

# Fast 3-D Thermal Simulation for Integrated Circuits With Domain Decomposition Method

Wenjian Yu, *Senior Member, IEEE*, Tao Zhang, Xiaolong Yuan, and Haifeng Qian

**Abstract**—For accurate thermal simulation of integrated circuits (ICs), heat sink components in chip package must be considered. In this letter, techniques based on the domain decomposition method (DDM) are presented for the 3-D thermal simulation of nonrectangular IC thermal model including heat sink and heat spreader. A relaxed nonoverlapping DDM algorithm is employed to convert the problem to subproblems on rectangular subdomains. Then, a nonconformal discretization strategy is proposed to reduce the problem complexity with negligible error. Numerical experiments on several 2-D and 3-D IC test cases demonstrate that the relaxed nonoverlapping DDM is faster than the other preconditioned conjugate gradient algorithms with same mesh grid. The nonconformal discretization achieves further 10× reduction of runtime and memory usage.

**Index Terms**—3-D thermal simulation, domain decomposition method, integrated circuit, irregular geometric domain.

## I. INTRODUCTION

The chip-level thermal analysis is indispensable for both sign-off verification and design-time optimization of integrated circuits (ICs) [1]. This is more important for 3-D ICs, due to more severe challenge of heat dissipation therein. Several thermal simulation algorithms have been proposed for chip-level analysis, which use the geometric multigrid solver [2] or Green's function based fast algorithms [3]. One limitation is that they consider the rectangular domain of die, with a simplified boundary assumption (often the Dirichlet condition) accounting for the effect of heat sink components. In practice, the heat spreader and heat sink attached to the die are much wider than IC die (Fig. 1). By approximating the whole thermal system with a single rectangular domain, substantial error (up to tens of degrees in temperature) may be introduced [4], [5]. Algorithms were recently proposed for the thermal simulation of 3-D ICs [6], [7], where the nonhomogeneous thermal conductivity due to through silicon vias (TSVs) was incorporated. However, only the rectangular simulation domain was considered in them. For the realistic pyramid-shape IC model, the algorithms would become inapplicable or inefficient.

In [5], a preconditioned conjugate gradient (PCG) algorithm was proposed for the realistic pyramid IC model, which takes the solution of a larger and approximate rectangular domain by fast Poisson solver (FPS) as the preconditioner. It is called FPS-PCG algorithm, whose efficiency is however related to the

Manuscript received February 5, 2013; revised May 21, 2013; accepted July 12, 2013. Date of current version November 18, 2013. This work was supported in part by NSFC under Grant 61076034, Tsinghua University Initiative Scientific Research Program, and Fundamental Research Funds for the Central Universities under Grant 2011JBZ002. This paper was recommended by Associate Editor P. Li.

W. Yu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: yu-wj@tsinghua.edu.cn).

T. Zhang and X. Yuan are with the Department of Electronic Science and Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: stu.zhangtao@gmail.com; yuanxl@bjtu.edu.cn).

H. Qian is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: qianhaifeng@us.ibm.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2013.2273987

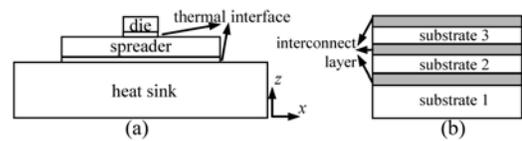


Fig. 1. Pyramid-shape IC geometry for thermal simulation. (a) Side view. (b) Details of a 3-D IC including three tiers of dies.

degree of irregularity in the realistic geometry. And, because a uniform discretization is imposed on the larger rectangular domain to enable the fast Poisson solver, it causes more unknowns, and therefore more runtime and memory usage.

In this letter, we investigate the techniques based on the domain decomposition method (DDM) for fast thermal simulation of the pyramid-shape model of 2-D and 3-D ICs. We employ the relaxed nonoverlapping DDM, which divides the whole model into three rectangular subdomains, and enables fast convergence. For each subdomain, fast thermal simulator, like the FPS in [5], can be used to achieve the best efficiency. A nonconformal discretization technique is then proposed, which discretizes the subdomains of heat spreader and heat sink with much coarser mesh, and thereby substantially reduces the solution time for them. With several 2-D and 3-D IC test cases, we demonstrate that the proposed DDM can be over 16× faster than other fast PCG algorithms [5], [8], [9], while using much less memory. And, the nonconformal discretization causes negligible error and is able to capture the hot spot accurately. The proposed method is capable of thermal simulation with very high discretization resolution, which is favorable for 3-D IC.

