Temperature-Aware Routing in 3D ICs *

Tianpei Zhang, Yong Zhan, and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering, University of Minnesota
zhangt, yongzhan, sachin@ece.umn.edu

Abstract

Three-dimensional integrated circuits (3D ICs) provide an attractive solution for improving circuit performance. Such solutions must be embedded in an electrothermally-conscious design methodology, since 3D ICs generate a significant amount of heat per unit volume. In this paper, we propose a temperature-aware 3D global routing algorithm with insertion of "thermal vias" and "thermal wires" to lower the effective thermal resistance of the material, thereby reducing chip temperature. Since thermal vias and thermal wires take up lateral routing space, our algorithm utilizes sensitivity analysis to judiciously allocate their usage, and iteratively resolve contention between routing and thermal vias and thermal wires. Experimental results show that our routing algorithm can effectively reduce the peak temperature and alleviate routing congestion.

1 Introduction

By stacking multiple device layers into a monolithic structure, three-dimensional integrated circuits (3D ICs) can achieve higher transistor densities and shorter interconnect lengths than two-dimensional (2D) ICs, and work as a platform to integrate different components and provide good substrate isolation among them [1]. Despite the advantages of 3D ICs over 2D ICs, thermal effects are expected to be significantly exacerbated in 3D ICs due to higher power density and greater thermal resistance of the insulating dielectric, and this can cause greater degradation in device performance and chip reliability which have already plagued 2D ICs [1]. Thus, it is essential to develop 3D-specific design tools that take a thermal co-design approach so as to address the thermal effects and generate reliable and high performance designs. This work considers the problem of developing routing solutions for 3D ICs.

To address thermal issues in global routing, an efficient and accurate thermal computation method is needed. Previous work on on-chip thermal analysis falls into the following categories: the finite difference method (FDM) and the finite element method (FEM) as in [3–5], and Green function method as in [6, 7]. In this paper, we utilize the finite difference based thermal circuit model presented in [3] to obtain fast temperature and sensitivity analysis. Several 3D routing algorithms have addressed the problems of 3D channel routing [8], 3D maze routing [9], 3D Field-Programmable Gate Array (FPGA) routing [10], and 3D hierarchical routing [11]. However, none of these approaches considers the thermal problem associated with 3D ICs in the routing phase. A first approach to this problem is presented as a thermally-driven 3D routing algorithm in [12] with the planning of thermal vias to reduce temperature. However, it does not fully address the contention issues between thermal vias and routing resources.

In this paper, we propose a novel 3D routing algorithm which can effectively reduce on-chip temperatures by appropriate insertion of "thermal vias" and a new construct that we call "thermal wires," and generate a routing solution free of thermal and routing capacity violations. Thermal vias correspond to vertical interlayer vias that do not have any electrical function, but are explicitly added as thermal conduits. Thermal wires perform a similar function, but conduct heat laterally within the same layer. Thermal vias perform the bulk of the conduction to the heat sink, while thermal wires help distribute the heat paths over multiple thermal vias. The routing scheme begins with routing congestion estimation and signal interlayer via assignment, followed by thermally-driven maze routing. Sensitivity analysis is employed and linear programming (LP) based thermal via/wire insertion is performed to reduce temperature. The above process is iteratively repeated until temperature and routing capacity violations are resolved. Experimental results show that the scheme can effectively resolve the contentions between thermal via/wire and routing, generating a solution satisfying both congestion and temperature requirements.

*This work was supported in part by DARPA under grant N66001-04-1-8909, and NSF under award CCR-0205227.

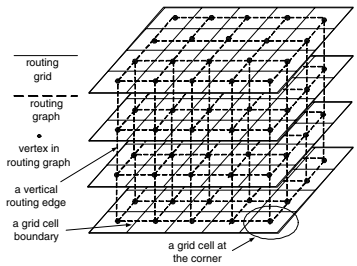

Figure 1: Routing grid and routing graph for a four-layer 3D circuit.

2 3D IC Global Routing Model

2.1 3D Interconnects and Routing Model

In 3D ICs, devices are fabricated on a number of active layers, which are separated by silicon dioxide and joined by an adhesive material. Within each device layer, interconnections among devices can be achieved with traditional interconnect wires and vias. Connections between active layers are facilitated by vertical interconnect vias that span through multiple layers, providing a means for electrically connecting wires in those layers. This type of via is different from a regular 2D via: in particular, it is significantly taller than conventional vias, and has a larger landing pad to maintain a viable aspect ratio. We refer to such vias as *interlayer vias*.

The multiple-layer structure of 3D IC complicates the global routing problem, but we may extend some basic constructs from 2D routing to 3D. As shown in Figure 1, our global routing model tessellates each layer in the 3D circuit into a two-dimensional array, referred to as the *routing grid*. A net N consists of a set of electrically equivalent pins, $\{n_0, n_1, n_2, ..., n_k\}$, distributed in different routing grid cells (possibly in different active layers), of which n_0 is the source and $n_1, n_2, ..., n_k$ are sinks. If a net has pins distributed on different layers, it is referred to as an *interlayer net*; otherwise it is called a *2D net*.

The dual graph of a 2D routing grid tessellation is a *2D routing graph*. By stacking 2D routing graphs of all layers and connecting each pair of vertically neighboring nodes with a vertical routing edge, we build a *3D routing graph*, denoted as G, which is shown by the dashed lines in Figure 1, and the routes for each net are to be determined along the edges of this 3D routing graph. There are two classes of edges in the graph: (i) Each lateral edge in the 3D routing graph corresponds to a boundary e_{ij} in the routing grid that connects grid cells i and j. Due to geometrical limitations on each lateral boundary, we require that $W_e \leq C_e$, in which W_e is the total width (including wire spacing) used by all wires, and C_e is the geometrical width of boundary e, or the *boundary capacity*. Any violation of this requirement results in a *boundary overflow*. (ii) Each vertical edge in the 3D routing graph corresponds to the interlayer vias connecting two vertically neighboring grid cell positions. Since interlayer vias go through the active device layer(s), and can only use the white space that is not occupied by devices, there is an upper limit on the number of interlayer vias for each vertical edge in the routing graph, referred to as *interlayer via capacity* U_i, and it is determined by the white area of grid cell i through which the interlayer vias pass, and the size of the landing pad of the interlayer via. We require $V_i \leq U_i$, in which V_i is the actual number of interlayer vias through grid cell i.

