dead-space constraint, the TSV number is limited for both temperature and interconnection optimization. How to balance them with least TSV number is a serious problem. In this paper, we insert the total required number of TSVs to reduce heat first, and then redistribute them in each dead-space by removing redundant ones in cooler regions and adding more of them to the "critical" regions. Priorities are used to control the process. We call this as priority based *TSV redistribution*. More detail information is presented in section V.

Min-cost maximum network flow algorithm has shown good performance on routing and path selection in both 2D and 3D. [14-16] treated TTSVs and STSVs together and proposed an approach to optimize the temperature and total wire length in two separate steps. However, these methods just focused on total wire length rather than critical path delay. And they did not consider the impact of leakage power and temperature-interconnection dependence on critical path as shown in Figure 1 and Figure 2.

In this paper, we propose a leakage-aware performance-driven TSV-planning based on network flow algorithm to improve both TSV number and critical path delay. We first analyze the thermal effects on delay of the critical paths with leakage-temperature-timing dependence considered. Then we propose a priority based TSV redistribution algorithm to improve via number and via distribution, which takes temperature-wire delay dependence into consideration. After that, a network flow based signal via allocation algorithm is proposed for interconnection, aiming to optimize the critical path delay. We treat "TTSV" and "STSV" together as "TSV" in TSV-planning stage and treat STSV individually in the allocation stage. The contributions are summarized as follows:

1. Integrating the impact of leakage power on critical path delay into TSV-planning. Different from previous work, we take leakage power-temperature dependence into account during TSV planning and analyze the impact of leakage power on critical path for the first time. Experimental results show that with leakage power considered, the critical path delay can be greatly changed.

2. Proposing a leakage-aware performance-driven 3D TSV-planning (*3D-PTSP*). In this paper, we propose the architecture of *3D-PTSP*, including priority based TSV redistribution and STSV allocation to optimize both total via number and critical path delay. Experimental results show that *3D-PTSP* can achieve a good trade-off among temperature, via number and critical path delay.

3. Proposing a priority based STSV allocation for 3D routing considering temperature-wire delay dependence. Different from previous routing, temperature-wire delay dependence is taken into account in this paper. A network flow based STSV allocation algorithm is proposed, which can optimize critical path delay without increasing total TSV number and wire length. Experimental results show that the proposed algorithm can improve critical path delay by about 15.8%.

The rest of the paper is organized as follows. In Section II, we describe problem and motivation of this paper. In Section III and IV, thermal estimation and delay estimation models are introduced, with which we analyze the impact of leakage-temperature-delay and TSV distribution on critical path delay. Network flow based TSV-planning algorithm, including priority based priority based TSV redistribution and signal TSV allocation, is proposed in Section V. Experiment results are shown in Section VI, and the conclusions are provided in Section VII.

2. Problem definition

Given a 3D floorplan with certain number of blocks $\{b_1, b_2, \ldots b_n\}$ on L layers with width, height, power density aware, and a netlist N between the blocks, our approach tries to insert the TSVs properly with leakage power and performance (delay) considered. Then STSVs allocation is performed from the TSV-planning result and routing is done to evaluate the critical path delay. The object is just to optimize critical path delay, total via number, temperature with certain constraints.

The notations of the TSV planning problem are shown in Table I.

Table I: notation used in TSV planning

$VN_{i,j,k}$	number of vias inserted in $grid_{i,j,k}$
$UVN_{i,j,k}$	number of signal vias in $grid_{i,j,k}$
A_{via}	area of one *TSV*
A_{uvia}	area of one *STSV*
$Area^{ds}_{i,j,k}$	area of dead-space in $grid_{i,j,k}$
$T_{i,j,k}$	temperature on $grid_{i,j,k}$
$target_T$	target maximal on-chip temperature
WL_{path}	total wire length on a path
$Delay_{path}$	delay on a path
$Delay_threshold$	the delay constraint set for the design