Although DDM has been applied to circuit simulation problems [10], [11], its merit has not been realized in existing works on IC thermal simulation. The aim of this letter is to present specific DDM techniques for the thermal simulation of realistic IC models.

## II. PRELIMINARIES

In the pyramid-shape IC model (Fig. 1), the IC region mainly includes two parts: silicon substrate and the interconnect layer. The former is made of silicon, while the latter is filled with metal and dielectrics. Considering the percentage of metal volume, the interconnect layer can be approximated by a homogeneous layer with an effective thermal conductivity. This results in a model that is composed of homogeneous layers, which considered in this letter and is typical in most previous works. Without loss of generality, in the following discussion, we ignore the heat dissipation through packaging and board, for that being a minor dissipation channel, although it can be easily handled by the proposed methods.

The steady-state IC thermal analysis involves solving the temperature distribution  $T(x, y, z)$  from the 3-D Poisson equation

$$k \cdot \left( \frac{\partial^2 T(x, y, z)}{\partial x^2} + \frac{\partial^2 T(x, y, z)}{\partial y^2} + \frac{\partial^2 T(x, y, z)}{\partial z^2} \right) = -p(x, y, z) \quad (1)$$

where  $k$  is thermal conductivity and  $p(x, y, z)$  is the internal heat generation density at point  $(x, y, z)$ . The heat generation is due to the device modules, or function blocks located around the top surface of the silicon die. Equation (1) holds for a homogeneous region. For a problem with multiple homogeneous regions, the equation of continuous heat flux

should be applied at the interface between two regions. The heat flux density satisfies:  $\vec{q}(x, y, z) = -k \cdot \nabla T(x, y, z)$ , where  $\nabla$  is the gradient operator.

The finite volume method (FVM) is conventionally used for the 3-D thermal simulation, where the domain is discretized into cells and each cell is associated with a temperature [1], [2]. Similar to simulating the steady-state electric current field with electric resistors, we can define and calculate thermal resistor to model the heat flow through the interface between any two adjacent cells. The heat source resembles the current source in electric circuit. Thus, an equivalent circuit with resistors and current source is generated. With the nodal analysis approach, a linear equation system

$$\mathbf{A}\mathbf{T} = \mathbf{f} \quad (2)$$

is formed, where  $\mathbf{A}$  is a sparse symmetric positive definite matrix,  $\mathbf{f}$  is the vector of current sources, and  $\mathbf{T}$  is the temperature vector. The temperature profile can be obtained by solving (2) with direct or iterative equation solvers [8], [12].

There are different boundary conditions for the simulation domain. At the bottom surface of heat sink, a convective condition should be set, which models the heat transfer mechanism at the interface of heat sink and air

$$k \frac{\partial T}{\partial \vec{n}} + h(T - T_{amb}) = 0 \quad (3)$$

where  $\vec{n}$  is the out normal direction of the boundary,  $T_{amb}$  is the ambient temperature, and  $h$  is the convective coefficient. The partial derivative in (3) can be approximated with finite difference formula. By defining  $R_{amb} = 1/(h \cdot h_x \cdot h_y)$ , where  $h_x$  and  $h_y$  are the edge sizes of the cell along the  $x$ -axis and  $y$ -axis, respectively, we can model the effect of convective boundary with the thermal resistors of value  $R_{amb}$ . They connect the nodes of boundary cells to a virtual node with temperature  $T_{amb}$ . For other boundaries of the domain, the adiabatic condition is usually assumed. It is naturally modeled by the equivalent circuit.

An FPS algorithm for thermal analysis was presented in [5]. It is equivalent to the Green's function based fast algorithms [3], and is highly efficient for the rectangular IC model. The FPS requires the uniform discretization along the  $x$ -axis and  $y$ -axis. This gives matrix  $\mathbf{A}$  in (2) a distinct block structure, where each block has analytical eigen-decomposition. By multiplying the matrix consisting of eigenvectors, the problem is transformed for solving a series of smaller equation systems. Using the same transformation again, we only need to solve the linear systems with the order- $n_z$  tridiagonal coefficient matrix, where  $n_z$  is the number of grids along the  $z$ -axis. And, the product with the eigenvector matrix can be realized with the fast Fourier transform (FFT) algorithm without the loss of accuracy. The FPS algorithm has  $O(n \log n)$  time complexity and  $O(n)$  space complexity, where  $n$  is the number of discretized unknowns [5].