0-7803-9451-8/06/$20.00

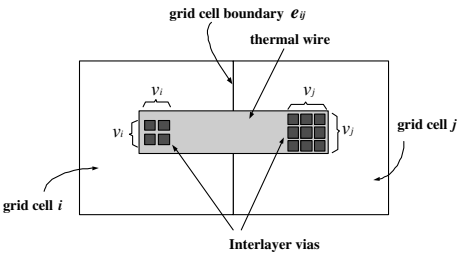

Figure 2: Reduction of lateral routing capacity due to the interlayer vias in neighboring grid; thermal wires are lumped, and together with thermal vias, form a thermal dissipation network.

2.2 Thermal Vias and Thermal Wires

The silicon dioxide layer acts as a thermal insulator that strongly inhibits heat flow, potentially leading to elevated temperatures. Interlayer vias connecting different active layers serve to to conduct heat and alleviate hot spots in 3D ICs [13, 14]. Straightforward extensions of traditional routing scheme to 3D are inadequate, since they leave out several new 3D-specific complications. The locations of the interlayer vias that carry signals from one layer to another should not be determined purely by the signal wire routing process, but should also incorporate thermal considerations, determining the most desirable interlayer via positions from the point of view of heat conduction. Moreover, a sufficient number of interlayer vias must be used so that the temperature can be reduced to below the target level.

Our work uses two types of vias: (i) thermal vias are deliberately introduced interlayer vias that serve no electrical function, but are dedicated to the purpose of temperature reduction [15], and (ii) signal interlayer vias, which carry electrical signal, and thus perform signal and heat conduction simultaneously. Like signal interlayer vias, thermal vias can be planned in each grid cell position i between two layers. We require that V_i, the total number of signal interlayer vias and thermal vias going through grid cell i, be smaller than its capacity U_i, defined above. Here we assume that the interlayer vias for power grid are predetermined and there are dedicated resources for them, which are excluded from U_i.

Although interlayer vias can effectively reduce the on-chip temperature, their large size also acts as a significant routing blockage; as more heat removal is achieved, less routing space is available, and this problem has been identified in the packaging of multi-chip modules (MCMs) [16]. Figure 2 shows how interlayer vias can reduce the routing capacity of a neighboring lateral routing edge. If $v_i \times v_i$ interlayer vias pass through grid cell i, and $v_j \times v_j$ interlayer vias pass through the adjacent grid cell j, the signal routing capacity of boundary e_{ij} will be reduced from the original capacity, C_e, to a reduced capacity, $C_{e,red.}$, and we require that signal wire usage W_e should satisfy:

$$C_{e,red.} = \min\left(C_e - v_i \cdot w, C_e - v_j \cdot w\right)$$
$$W_e \leq C_{e,red.} \tag{1}$$

where w is the geometrical width of an interlayer via. Here we define the smaller of the two reduced routing widths as the reduced edge capacity, so that there can be a feasible translation from the global routing result to a detailed routing solution. On the other hand, given the actual signal wire usage W_e of a routing edge, we can also use equation (1) to determine how many interlayer vias can go through the neighboring grid cells so that there is no overflow at the routing edge. Since temperature reduction requires insertion of a large number of thermal vias, careful planning is necessary to meet both the temperature and routability requirements.

Just as thermal vias enhance vertical heat conduction, we can also introduce *thermal wires* to improve lateral heat conduction. The lateral routing tracks in the circuits are used by power grid wires and signal wires. For simplicity of analysis, we assume a predetermined power grid architecture as well as dedicated routing tracks for power wires, and they are excluded from the routing capacity calculation. Along a routing edge, signal wires may not utilize all of the routing tracks, and we employ the remaining tracks to connect the thermal vias in adjoining grid cells with *thermal wires*. Like thermal vias, thermal wires do not carry signal, and therefore, they can be connected directly to thermal vias to form an efficient heat dissipation network as shown in Figure 2. Thermal wires enable the conduction of heat in lateral direction, and can thus help vertical thermal vias to reduce hot spots temperature efficiently: for those

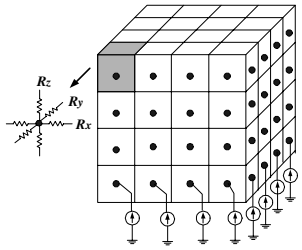

Figure 3: Equivalent thermal circuit model of a 3D IC.

hot spots where only a restricted number of thermal vias can be added, we can use thermal wires to conduct heat laterally, and then remove heat through thermal vias in adjoining grids. Note that although signal wires can also conduct heat laterally, they must be separated from the thermal vias by an oxide that has high thermal resistivity, and are therefore not efficient in lateral heat removal. Moreover, unlike signal routes that only use a small number of interlayer signal vias, the thermal vias form a global net and are therefore very effective in heat removal. Another advantage of thermal wires is improved design for manufacturing. Filling the remaining tracks with thermal wires can create a dense metal fill, which improves manufacturability and performance predictability which may be deteriorated by the Chemical-Mechanical Planarization (CMP) step. Since thermal wires contend for lateral routing resources with signal wires, they should be well planned to satisfy the temperature and routability requirements. Along a routing edge e with total routing capacity C_e, the number of signal wires s_e and the number of thermal wires m_e should satisfy

$$(s_e + m_e) \cdot w_t \leq C_e \tag{2}$$

where w_t is the width of a routing track.

2.3 Temperature-Aware 3D Global Routing Problem

Our formulation of the temperature-aware 3D global routing problem is as follows. The inputs to the problem include:

1. The result of placement for a 3D IC
2. 3D technology parameters
3. A specified value of the maximum temperature, T_{spec}

Our global routing algorithm will route the circuit netlist and efficiently insert thermal vias and thermal wires at appropriate locations so that the solution satisfies the requirements on the routing capacities and the interlayer via capacities. At the same time, the peak temperature will be lowered towards T_{spec}. Moreover, for reasons related to reliability and to the limited availability of interlayer via resources, we require that the number of signal interlayer vias used in routing should be minimized.