With the above notations, we can define the problem as:
Objective:

$$Min \ \alpha \cdot \sum_{g_{i,j,k}} VN_{i,j,k} + \beta \cdot Max\{Delay_{path}\} \tag{1}$$

a. Temperature constraints

$$Max\{T_{i,j,k}\} \le required_T \quad (2)$$

b. Timing constraints

$$Max\{Delay_{path}\} \le Delay_threshold \quad (3)$$

c. Thermal resource constraints

$$A_{via} \times VN_{i,j,k} \le Area^{ds}_{i,j,k} \quad (4)$$

d. STSV allocation constraints

$$A_{uvia} \times UVN_{i,j,k} \le A_{via} \times VN_{i,j,k} \quad (5)$$

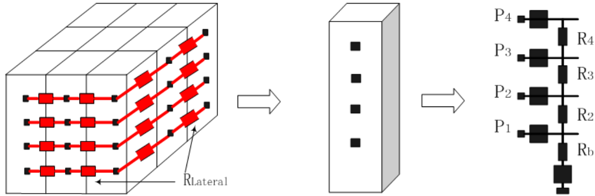
TSV planning is to minimize (1) subject to equation (2) (3) (4) and (5). With leakage-performance-temperature dependence considered, not only the total via number and via distribution will be influenced, but also the optimization approach needs to be extended.

3. Thermal and delay estimation

3.1. Thermal Estimation Model

Thermal analysis is the simulation of heat transfer through heterogeneous material among heat producers (e.g., transistors) and heat consumers (e.g. heat-sinks attached to an IC). Similar to [9], the 3D circuit stacking is divided by a two-dimensional array of tile stacks, as shown in Figure 3 (a). Each tile stack is 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). P_i presents power density of the same tile stack at the i-th layer.

As shown in Figure 3, the 3D circuit stacking is divided by a two-dimensional array of tile stacks. Each tile stack is composed of several vertically-stacked tiles. These tiles are connected by lateral thermal resistances $R_{lateral}$ and vertical resistor R_i for layer i.



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

Figure 3: Resistive thermal model for a 3D IC [9]

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. The system can be spatially discretized and be solved using the following equation to determine the steady-state thermal profile as a function of power profile:

$$T = PA^{-1} \quad (6)$$

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

3.2. Leakage-temperature dependence model

The relation between leakage power and TSV insertion is a chicken-egg problem and leakage power increases with temperature super-linearly [19]. The work in [17] modeled the leakage current with a third-order polynomial which can describe the dependencies very well. In this paper, we use this model to calculate leakage power, as shown followed:

$$\frac{I_{leakage}(T)}{I_{leakage}(T_0)} = 1 + \alpha_1 \cdot (T - T_0) + \alpha_2 \cdot (T - T_0)^2 + \alpha_3 \cdot (T - T_0)^3 \quad (7)$$

where $I_{leakage}(T)$ is the leakage power under the current temperature T. α_1, α_2 and α_3 are empirical coefficient that have different values for different technologies. Typically, α_1=0.0226, α_2=0.00033, α_3=1.77e-6, and in this paper, we define $I_{leakage}(T_0)$=0.01 whenT_0=0°C. The maximum error of this model is shown to be just 5% for the temperature range between 0°C and 250°C.

3.3 Critical path delay estimation

As is well known, the path delay consists of module delay and wire delay. Previous work has shown that both module delay and wire delay are relative to temperature but few of them consider the two delays together. In this paper, we apply two temperature-delay models to calculate both module and wire delay to get more accurate values.

3.3.1. Module delay model

In this paper, we consider both module delay-temperature and wire delay-temperature dependence. An accurate module delay-temperature dependence model can be expressed as (8).

$$delay(T) = \frac{delay(T_0)(V_{DD} - V_{TH}(T_0))^{\alpha} T^{\beta}}{T_0^{\beta}(V_{DD} - V_{TH}(T_0) + k(T - T_0))^{\alpha}} \quad (8)$$

Where k=k_0+γ(T-T_0). T is the absolute temperature in Kelvin, α is the velocity saturation index, and μ is the mobility.

Here we use the values of α, β, γ, k_0 in [17]. The maximum error of these values is just 6.1%.

3.3.2. Wire delay model

The work from [21] has shown that interconnect delay is also relative to temperature and increase with temperature greatly. In this paper, we use this temperature-dependent distributed RC interconnect delay model as shown followed:

$$D = D_0 + (c_0 L + C_L)\gamma_0 \beta \int_0^L T(x)dx - c_0 \gamma_0 \beta \int_0^L x \cdot T(x)dx \quad (9)$$

where:

$$D_0 = R_d(c_0 L + C_L) + (c_0 \gamma_0 \frac{L^2}{2} + \gamma_0 L C_L) \quad (10)$$

The parameters are valued according to [21] at room temperature (25°C). We use the thermal model tile stack to divide the total interconnection wire into several parts, and each part has a length equal to the width of each tile. Over each interconnection part we assume the temperature is uniform and equal to the peak temperature of the referred tile.