### III. DOMAIN DECOMPOSITION METHOD FOR THERMAL SIMULATION WITH IRREGULAR GEOMETRY MODEL

In this section, we first present the nonoverlapping DDM for the thermal simulation of pyramid-shape IC model. Then, a nonconformal discretization technique is proposed.

#### A. Nonoverlapping DDM and Iterative Schemes

For the IC thermal model shown in Fig. 1, our idea is to divide the whole domain into three rectangular subdomains representing the chip, heat spreader, and heat sink,

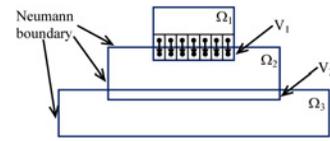


Fig. 2. Boundary settings in the DDM for pyramid-shape IC model.

respectively. Therefore, each subdomain can be simulated with existing fast algorithms. There are two kinds of DDMs: nonoverlapping DDM, and overlapping DDM (also called the Schwarz alternating method) [13]. The overlapping DDM includes subdomains with larger size, and thus takes more time to solve subdomain problems. Furthermore, the overlapping DDM usually sets the Dirichlet condition at the fictitious boundaries of subdomains, and results in a mixture of Dirichlet boundary and Neumann boundary on a surface of the rectangular subdomain. This causes difficulty for certain fast algorithms, including FPS [5]. For the above reasons, we adopt the nonoverlapping DDM, which makes each subdomain the smallest size. The Dirichlet and Neumann boundaries are used alternately for two neighboring subdomains during the iteration.

Since the volume discretization is employed, in the nonoverlapping DDM, two neighboring subdomains share the same layer of volume cells (see the adjacent-to-interface cells in subdomain  $\Omega_2$  in Fig. 2). The solution of  $\Omega_2$  provides the Dirichlet boundary condition for  $\Omega_1$ . On the other hand, the heat flow across the top surface of  $\Omega_2$  (the downward arrows in Fig. 2) derived from the solution of  $\Omega_1$  constitutes the Neumann boundary condition for  $\Omega_2$ . A similar boundary setting applies to the interface between  $\Omega_2$  and  $\Omega_3$ . Note that the Dirichlet and Neumann conditions can be regarded as a set of voltage sources and current sources, respectively, in the equivalent circuit model.

There are various iteration schemes for the DDM. Considering the three-subdomain problem in Fig. 2, we can have the following calculation orders for an iteration step.

- 1) **Top-to-bottom order:** the subdomains are solved in the order of  $\Omega_1$ ,  $\Omega_2$ , and  $\Omega_3$ . To start with, an initial temperatures should be assumed on the bottom surfaces of  $\Omega_1$  and  $\Omega_2$ .
- 2) **Bottom-to-top order:** the subdomains are solved in the order of  $\Omega_3$ ,  $\Omega_2$ , and  $\Omega_1$ . At the beginning, initial heat flows should be assumed on the top surfaces of  $\Omega_2$  and  $\Omega_3$ .
- 3) **Middle-to-end order:** the subdomain  $\Omega_2$  is firstly solved, and then  $\Omega_1$  and  $\Omega_3$  are solved. At the beginning, initial temperatures and heat flows should be set on the bottom and top surfaces of  $\Omega_2$ , respectively.
- 4) **End-to-middle order:** it is the inverse of the middle-to-end order. At the beginning, the subdomain  $\Omega_1$  is solved with initial temperatures at the bottom surface, and  $\Omega_3$  is solved with initial heat flows at its top surface.
- 5) **Nested two-subdomain order:** the whole domain is divided into two subdomains. If a subdomain is of irregular shape, it is solved recursively with the two-subdomain DDM. At the beginning, the Dirichlet conditions are assumed.