3 Thermal Analysis Model

Equivalent thermal circuit models have been used extensively in the simulation of thermal effects because of their clear physical origins, ease of implementation, and flexibility in handling different kinds of boundary conditions and non-uniform thermal conductivity. Our thermal analysis uses a grid of the same size as that of global routing. We model a 3D chip as a resistive thermal network and solve the corresponding linear equation to obtain the temperature of each node. As shown in Figure 3, we use a node to represent all of the heat dissipating devices in a global routing grid cell, using an equivalent current source that represents the total power generated in the cell. Adjacent nodes are connected by a thermal resistance, forming a thermal resistive grid. In this work, we only consider power dissipation from active devices, since heat generated by interconnect wires is small because of their relatively small resistance; however, it is possible to extend the approach to incorporate this.

The numerical value of thermal resistance connecting to a node in x, y, and z directions can be computed as:

$$R_x = R_{oxide,x} || R_{int.,x} || R_{mix,x}$$
$$R_y = R_{oxide,y} || R_{int.,y} || R_{mix,y}$$
$$R_z = R_{oxide,z} || R_{int.\ via,z} || R_{mix,z} \tag{3}$$

where the symbol "||" refers to a parallel connection between the resistors. The thermal resistance in each direction is determined by the materials contained in the cuboid volume extending to the neighboring

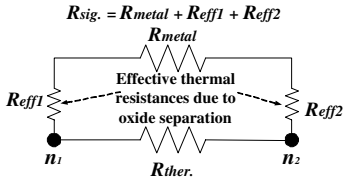

Figure 4: Thermal circuit model of signal wire and thermal wire.

grid cell and can be calculated as parallel connections of three parts. $R_{oxide,x}$, $R_{oxide,y}$ and $R_{oxide,z}$ are the oxide thermal resistance of pure oxide extending to neighboring node in x, y and z directions respectively. The terms $R_{int.,\,x}$ and $R_{int.,\,y}$ are the thermal resistance of interconnect wires running along x and y directions, respectively. $R_{int.\ via,\,z}$ is the thermal resistance of the vertical interlayer vias which act as one of the major heat conduction paths, and it is proportional to the number of interlayer vias connecting the vertically neighboring grid cells. Along each direction, besides pure oxide and metal wires or vias extending in that direction, the remaining volume is a mixture of metal wires/vias extending in the orthogonal directions and the oxide separating them; along the direction we are considering, alternate metal and oxide form a "sandwich" structure. We denote the expected thermal resistance of such a mixture as $R_{mix,x}$, $R_{mix,y}$ and $R_{mix,z}$ for x, y and z direction respectively, and their values are calculated as series thermal resistance of the metal wires/vias (running in orthogonal directions) and the oxide which separates the metal.

To differentiate between the the heat conducting abilities of signal wires and thermal wires, we use the model depicted in Figure 4. We model a signal wire of thermal resistance $R_{sig.}$, connecting neighboring grid cells n_1 and n_2 as series connection of metal resistance R_{metal} and two additional resistances R_{eff1} and R_{eff2} connecting to the thermal via nodes at the center of n_1 and n_2. R_{eff1} and R_{eff2} correspond to the effective thermal resistance of the oxide which separates signal wires from thermal vias. The effective oxide thickness is estimated by a probabilistic approach which assumes a uniform distribution of signal wires crossing a grid cell boundary. By comparison, the thermal resistance of a thermal wire connecting n_1 and n_2 only consists of the metal resistance $R_{ther.}$.

The vertical edges of the 3D chip are modeled to be adiabatic to the ambient, and the bottom of the chip is connected to an isothermal heat sink by a bulk substrate. With the thermal circuit established, the nodal analysis (NA) equations for the thermal grid can be set up as

$$M\mathbf{T} = \mathbf{P} \tag{4}$$

where *M*, **T** and **P** are the NA coefficient matrix, the temperature profile vector and power dissipation vector for the grid cells, respectively. This equation is sparse and symmetric positive definite (SPD), and we use the LASPACK package [17] as the linear solver to obtain the temperature profile efficiently.

To effectively remove temperature violations in circuit under constrained resources, it is very important that we insert thermal vias and thermal wires in the right places; otherwise even a large number of them can not effectively reduce the high temperatures. Thus, we perform adjoint sensitivity analysis which can help determine the potential thermal via and thermal wire locations where insertion of a thermal via/wire can reduce hot spot temperature effectively. Adjoint sensitivity analysis has the advantage of obtaining the sensitivities of one output value to many parameters simultaneously and efficiently. We now briefly describe the calculation of the sensitivity of the temperature at a hot spot to the thermal via density as an example; sensitivity analysis for hot spot temperature to thermal wire density can be performed similarly.

The temperature profile of a 3D circuit is determined by equation $M\mathbf{T} = \mathbf{P}$ as described above. For a hot grid cell in the circuit with temperature T_i at the i^{th} position of vector **T**, the sensitivity of T_i to matrix element M_{kl} is expressed as follows [18]:

$$\frac{\partial T_i}{\partial M_{kl}} = -\xi_{ik} T_l \tag{5}$$

in which $\xi_{\mathbf{i}}$ is the solution of equation $M^T \xi_{\mathbf{i}} = \mathbf{e_i}$, and $\mathbf{e_i}$ is a column vector with all zeros except for a one in its i^{th} row. We can then obtain the sensitivity s_{ij} of the temperature T_i to the number of interlayer vias at location j, V_j, by applying the chain rule:

$$s_{ij} = \frac{\partial T_i}{\partial V_j} = \sum_{M_{kl} \in S_j} \frac{\partial T_i}{\partial M_{kl}} \frac{\partial M_{kl}}{\partial V_j} \tag{6}$$

where set S_j contains all of the entries, M_{kl}, that depend on V_j. There is a small number of such entries in each S_j, since the V_j term only appears in the vertical and lateral thermal conductance related to this location (thus, there are at most 12 entries in each S_j since each conductance appears four times in matrix M). This sensitivity calculation indicates how effective thermal via insertion can be in reducing high temperatures, and we can use it to guide thermal via insertion. Similarly, we can perform sensitivity analysis for hot spot temperature to thermal wire density and identify the most effective location for thermal wire insertion. Due to the advantage of adjoint sensitivity and fast iterative solution of LASPACK, practically the complexity of obtaining sensitivity information is only $O(RG)$, in which R is the number of temperature violation spots, and G is the number of nodes in the 3D routing graph.