In this paper, the performance of the chip is evaluated by the total delay (both module delay and wire delay) on the longest critical path. Therefore, to meet the timing constraints in (3), the delay on the critical path should not exceed the maximum delay allowed in circuits. To estimate the wire delay, a network flow based 3D router [17] is

implemented to fulfill the wire connection.

4. Impact of leakage-temperature-delay on critical path

As mentioned in Section I, module delays are relative to temperature and increase as temperature increases. With leakage-temperature dependence considered, the critical path can be larger than ever expected. However, most of previous researches didn't think of the impact of leakage power on thermal profile, which will definitely make the estimation not exact.

On the other hand, traditional TSV insertion method just inserted TSVs in hot spot instead of thinking of the influence of TSV distribution for critical path, which cannot optimize the wire length and critical path delay.

In this paper, we take all these factors into account and propose a method to solve them. We first research the impact of leakage-temperature-delay dependence and TSV distribution on critical path delay in this section and then propose a network flow based TSV-planning algorithm to reach a good trade-off among via number, temperature and critical path delay .

Table II shows the impact of leakage-temperature dependence on temperature and critical path delay. ***Path_delay*** stands for critical path delay, including both module delay and wire delay. We compare the delay value with leakage power considered (***CLP***) to that without leakage power considered (***NLP***). We estimate leakage power and temperature according to the leakage model and thermal model in Section III.

Table II: Impact of leakage-temperature dependence on critical path delay

Test cases	***Initial T***(°C)	***T with leakage***(°C)	***Path_Delay (NLP)***	***Path_Delay (CLP)***
Ami33	388.9	435.5	166.967	193.402
Ami49	215.438	245.342	145.021	156.359
N100	191.358	261.11	360.653	394.059
N200	185.65	267.695	478.749	524.939
N300	258.683	396.905	617.944	684.176
Ratio	1	1.2961	1	1.104

From Table II we can see that, with leakage power considered, both the maximal on-chip temperatures and critical path delays increase greatly. Temperature increases by about 29.6% on average, and the critical path delay can be changed by about 10.4%. This delay estimation error can lead to serious problem, because the critical delay may exceed the cycle threshold. In order to reduce the impact of leakage power, we should insert additional TSVs around hot spots.

5. Network flow based TSV-planning algorithm

In this section, we will introduce a priority based TSV planning (*3D-PTSP*) to optimize both TSV number and critical path delay, including priority based TSV distribution and network flow based STSV allocation algorithm.

5.1. 3D-PTSP algorithm flow

The algorithm flow of *3D-PTSP* is shown in Figure 4. Given an initial solution, take turns to do thermal analysis of leakage power to get the current chip temperature distribution and estimate grid priorities and module priorities for the following initial TSV insertion and allocation. Priority based TSV redistribution and network flow based STSV allocation are then applied to reduce TSV number and get the optimized critical path delay. *3D-PTSP* is repeated with updated grid priorities, until there is no better solution. Temperature and critical path delay are calculated according to thermal model and delay estimation model.

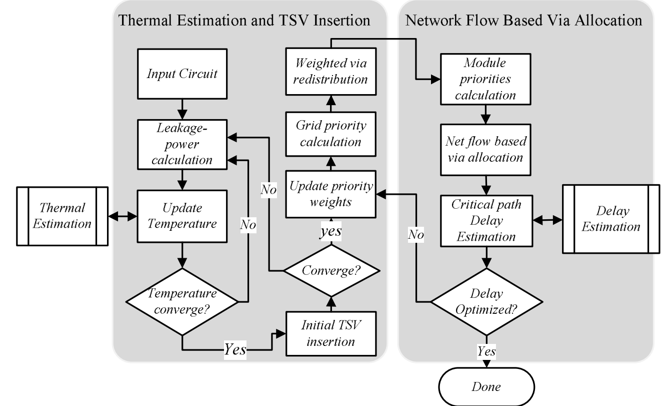


Figure 4: Flow of *3D-PTSP*

5.2. Dead-space priority definition

To guarantee critical path delay, we should leave enough TSVs in the bounding box of critical path. As a result of the limited dead-space constraint, temperature and critical path delay will compete with each other for the limited TSVs. In order to alleviate the competition, we increase the number of TSVs in the more "critical" regions and reduce the number in "noncritical" regions. Our previous work [19] proposed a weighted method to insert TSVs. In this paper, we improve this method in a perspective of dead-space priority and take priority of bounding box into account.

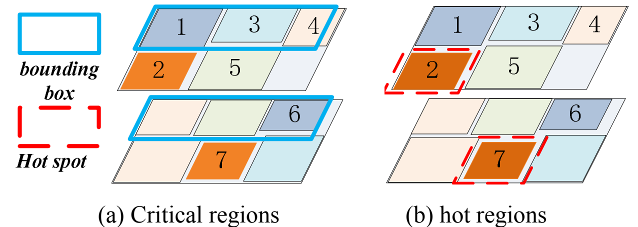


Figure 5: TSV insertion regions

First, we bring in priorities to guarantee the wire length of critical path. As shown in Figure 5, Module 1 and 6 are along critical path. Modules 2 and 7 are two hot modules where temperatures are higher than the others. The solid line box shows the bounding box between module 1 and 6, as shown in Figure 5(a), while the dashed line box shows the hot spots, as shown in Figure 5(b). All dead-spaces in these bounding boxes have higher priority to get more TSVs while TSVs in the cooler and noncritical regions can be reduced to save cost. Based on the priority mechanism, the total TSVs can be redistributed in a more appropriate way.

Furthermore, we pay attention to the "critical path candidates", whose delay is not far from critical path. These

paths are all possible to be the real critical path with leakage power and via positions considered. Therefore, when doing TSV insertion, all the critical path candidates are taken into account and optimized.

5.2.1. Grid priority

Grid priority (*GP*) is used to evaluate the dead-space in grids. TSVs would be first inserted in the dead-space of grids with higher priority. The priority is evaluated by the effect of leakage power and delay, formulated by:

$$GP_{i,j,k} = \alpha \times \frac{T_{i,j,k}}{T_{\max}} \times \frac{P_{i,j,k}}{P_{\max}} + \beta \times \frac{T_{i,j,k}}{T_{\max}} \times \frac{delay(T_0)}{Delay_threshold} \quad (11)$$

T_{max} is the maximal on-chip-temperature before TSV insertion. $T_{i,j,k}$ is the temperature on $grid_{i,j,k}$. $P_{i,j,k}$ is the dynamic power density of $grid_{i,j,k}$, and P_{max} is the maximum power density for all grids. $delay(T_0)$ is the average delay of modules in $grid_{i,j,k}$ at the temperature of T_0, while *Delay_threshold* is the target maximal delay on critical path. α and β are constant factors defined by users. Here, we set α=β=0.2.

On the other hand, an additional number of vias are required to guarantee the wire length (wire delay) of the critical path, which should be in dead-space of the bounding box along critical path. For this, we bring in another priority into *grid priority* $B(g_{i,j,k})$ to add to delay criticality, which stands for whether $g_{i,j,k}$ is in critical bounding box or not, valued 0 and 1, The grids in critical bounding boxes have higher priority to get vias. $B(g_{i,j,k})$ reflects the impact of leakage and module delay on TSV distribution to some extent.

$$B(g_{i,j,k}) = \begin{cases} 1 & g_{i,j,k} \text{ is in critical bounding box} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Here we combine the weights with the heat flow for each grid as:

$$H_{i,j,k} = (GP_{i,j,k} + \gamma \cdot B(g_{i,j,k})) \cdot I_{i,j,k} \quad (13)$$

Where the heat flow $I_{i,j,k}$ for each grid is updated through path counting approach according to [9]. γ is also constant factor and we set γ = 1.Then the TSV number initially assigned to each tile is proportional to $H_{i,j,k}$ instead of $I_{i,j,k}$.

5.3. Priority based TSV redistribution

In this paper, we use a priority based via redistribution to reduce the maximum temperature and optimize the critical path delay.

We distribute the priorities on blocks to the nearby grids which have available dead-space and then TSV insertion will be guided by the weight values along with the heat dissipation. Here we combine the weights with the heat flow for each grid as (13). Ideally without the dead-space resources, for two tiles on a layer k, the number of TSVs allocated follows:

$$VN'_{i1,j1,k} : VN'_{i2,j2,k} = H_{i1,j1,k} : H_{i2,j2,k} \quad (14)$$

Considering the resource constraint, the TSV number in a grid is:

$$VN_{i,j,k} = \min\left(\frac{Area^{ds}_{i,j,k}}{A_{via}}, VN'_{i,j,k}\right) \quad (15)$$

Therefore, the vias can be inserted between blocks by iteratively updating the number of vias in each grid to reach the target temperature or meet some stop criteria.