4 3D Temperature-Aware Global Routing

In addition to the traditional 2D-like challenges of wire length and congestion, 3D global routing must deal with the complexity introduced by higher dimensional routing and by temperature constraints. Our global routing approach resolves these challenges in two phases. Phase I first performs signal interlayer via assignment, and then utilizes a thermally-driven 2D maze routing algorithm in each layer to generate an initial routing solution. Based on that, phase II resolves temperature and congestion violations iteratively: it identifies hot spots and sensitivity information, and then judiciously inserts thermal vias and thermal wires in sensitive locations; the insertions, together with rip-up-and-reroute are iteratively performed until both congestion and temperature violations are resolved. In the above rip-up-and-reroute, we process one net at a time and maintain a fixed order of all nets. We have experimentally found that, under different randomly chosen net orderings, the results change very little as long as we maintain the same fixed net order through all of the iterations. This is due to the fact that early iterations are seen to create good estimates of resource utilization, and this reduces the order dependence. This is consistent with the observation in [20]. Since thermal effects pose a significant problem in 3D ICs and the thermal via and thermal wire insertion for temperature reduction is a core part of our algorithm, we will first describe routing phase II in our paper; the initial routing procedures of phase I and the overall flow are discussed following that.

4.1 Thermal Via and Thermal Wire Insertion for Temperature Reduction

In an initial global routing solution generated from phase I, the high power density and low thermal conductivity of silicon dioxide can result in hot spots in 3D ICs, and we can employ thermal vias and thermal wires to build effective heat conduction paths. Assuming a predetermined distribution of wires and interlayer vias for power lines, we can analyze the heat conduction for a 3D IC. The role of the thermal via and thermal wire insertion algorithm is to reduce the peak temperature towards T_{spec}, and at the same time, generate a routable design. This task must take into account the routing blockages and resource utilization (via capacity and wire capacity) as these are added. Our algorithm iteratively inserts thermal vias and thermal wires with a linear programming (LP) approach, and performs a rip-up-and-reroute step to obtain a solution free from temperature and congestion violations.

4.1.1 Thermal Via and Thermal Wire Insertion Algorithm

Figure 5 shows the flow of the algorithm. The LP-based thermal via and thermal wire insertion and rip-up-and-reroute are iteratively performed until there is no violation or no further improvement. The latter is easily identified if it is detected that the congestion map changes insignificantly, or the peak temperature reduces trivially. In each iteration, we relax the specified temperature T_{spec} to a target temperature $T_{target} = T_{max}^{\mu} \cdot T_{spec}^{1-\mu}$ in identifying temperature violation, where T_{max} is the current highest temperature, and μ is a constant with value $0 < \mu < 1$ (we take $\mu = 0.2$ in practice); also if this calculated value of T_{target} is smaller than $T_{spec} + \Delta T_{min}$, we assign $T_{target} = T_{spec}$, where ΔT_{min} is a positive constant. By introducing ΔT_{min} we can avoid infinite number of temperature decreases before T_{target} reaches T_{spec}; in practice, we set ΔT_{min} = 5°C. As the final step we greedily insert thermal vias and thermal wires using all of the remaining space.

Algorithm: `Thermal_via/wire_insertion_for_temperature_reduction`
Input: Initial 3D global routing result and specified peak temperature requirement T_{spec}.
Output: A routing solution with thermal via and thermal wire insertion that is free from congestion and temperature violation.
begin
DO
{
Update temperature profile, get current highest temperature T_{max};
Update target temperature T_{target} according to T_{max} as described in Section 4.1.1;
Perform sensitivity analysis for temperature violating spots with $T > T_{target}$;
Construct and solve LP formulation: equations (7), (8), (9), (10) and (11);
Insert thermal vias and thermal wires and update congestion information;
Rip-up-and-reroute for resolving overflow;
} **UNTIL** (free from temperature and congestion violation or no further improvement)

Post insertion of thermal vias and thermal wires using all available space;
end

Figure 5: Thermal via and thermal wire insertion algorithm for temperature and congestion reduction.

4.1.2 Linear Programming Based Thermal Via and Thermal Wire Insertion

In the thermal via and thermal wire insertion algorithm of Figure 5, we first perform thermal analysis to update the temperature profile and identify hot spots that violate the temperature specifications. The thermal via and thermal wire planning procedure is then formulated as a LP problem to reduce the temperature of hot spots and minimize total thermal via and thermal wire usage, while leaving ample space for lateral routing.

For each of n temperature-violating hot spots with temperature $T_i > T_{target}$, $i = 1, 2, ..., n$, we perform sensitivity analysis for T_i with respect to the number of thermal vias at each thermal via location as in Section 3, and this can be performed fast due to the advantage of adjoint sensitivity analysis. However, to reduce the complexity of the LP problem, we only record those non-trivial sensitivity values $s_{v,ij} \geq s_{th}$, in which s_{th} is a constant threshold of sensitivity and in practice we take $s_{th} = 0.01$°C per via; the associated location j is then defined as a candidate thermal via location, and $j = 1, 2..., p$, where p is the total number of candidate thermal via locations. Similarly, we can define candidate thermal wire location and obtain the sensitivities $s_{w,ik}$ for T_i with respect to the number of thermal wires at each candidate location k, and $k = 1, 2, ..., q$, in which q is the total number of such locations. Based on the sensitivity analysis results, we can economically plan the number of inserted thermal vias, $N_{v,j}$, at each candidate thermal via location j, and the number of inserted thermal wires, $N_{w,k}$, at each candidate thermal wire location k, so that the temperature at each hot spot i can be reduced by ΔT_i. The LP problem is formulated as follows:

$$\text{minimize} \quad \sum_{j=1}^{p} N_{v,j} + \sum_{k=1}^{q} N_{w,k} + \Gamma \sum_{i=1}^{n} \delta_i \qquad (7)$$

$$\text{subject to:} \sum_{j=1}^{p} -s_{v,ij} N_{v,j} + \sum_{k=1}^{q} -s_{w,ik} N_{w,k} + \delta_i \geq \Delta T_i,$$

$$i = 1, 2, ..., n, \ \Delta T_i = T_i - T_{target} \qquad (8)$$

$$N_{v,j} \leq \min((1+\beta)R_{v,j}, \ U_j - V_j), \ j = 1, 2, ..., p \qquad (9)$$

$$N_{w,k} \leq (1+\beta)R_{w,k}, \ k = 1, 2, ..., q \qquad (10)$$

$$\delta_i \geq 0, \ i = 1, 2, ..., n; \ N_{v,j} \geq 0, \ j = 1, 2, ..., p;$$
$$N_{w,k} \geq 0, \ k = 1, 2, ..., q \qquad (11)$$

The objective function that minimizes the total usage of thermal vias and wires is consistent with the goal of routing congestion reduction. To guarantee the problem is feasible, we introduce relaxation variables δ_i, $i = 1, 2...n$. Γ is a constant which remains the same over all iterations. It is chosen to be a value that is large enough to suppress the value of δ_i to be 0 when the thermal via and thermal wire resources in constraint (9) and (10) are enough to reduce temperature as desired.

Constraint (8) requires the temperature reduction at hot spot i, plus a relaxation variable δ_i, to be at least ΔT_i during the current iteration, where ΔT_i is the difference between current temperature T_i and target temperature T_{target}, and δ_i is introduced to ensure that the problem is feasible. Constraints (9) and (10) are related to capacity constraints on the thermal vias and thermal wires, respectively, based on lateral boundary capacity overflows in the same layer, and interlayer via capacity overflows across layers. Constraint (9) sets the upper limit for the number of thermal via insertions $N_{v,j}$ with two limiting factors. $R_{v,j}$ is the maximum number of additional thermal vias that can be inserted at location j without incurring lateral routing overflow on neighboring edge, and it is calculated as $R_{v,j} = v_j - v_{cur.,j}$, in which $v_{cur.,j}$ is the current interlayer via usage at location j, and v_j is the maximum number of interlayer vias that can be inserted at location j without incurring lateral overflow which can be calculated from equation (1). Adding more interlayer vias in the most sensitive locations can be very influential in temperature reduction, and therefore, we intentionally amplify the constraint by a factor β to temporarily permit a violation of this constraint, but which will allow better temperature reduction. This can potentially result in lateral routing overflow after the thermal via assignment, but this overflow can be resolved in the iterative rip-up-and-rerouting phase; in practice, we find $\beta = 0.1 \sim 0.2$ works well. A second limiting factor for $N_{v,j}$ is that the total interlayer via usage can not exceed U_j, which is the interlayer via capacity at position j, and we take the minimum of the two limiting factors in the constraint formulation. Similarly, constraint (10) sets a limit on the number of thermal wire insertions with the consideration of lateral routing overflow. $R_{w,k}$ is the maximum number of additional thermal wires that can be inserted at location k without incurring lateral routing overflow, and it is calculated as $R_{w,k} = m_k - m_{cur.,k}$, where $m_{cur.,k}$ is the current thermal wire usage at location k, and m_k is the maximum number of thermal wires at location k without incurring lateral overflow, which can be calculated from equation (2). In the same spirit of encouraging temperature reduction, we relax $R_{w,k}$ by factor of β, and any potential overflow will be resolved in the rip-up-and-rerouting phase.

The LP problem above can be efficiently solved with the *lp_solve* package [25]. The solution to this LP formulation are rounded to the ceiling integer to guarantee the optimality, and the corresponding number of thermal vias and thermal wires are then inserted into the 3D circuit; the resulting overflow can be resolved in the following rip-up-and-reroute. This LP based insertion and rip-up-and-reroute are iteratively performed until a feasible solution is reached.

4.2 Temperature-Aware 3D Global Routing Flow

We now describe the steps involved in our 3D routing algorithm. Briefly, these include a Minimum Spanning Tree (MST) generation and routing congestion estimation step, a signal interlayer via assignment step, and a thermally-driven 2D maze routing in each active layer; these steps constitute phase I of our routing algorithm. The phase II step is iterative routing involving rip-up-and-reroute and LP-based thermal via/wire insertion which has been described above in Section 4.1.

4.2.1 3D MST and Routing Congestion Estimation

A 3D net $N = \{n_0, n_1, n_2, ..., n_k\}$ can have more than two pins distributed on different layers (if not, as described earlier, we refer to it as a 2D net). We first build a minimum spanning tree, using Prim's algorithm [19], for each multiple-pin net, so as to decompose it into a set of two-pin nets. The cost function for two pins of a net in Prim's algorithm is essentially the Manhattan distance but with large weighting factor assigned before the vertical separation distance, and this large weight factor will discourage the generation of interlayer two-pin net. Therefore, the usage of this cost function can reduce total wire length and minimize the use of signal interlayer vias. With a set of two-pin nets from MST decomposition, we can statistically estimate the routing congestion over each lateral routing edge using the L-Z shape statistical routing model proposed in [21]. To extend the model in [21] for 3D routing, we assume that a two-pin net with pins on different layers has an equal probability of utilizing any interlayer via position within the bounding box defined by the pins. The estimated congestion map provides important information for the later signal interlayer via assignment stage.