5.4. Network flow algorithm based signal TSV allocation

The work in [16] proposed a routing algorithm based on network flow algorithm to do interlayer via planning and pin assignment. However, it cannot be used directly as signal via allocation in this paper, as there are several disadvantages:

First, the routing in [16] just aimed to optimize the total wire length and didn't consider the critical path delay, which is definitely not a competent method for signal via allocation.

Second, it didn't consider temperature-wire delay dependence as shown in Figure 2. Therefore, the routing results cannot be optimized.

What's more, it paid no attention to the impact of leakage power. As a result, the TSV-planning result is not exact.

In this paper, we consider all of above and propose a more effective routing. We first insert TSVs to optimize temperature and TSV distribution with priority based TSV redistribution, and then allocate enough vias as interlayer STSVs based on min-cost network flow algorithm to optimize critical path delay. We solve the STSV allocation problem in the post-floorplan stage.

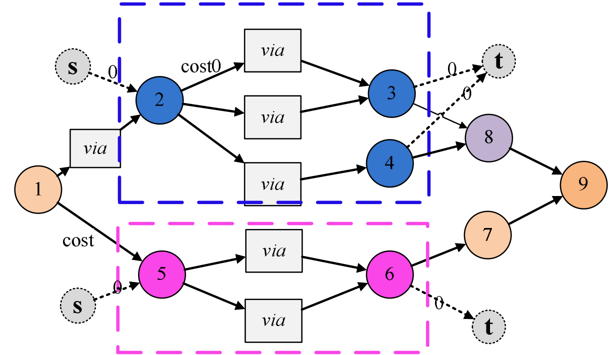


Figure 6: network graph for TSV allocation

5.4.1. Network flow algorithm definition

Let G_0 (N_0, A_0) be the network graph for 3D ICs, where N_0 stands for all the modules and A_0 are the set of interconnection between modules. To solve the routing problem, we add every via as via node into the graph, and then generate the interconnection between module and the available vias nodes with cost according to delay of the module, wire delay and priority of modules. The global network model can be presented as $G(N, A)$ with a cost c_{ij}, a capacity u_{ij} associated with every arc $(i, j) \in A$, as shown in Figure 6. The problem can be formulated as:

Objective:

$$Min \sum c_{ij} x_{ij} \quad (16)$$

Subject to:

$$\sum x_{ij} - \sum x_{ji} = \begin{cases} 1 & if \ node \ i \ is \ source \\ -1 & if \ node \ i \ is \ sink \\ 0 & otherwise \end{cases} \quad (17)$$

$$0 \le x_{ij} \le u_{ij} \quad (18)$$

The cost c_{ij} is defined by:

$$c_{ij} = w \cdot WD_{ij} + m \cdot (MD_i + MD_j s_j) + p \cdot MP_i \quad (19)$$

where WD_{ij} is wire delay between node i and node j, which is calculated according to Section 5.1.2; MD_{ij} is the module delay of node i and we set the values of *via nodes* to 0; MP_i is the module priority of node i while s_j shows whether node j is the sink node. w, m and o are custom-defined factor and we set all the factors valued w=0.4, m=0.4 and p=0.2.

5.4.2. Module priority

We introduce the priorities for ordering in TSV allocation, named as *MP* (*Module priorities*). *MP* is evaluated by module's slacks, two-pin wire length, local temperature and capacity in grids nearby. We wish to allocate vias for the modules with higher priorities to ensure the path delay which go through these modules. According to this ordering mechanism, the critical path delay can be guaranteed. It can be formulated as followed:

$$MP_i = s \times \frac{Delay_threshold}{slack_i} + l \times \sum_{i \in path} WL_{path} + t \times \sum cap_{grid_i} \quad (20)$$

where $slack_i$ is the value of required time subtracted by arrived time without wire delay[18] and evaluated according to both module delay and estimated wire delay using half perimeter model. When a module on the path has high priority, it means that wires through this module should be first guaranteed.

5.4.3. Signal TSV allocation algorithm

Based on the module priority, signal via allocation algorithm can be performed to allocate TSVs for signal interconnection. To do this, a network graph is generated as Section 5.4.1. However, directly performing network flow algorithm on the global network diagram will take a lot of time. For this, we propose a heuristic strategy to speed up the routing process.