4.2.2 Signal Interlayer Via Assignment

Based on the two-pin interlayer decompositions generated from 3D MST step, our approach uses a hierarchical min-cost network flow heuristic to simultaneously assign interlayer vias for all interlayer nets and minimize a cost function. This heuristic builds a group hierarchy by recursively dividing all device layers into two neighboring groups of equal size. Signal interlayer via assignment is then performed at the boundaries of group pairs at each level in a top-down way following the hierarchy: the assignment is conducted for the topmost level group boundary first, and then at the boundaries of group pairs of lower levels in the hierarchy. Figure 6(a) shows an example of signal interlayer via assignment

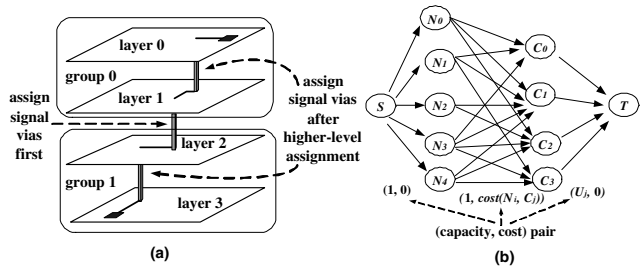

Figure 6: (a) Example of hierarchical signal via assignment for a four-layer circuit. (b) Example of min-cost network flow heuristics to solve signal via assignment problem at each level of hierarchy.

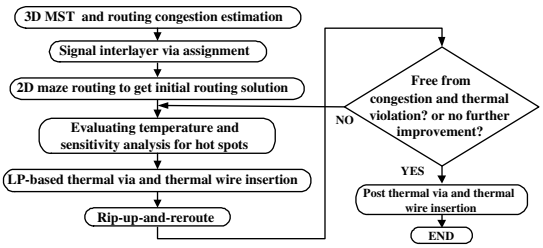

Figure 7: Overall flow for the temperature-aware 3D global routing algorithm.

for a decomposed 2-pin signal net in a four-layer circuit with two levels of hierarchy. The signal interlayer via assignment is first performed at the boundary of group 0 and group 1 at topmost level, and then it is processed for layer boundary within each group.

At each level of the hierarchy, the problem of signal interlayer via assignment is formulated as a min-cost network flow. Figure 6(b) shows the network flow graph for assigning signal interlayer vias of five interlayer nets to four possible interlayer via positions. Each interlayer net is represented by a node N_i in the network flow graph; each possible interlayer via position is indicated by a node C_j. If C_j is within the bounding box of the two-pin interlayer net N_i, we build a directed edge from N_i to C_j, and set the capacity to be 1, the cost of the edge to be $cost(N_i, C_j)$. The $cost(N_i, C_j)$ is evaluated as the shortest path cost for assigning interlayer via position C_j to net N_i when both pins of N_i are on the two neighboring layers; otherwise it is evaluated as the average shortest path cost over all possible unassigned signal interlayer via positions in lower levels of the hierarchy. The shortest path cost is obtained with Dijkstra's algorithm [19] in the 2D congestion map generated from the previous estimation step, and the cost function for crossing a lateral routing edge is a combination of edge length and a overflow cost function which is similar to that in [22]. The incorporation of overflow into cost function can guide signal interlayer via assignment to find the best location for minimizing routing congestion in each layer. The network flow graph in Figure 6(b) also includes a source node S and a target node T. There is an edge from S to every N_i with capacity 1 and cost 0; an edge is built from every C_j to target T with capacity U_j (the interlayer via capacity at C_j) and cost 0. There is a supply of N for source node S and a demand of N for target node T, where N is the number of interlayer net nodes in the flow graph. For the network flow graph, the min-cost network flow problem is optimally solved in polynomial time with the *cs2* package [23], and we can thus obtain the signal interlayer via assignment for all interlayer nets.

4.2.3 Thermally-driven 2D Maze Routing

After signal interlayer via assignment, we can decompose all interlayer nets into a set of 2D nets by adding a pseudo-pin at each signal interlayer via landing position. Moreover, with a known signal interlayer via distribution, we can perform temperature analysis as discussed in Section 3 and obtain an initial temperature map to guide 2D routing. 2D routing is performed on all nets using the maze routing algorithm from [24], The cost function is similar to that in signal interlayer via assignment, but with an additional temperature term in the cost, so that the resulting route will have an incentive to choose a low temperature path. This strategy not only decreases the temperature influence on the delay of signal nets [2], but also reduces congestion in hot regions, which is beneficial, since heuristically more thermal vias will later be inserted in these regions to reduce temperature. The overflow will be resolved by iterative rip-up and reroute. After 2D routing, we can further connect the pins at different layers for those interlayer nets by employing the previously assigned signal interlayer vias, thus build an initial 3D routing solution.

4.2.4 Overall Flow of the 3D Global Routing Algorithm

We summarize the individual pieces of the algorithm and outline the overall flow in Figure 7. The input is a tessellated 3D circuit with given power distribution. An initial 3D routing solution is found by generating a 3D MST, and then performing a network flow based interlayer via assignment, followed by thermally-driven 2D maze routing. Based on the thermal profile and sensitivity information, we then perform a LP-based thermal via and thermal wire insertion procedure. This insertion is performed iteratively; each time after the insertion, a rip-up-and-reroute step is employed to resolve the lateral routing congestion and overflow. This continues until there is no temperature and congestion violation or no further improvement is possible. The majority of the algorithm run time is on the iterative thermal via and thermal wire insertion and rip-up-and-reroute, and each such iteration has a complexity of $O(NGlogG + G^3)$, in which N is the number of nets in circuit, and G is the number of grid cells in the routing graph. Practically, it is seen that the optimized solution can be reached in a small number of iterations.