Stage 1: sub-graph partition for speedup

We classify the whole graph into several sub-graphs and run the network flow in each sub-graph for speedup. A single sub-graph consists of an input module node, its out-degree module nodes and all available grid nodes. Without loss of generality, we create each sub-graph with an additional source node and sink node, as shown in dashed line box in Figure 6. The total number of sub-graphs is no more than the number of modules.

Stage 2: criticality ordering

As the number of vias is limited, the order of routing is quite important. The modules which are along critical path candidates should obtain the nearest vias first to guarantee the delay on critical path candidates. After sub-graphs generated, a global sort of all the graphs is done according to the priorities of their modules and the total via number in this sub-graph. We route for the critical sub-graphs first and allocate TSV for interconnection in these sub-graphs. Then we allocate for the rest sub-graphs with the remaining unused vias. The definition of criticality is shown as followed:

$$Criticality_i = v \cdot viaDen_i + n \cdot \sum_{j \in Subgraph_i} MP_j \quad (21)$$

$viaDen_i$ shows the via density in $subraph_i$. v and n are user-defined factors, and here we set them to 0.5.

Stage 3: local routing optimization algorithm

In each sub-graph, we perform the min-cost max-flow algorithm based routing to get the best allocation result. After each sub-graph network problem is solved, the used via nodes are removed and criticalities will be recalculated for the unsettled sub-graphs.

The flow of network flow algorithm in this paper is shown as Figure 7. We first calculate the capacity of interlayer TSVs in each grid node region and calculate the *module priority* for each module. Then construct the network graphs and assign capacity and cost to each edge. Sub-graphs are generated from the network for simplification. Finally, we use improved min-cost maximum flow algorithm to do via allocation on each sub-graph and store results.

6. Experimental results

In this section, we show the effect of *3D-PTSP*. Experimental results show that it is effective on improving via number and critical path delay, without increasing the peak temperature.

All experiments were performed on a workstation with 3.0 GHz CPU and 4GB physical memory. We use five typical MCNC and GSRC benchmarks [3] in our experiments. A four-device layer configuration is assumed for all circuits. The parameters of TSVs between layers are set as [9]. The floorplan layout is generated by a 3-D thermal-driven floorplanning tool [3].

```
Input: gridVias, module_delay, module_number // total via number in each grid , delay of modules, and total module number
Output: wireLength_ij //Calculate the wire length between module i and module j
Method:
For (i between 1 and module_number){ //calculate priorities of all the modules
 gridCap=gridCapCalculation(gridVias); //calculate capacity of grids referred to module i
 slacki=slackCalculation();//calculate the slack of module i
 M_priorityi=priorityCalculation(slacki, gridCap); //calculate the priority of module i
}
priorityOrdering();//all the modules are sorted according to priority
For (i between 1 and module_number){ //net flow with ordered priority
 minCostFuntion (i); //run the min-cost maximum flow with i as source module
 extractResult (i);// extract results which store the wireLength_ij
}
Output wireLength_ij;
```

Figure 7: Flow of signal via allocation algorithm

6.1. Effect of priority based TSV redistribution

In order to compare the results before redistribution and after redistribution, we divide the results into two groups: ***BR*** (*Before Redistribution*) and ***AR*** (*After Redistribution*). We enhance the leakage weights to improve TSV number to show the effect of TSV redistribution and pitch of STSVs is adjusted according to the circuit size to make the comparisons more fairly. Table III shows the results on N200 with initial temperature 185.7^{o}C.

From Table III, we can see that our approach can improve both via number and critical path delay without increasing the peak temperature. After redistribution, total via number can be reduced by 8.9% at most and 5.7% on average. At the same time, the critical path delay is improved by about 6.9% at most and 4.3% on average. The lower temperature is required, the more significant effect on via number is

created. With *target_T* increases, more significant effect on critical path delay is made.

Table III: Effect of TSV redistribution on via number and critical path delay of N200 with initial temperature 185.7 °C

Required Tem(°C)	*Before Redistribution (BR)*			*After Redistribution (AR)*		
	Max T(°C)	*Via Num*	*Path Delay*	*Max T(°C)*	*Via Num*	*Path Delay*
77	77.1	8544	426.05	77.1	8232	409.87
87	87.1	6764	434.28	89.3	6159	416.94
97	97.1	5560	445.82	97.5	5298	425.94
107	106.8	4731	467.52	106.8	4459	458.09
117	117.2	4054	513.85	117.8	3807	478.51
Ratio	1	1	1	1.002	0.943	0.957