5 Experimental Results

The temperature-aware 3D router was implemented in C++ and tested on an Intel Pentium 4 2.8GHz Linux machine, on benchmarks from the MCNC [26] and IBM placement benchmark [27] suites. Table 1 lists parameters of the benchmark circuits and routing grid size for each layer. The benchmarks are placed with the placer from [5]. Four layers are used, the chip size is fixed at $5\text{mm} \times 5\text{mm}$; the oxide layer separation is $7\mu\text{m}$, and the silicon substrate is $500\mu\text{m}$ thick. The power of each circuit is randomly generated with a power density between $10\text{W}/\text{cm}^2$ to $800\text{W}/\text{cm}^2$. In practice, the power density at each cell position can be estimated with two components: dynamic power and static power. In evaluating dynamic power, the effects of interconnect wire load capacitance on switching power should be considered, and the interconnect length can be estimated with half-perimeter wire length from placement results. The white spaces of grid cells is calculated and the average is about 20% of the total chip area. The interlayer via has a size of $5\mu\text{m} \times 5\mu\text{m}$. The thermal conductivity of silicon, oxide and metal are 119W/(m°C), 1W/(m°C) and 396W/(m°C), respectively. The bottom of the chip was made isothermic with the ambient temperature to represent the heat sink, and the top and sides of the chip are assumed to be adiabatic to the ambient. The temperature of the heat sink is set to a reference value of 0°C for convenience; if the temperature is nonzero, it is well known that the calculated values can be obtained by a simple translation.

Circuit	# cells	# nets	Grid	Circuit	# cells	# nets	Grid
biomed	6417	5743	28 ×28	ibm02	19321	18429	34×34
industry2	12149	12696	31×31	ibm03	22207	21905	37×37
industry3	15059	21939	35×35	ibm04	26633	26451	36×36
ibm01	12282	11754	31×31	ibm06	32185	33521	40×40

Table 1: Benchmark circuit parameters.

The experimental results are listed in Table 2. We compare our algorithm of temperature-aware 3D global routing (denoted as *TA*) with three other 3D global routing schemes. The first comparison scheme performs initial 3D global routing in the same way as our algorithm, but after that, both thermal vias and thermal wires are post-inserted into the 3D circuit in a greedy way, under the constraint of existing routing resource usage and white space allowance, and we denote it as *P* in Table 2. The second comparison scheme follows the same routing procedure as our approach, but only thermal vias are inserted and optimized to improve heat conduction; thermal wires are not used, and we denote this scheme as *V*. The third comparison scheme uses the same number of thermal vias and thermal wires as our routing algorithm, but they are distributed uniformly across the chip. The level of thermal via distribution is equal to the average number of thermal vias per grid cell position from our algorithm, and the thermal wire level is assigned in the similar uniform way. We denote this comparison scheme as *U* in the table. For all four approaches, we set up our specified peak temperature T_{spec} to be 80°C to guide the temperature reduction process.

Circuit	T_{init} (°C)	Peak temperature (°C)				Ave. top 1% temperatures (°C)				Wire length ($\times 10^5$)				run time (seconds)			
		TA	*P*	*V*	*U*	*TA*	*P*	*V*	*U*	*TA*	*P*	*V*	*U*	*TA*	*P*	*V*	*U*
biomed	237.1	81.9	105.6	115.3	87.5	73.7	97.8	80.0	76.6	1.82	1.77	1.78	1.76	255	137	188	131
industry2	207.5	82.4	106.6	116.3	98.0	64.0	75.2	77.9	68.3	6.04	5.92	6.01	5.90	855	473	591	519
industry3	202.0	79.2	99.1	112.5	89.8	68.0	78.0	74.9	71.9	9.85	9.75	9.71	9.93	1807	1405	1686	1193
ibm01	264.8	79.1	109.9	99.3	108.4	61.7	86.9	68.7	74.4	2.63	2.46	2.63	2.56	238	87	224	87
ibm02	257.5	80.5	105.6	111.6	97.5	57.3	65.2	63.3	60.8	7.73	7.57	7.68	7.60	769	400	717	469
ibm03	218.5	82.9	106.9	123.4	85.4	68.1	78.1	77.8	67.3	10.27	10.30	10.26	10.00	1645	1490	1344	864
ibm04	218.1	80.0	96.0	107.8	84.6	68.6	73.3	78.1	68.7	8.07	8.15	8.21	8.25	886	597	974	581
ibm06	236.4	81.2	99.2	131.4	89.2	63.0	69.3	66.9	63.8	18.08	18.19	18.12	18.08	2956	1585	2274	1581

Table 2: Comparison of temperature and routing performance results among four approaches: 3D global routing using our thermal-aware (*TA*) method; using post (*P*) insertion of thermal vias and thermal wires; using thermal via (*V*) insertion only; and using uniform (*U*) thermal via and thermal wire insertion.

Routing results and temperature performance for each approach are shown in Table 2, for a set of circuits listed in the first column. The second column T_{init} lists the peak temperature after simply performing initial 3D routing but without any thermal via and thermal wire insertion. As we can see, the peak temperature is well above our expected value without insertion of thermal vias and thermal wires, which shows the necessity of these insertions in 3D ICs. The following four columns list the peak temperature for all routing algorithms. Experimental results show that our temperature-aware (*TA*) algorithm can successfully bring the peak temperatures of circuits down to specified temperature T_{spec} and performs much better than other routing schemes. The post-insertion of thermal vias and thermal wires in approach P does not optimize resource usage during routing phase and results in inferior performance; the peak temperature from P can be as high as 30.8°C more than that from *TA* algorithm (*ibm01*). Approach *V* does not employ thermal wires as an effective lateral heat conduction resource, thus for some spots where thermal via insertion is restricted from routing congestion, hot spots are generated due to the lack of lateral thermal conduction path; experimental results show a maximum peak temperature difference of 50.2°C (*ibm06*) between *V* and *TA*. The uniform insertion approach *U* applies thermal vias and wires uniformly instead of distributing them according to sensitivity analysis, therefore its performance is inferior to our *TA* routing algorithm. Moreover, it creates great routing overflow as discussed later. The next four columns listed the average temperature of top 1% hot spots for all four approaches. The results show the same tendency as peak temperature results, our temperature-aware algorithm can reduce the temperature of hot spots better than other approaches from an iterative routing and temperature reduction approach.