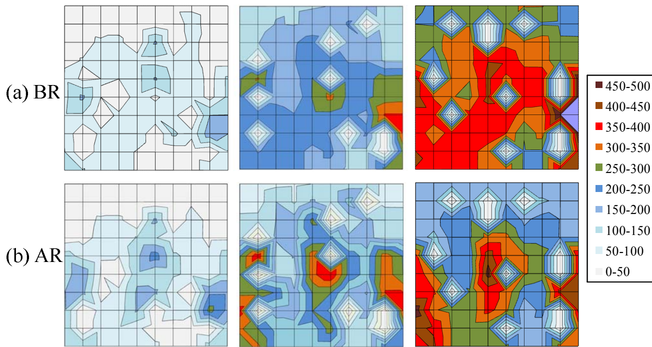


Figure 8: TSV distribution for n200

Figure 8 shows the TSV distribution for n200 on 3 layers under *target_T* =117 °C in two patterns: before redistribution (a) and after redistribution (b). Layer 0 is not shown in this figure because no vias are needed. The regions with darker color are filled with more vias, while the lighter ones are inserted with less vias. From Figure 8 we can see that after redistribution, more vias are inserted on layer 1 and layer 2 in hot spots and critical regions, which aims to guarantee total TSV number along the critical path. At the same time, less TSVs are inserted in cooler regions to improve TSV number. In this way, the redistribution method can improve TSV distribution to optimize total TSV number and critical path delay, without increasing peak temperature.

6.2. Network flow algorithm based signal TSV allocation

After TSV redistribution, both the total via number and via distribution can be improved. Then network flow algorithm based TSV allocation is performed to decrease wire length and optimize the critical path delay in 3D routing stage.

Table IV: Comparisons between 3D-PTSP algorithm and previous work

Test cases	*Initial T(°C)*	***NP*+[16]**				***RP*+[16]**				***3D-PTSP***				***Run Time (s)***
		Max T(°C)	*Via num*	*Critical Delay*	*Wire length (×10³)*	*Max T(°C)*	*Via num*	*Critical Delay*	*Wire length (×10³)*	*Max T(°C)*	*Via num*	*Critical Delay*	*Wire length (×10³)*	
Ami33	388.9	117.0	550	145.54	38719	117.1	551	143.73	38654	117.1	551	141.10	38802	4.59
Ami49	215.4	117.1	13450	146.59	626459	119.6	13001	128.96	626492	119.6	13001	123.41	626702	5.12
N100	191.4	117.1	2679	414.85	118494	116.8	2569	385.38	118730	116.8	2569	378.56	119012	10.19
N200	185.7	117.1	4053	513.85	324114	117.7	3807	478.51	324276	117.7	3807	465.25	324612	21.04
N300	258.7	116.9	16638	564.11	480674	117.0	15895	550.60	480569	117.0	15895	534.77	481293	32.36
Ratio		1	1	1	1	1.005	0.959	0.945	1.000	1.005	0.959	0.920	1.000	

We compare the results of the following three kinds of patterns: ***NP*** (*Normal Pattern*): traditional TSV insertion method which only aims to reduce thermal and TSV allocation in [16] without considering critical path delay; ***RP*** (*Redistributed Pattern*): priority based TSV redistribution method which improves both via number and via distribution, with TSV allocation method in [16]; ***3D-PTSP***: TSV redistribution and STSV allocation method which pay attention to critical path delay and use ordering mechanism .

In order to make the results more significant, we reduce *via_ratio* of each grid to 0.2, which means that the total space for TSVs is only 20% of the total dead-space area. The allocation results are shown in Table IV.

From Table IV we can see that our redistribution method ***RP*** can improve total via number by 4.1% and reduce the critical path delay by about 5.5% on average, compared to ***NP***. With criticality ordering mechanism, ***3D-PTSP*** can further improve critical path delay. Take n200 for example, without TSV redistribution method, the routing can create 513.9 of delay. While in ***3D-PTSP***, it creates only 465.3, improved by 9.5%. In other test cases, ***3D-PTSP*** can also improve delay, by about 8.0% on average and 15.8% at most, without increasing the max temperature. Furthermore, neither of ***RP*** and ***3D-PTSP*** increases the total wire length.