Circuit	# routing overflows				Circuit	# routing overflows			
	TA	*P*	*V*	*U*		*TA*	*P*	*V*	*U*
biomed	11	0	8	6750	*ibm02*	34	0	15	4262
industry2	1	0	4	6201	*ibm03*	11	0	1	16873
industry3	14	0	1	10021	*ibm04*	2	0	1	5596
ibm01	34	0	15	4262	*ibm06*	7	0	5	5404

Table 3: Comparison of routing overflow of four approaches.

Table 2 also lists the wirelength and run time results. All four approaches report similar wirelength results. Averagely, our *TA* algorithm generates 0.6% longer wirelength than approach *P*, because during iterations *TA* algorithm will detour to leave space for thermal via and thermal wire insertion for temperature reduction and thus increase wirelength; however, this increase is trivial. *TA* and *V* algorithms need longer run time, which is due to the iterations to resolve temperature violation and routing congestion. Table 3 reports the lateral routing overflow of four algorithms. The *TA* algorithm can successfully resolve routing congestion and temperature violations by employing LP-based thermal via and thermal wire insertion as well as iterative approach, and reports trivial routing overflow. However, the uniform insertion algorithm *U* distributes thermal vias and thermal wires uniformly across the chip without addressing the routing capacity reduction from them, thus approach *U* generates huge routing overflow, which makes the routing infeasible. Overall, the experimental results show that our temperature-aware algorithm can effectively reduce hot spot temperature in 3D ICs to specified value, and maintain a good wirelength and overflow property through an iterative LP-based thermal via and thermal wire insertion as well rip-up-and-rerouting.

References

[1] K. Banerjee, S. J. Souri, P. Kapur and K. C. Saraswat, "3-D ICs: A Novel Chip Design for Improving Deep Submicrometer Interconnect Performance and System-on-Chip Integration," *Proc. of IEEE*, 89(5), pp. 602-633, May 2001.

[2] K. Banerjee, M. Pedram and A. Ajami, "Analysis and Optimization of Thermal Issues in High-Performance VLSI," *Proc. ISPD*, 2001, pp. 230-237.

[3] C. H. Tsai and S. M. Kang, "Cell-Level Placement for Improving Substrate Thermal Distribution," *IEEE Trans. CAD*, 19(2), pp. 253-266, Feb. 2000.

[4] P. Li, T. Pileggi, M. Asheghi and R. Chandra, "Efficient Full-Chip Thermal Modeling and Analysis," *Proc. ICCAD*, 2004, pp. 319-326.

[5] B. Goplen and S. S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs Using a Force Directed Approach," *Proc. ICCAD*, 2003, pp. 86-89.

[6] Y. K. Cheng and S. M. Kang, "An Efficient Method for Hot-Spot Identification in ULSI Circuits," *Proc. ICCAD*, 1999, pp. 124-127.

[7] Y. Zhan and S. S. Sapatnekar, "Fast Computation of the Temperature Distribution in VLSI Chips Using the Discrete Cosine Transform and Table Look-up," *Proc. ASP-DAC*, 2005, pp. 87-92.

[8] R. J. Enbody and K. H. Tan, "Routing the 3-D Chip," *Proc. DAC*, 1992, pp. 132-137.

[9] Y. Deng and W. Maly, "Physical Design of the "2.5D" stacked System," *Proc. ICCD*, 2003, pp. 211-217.

[10] C. Ababei, H. Morgal and K. Bazargan, "Three-dimensional Place and Route for FPGAs," *Proc. ASP-DAC*, 2005, pp. 773-778.

[11] S. Das, A. Chandrakasan and R. Reif, "Design Tools for 3-D Integrated Circuits," *Proc. ASP-DAC*, 2003, pp. 53-56.

[12] J. Cong and Y. Zhang, "Thermal-Driven Multilevel Routing for 3-D ICs," *Proc. ASP-DAC*, 2005, pp. 121-126.

[13] T-Y Chiang, K. Banerjee and K. C. Saraswat, "Effect of Via Separation and Low-k Dielectric Materials on the Thermal Characteristics of Cu Interconnects," *Tech. Dig. IEDM*, 2000, pp. 261-264.

[14] T-Y Chiang, K. Banerjee and K. C. Saraswat, "Compact Modeling and SPICE-Based Simulation for Electrothermal Analysis of Multilevel ULSI Interconnects," *Proc. ICCAD*, 2001, pp. 165-172.

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

[16] S. Lee, T. F. Lemczyk and M. M. Yovanovich, "Analysis of Thermal Vias in High Density Interconnect Technology," *Proc. IEEE Semi-Therm Symp.*, 1992, pp. 55-61.

[17] http://www.tu-dresden.de/mwism/skalicky/laspack/laspack.html

[18] L. Pillage, R. Rohrer and C. Visweswarish, "Electronic Circuit and System Simulation Methods," McGraw-Hill, New York, NY, 1995.

[19] T. H. Corman, C. E. Leiseron, R. L. Rivest and C. Stein, "Introduction to Algorithms," 2nd edition, MIT Press, Cambridge, MA, 2001.

[20] R. Nair, "A Simple Yet Effective Technique for Global Wiring," *IEEE Trans. CAD*, 6(2), pp. 165-172, March 1987.

[21] J. Westra, C. Bartels and P. Groeneveld, "Probabilistic Congestion Prediction," *Proc. ISPD*, 2004, pp. 204-209.

[22] R. T. Hadsell and P. H. Madden, "Improved Global Routing through Congestion Estimation," *Proc. DAC*, 2003, pp. 28-34.

[23] A. V. Goldberg, "An Efficient Implementation of a Scaling Minimum-Cost low Algorithm",*Journal of Algorithms*, 22, pp. 1-29, 1997.

[24] C.J. Alpert, T.C. Hu, J.H. Huang, A.B. Kahng and D. Karger, "Prim-Dijkstra Tradeoffs for Improved Performance-Driven Routing Tree Design," *IEEE Trans. CAD*, 14(7), pp. 890-896, July 1995.

[25] http://groups.yahoo.com/group/lp_solve/

[26] http://www.cbl.ncsu.edu/pub/Benchmark_dirs/LayoutSynth92

[27] http://er.cs.ucla.edu/benchmarks/ibm-place/