7. Conclusion

In this paper, we first research the impact of thermal on critical delay and take temperature-leakage-timing dependence into consideration, proposing a TSV planning method to make the best trade-off among temperature, via number, and critical path delay. TSV redistribution mechanism is to optimize TSV number while the network flow algorithm based signal via allocation aims to optimize the critical path delay. Experimental results show that, the proposed approach is effective on improving via number and critical path delay, without increasing the peak temperature and total wire length.

8. References

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

[2] J. H. Lau, T. G. Yue, “Thermal Management of 3D IC Integration with TSV (Through Silicon Via)”, Electronic Components and Technology Conference, 2009, Pages: 635–640, San Diego, CA.

[3] J. Cong, J. Wei and Y. Zhang, “A Thermal-Driven Floorplanning Algorithm for 3D ICs”, in Proceedings of ICCAD, 2004, USA.

[4] L. Xiao, S. Sinha, J.Y. Xu, F.Y. Young, “Fixed-outline Thermal-aware 3D Floorplanning”, ASP-DAC, 2010, Pages: 561-567, Japan.

[5] P. Zhou, Y. Ma, Z. Li, R.P. Dick, L. Shang, H. zhou, X.L. Hong and Q. Zhou, “3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three Dimensional Integrated Circuits”, In proceedings of ICCAD, 2007.

[6] W-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. J. Irwin, “Thermal-Aware Floorplanning Using Genetic Algorithms”, ISQED, 2005.

[7] K. Balakrishnan, V. Nanda, S. Easwar, Sung Kyu Lim, “Wire congestion and thermal aware 3D global placement”, ASP-DAC, 2005, Pages: 1131 - 1134, Japan.

[8] W. L. Hung, G.M. Link, Yuan Xie, N. Vijaykrishnan, M.J. Irwin, “Interconnect and thermal-aware floorplanning for 3D microprocessors”, ISQED, 2006, San Jose, CA.

[9] J. Cong and Y. Zhang, “Thermal Via Planning for 3D ICs”, Proc. Of IEEE/ACM ICCAD, Nov. 2005.

[10] Z Li, X Hong, Q Zhou, S Zeng, J Bian, “Efficient Thermal via Planning Approach and Its Application in 3-D Floorplanning”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Pages: 645-658, April 2007.

[11] J Cong, G Luo, J Wei, “Thermal-Aware 3D IC Placement Via Transformation”, ASP-DAC, 2007, Pages: 780-785, Japan.

[12] B. Goplen, S. Sapatnekar, “Placement of 3D ICs with Thermal and Interlayer Via Considerations”, DAC, 2007, Pages: 626-631, CA

[13] T Zhang, Y Zhan, Sachin S. Sapatnekar, “Temperature-Aware Routing in 3D ICs”, ASPDAC, 2006, pages: 309-314, Japan.

[14] J. Cong, Y. Zhang, “Thermal-driven multilevel routing for 3-D ICs”, ASPDAC, 2006, Pages: 121-126, Japan.

[15] H. Xiang, X. P. Tang and D. F. Wong, “Min-cost flow-based algorithm for simultaneous pin assignment and routing”, IEEE Trans. Computer-Aided Design, vol. 22, pp 870-878, July. 2003.

[16] X He, S Dong, X Hong, S Goto, “Integrated Interlayer Via Planning and Pin Assignment for 3D ICs”, SLIP, 2009, Pages: 99-104, USA

[17] H. Hua, C. Mineo, K. Schoenfliess, A. Sule, S. Melamed, R. Jenkal, and W. R. Davis, “Exploring Compromises among Timing, Power and Temperature in Three-Dimensional Integrated Circuits”, DAC, 2006, Pages:997 – 1002, San Francisco.

[18] X. Qiu, Y. Ma, X. He, X. Hong, “IPOSA: A Novel Slack Distribution Algorithm for Interconnect Power Optimization”, ISQED, 2008, Pages: 873 - 876, San Jose, CA.

[19] K. Wang, Y. Ma, S. Dong, Y. Wang, X. Hong, J. Cong, “Rethinking Thermal Via Planning with Timing-Power-Temperature Dependence for 3D ICs”, ASPDAC, 2011, Japan.

[20] T. Mitsuhashi, T. Aoki, M. Murakata, “Physical Design CAD in Deep Sub-micron Era”, EURO-DAC 1996, Switzerland.

[21] A.H.Ajami1, K.Banerjee, M.Pedram1, L.P.P.P van Ginneken, “Analysis of Non-Uniform Temperature-Dependent Interconnect Performance in High Performance ICs”, DAC, 2001, June 18-22, 2001, Las Vega, Nevada, USA.