DDM with a nested two-subdomain order is usually more reliable, and easy to converge [13]. However, it forms several levels of loops and the subdomains in the inner loops are solved for much more times than the out-loop subdomain. This causes more total computing time even though the

---

**Algorithm 1** Relaxed Nonoverlapping DDM With Top-to-Bottom Order for the Thermal Simulation of Pyramid-Shape IC Model
 

---

1. Discretize the subdomains of IC ( $\Omega_1$ ), heat spreader ( $\Omega_2$ ) and heat sink ( $\Omega_3$ ) separately;
  2. Assign the initial temperature  $T_{V1}^{(0)}$  at the bottom of  $\Omega_1$ , and the initial temperature  $T_{V2}^{(0)}$  at the bottom of  $\Omega_2$ ;
  3. Set the relaxation factor  $\omega$ ;  $i := 0$ ;
  4. **Repeat**
  5. Solve the temperature profile of rectangular subdomain  $\Omega_1$ , considering the boundary value  $T_{V1}^{(i)}$  at the bottom boundary;
  6. Calculate heat flow  $q_{V1}^{(i+1)}$  across the top surface of  $\Omega_2$ ;
  7. Solve the temperature profile of rectangular subdomain  $\Omega_2$ , considering the boundary value  $T_{V2}^{(i)}$  on bottom boundary and  $q_{V1}^{(i+1)}$  on top surface;
  8. Calculate heat flow  $q_{V2}^{(i+1)}$  across the top surface of  $\Omega_3$ ;
  9. Solve the temperature profile of rectangular subdomain  $\Omega_3$ , considering the boundary value  $q_{V2}^{(i+1)}$  on top surface;
  10. Suppose the calculated temperature at the bottom boundary of  $\Omega_1$  is  $\tilde{T}_{V1}^{(i+1)}$ ;  $T_{V1}^{(i+1)} := T_{V1}^{(i)} + \omega(\tilde{T}_{V1}^{(i+1)} - T_{V1}^{(i)})$ ;
  11. Suppose the calculated temperature at the bottom boundary of  $\Omega_2$  is  $\tilde{T}_{V2}^{(i+1)}$ ;  $T_{V2}^{(i+1)} := T_{V2}^{(i)} + \omega(\tilde{T}_{V2}^{(i+1)} - T_{V2}^{(i)})$ ;
  12.  $i := i + 1$ ;
  13. **Until** the stopping criterion is met.
- 

outer-loop iteration number may be decreased. In contrast, with the other orders every subdomain is solved for the same number of times. Our experiments reveal that the nested two-subdomain order has no advantage over the other orders, with the longest runtime. So, below we only consider the first four orders for the DDM iteration.

The convergence rate of DDM can be accelerated by the relaxed iterative scheme. Suppose the temperature at the top surface of  $\Omega_2$  got by solving  $\Omega_2$  is  $\tilde{T}_{V1}^{(i+1)}$ . The relaxed iterative scheme calculates the value for the  $(i+1)$ th iteration step with

$$T_{V1}^{(i+1)} = T_{V1}^{(i)} + \omega(\tilde{T}_{V1}^{(i+1)} - T_{V1}^{(i)}) \quad (4)$$

where  $\omega$  is a relaxation factor, and superscript  $(i)$  indicates the quantities in the  $i$ th iteration step. The value of  $\omega$  can be chosen in  $[0, 1]$ , but its optimal value may vary. Algorithm 1 describes the relaxed DDM with the top-to-bottom order scheme for the thermal simulation of the pyramid-shaped IC model.

In Algorithm 1, the stopping criterion should evaluate the difference of nonrelaxed quantities, like  $\tilde{T}_{V1}^{(i+1)} - T_{V1}^{(i)}$ , to remove the influence of  $\omega$  on convergence. Algorithms of the DDMs with other iteration orders can be similarly derived.

### B. Nonconformal Discretization and Discussion

Since the subdomains are solved separately in DDM, we can impose nonconformal discretization grids on different subdomains. It is based on the observation that the temperature of heat sink and heat spreader is of less importance than that of IC. Moreover, due to the absence of heat source, the temperature varies more smoothly in the two subdomains. Therefore, much coarser grid can be used in them. In practice, we set the discretization step sizes along the  $x$ -axis and  $y$ -axis for the heat spreader to be several times larger than those

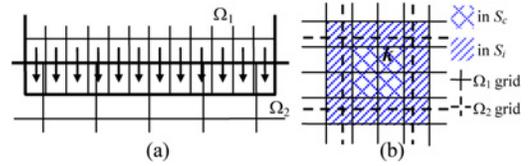


Fig. 3. (a) Side view and (b) top view of subdomain interface with nonconformal discretization grids (arrows represent heat flow).

for the IC region, and the step sizes for the heat sink several times larger than those for the heat spreader. In the DDM with nonconformal discretization grids, a linear interpolation technique is used to convert the values of temperature and heat flow across the interface of subdomains (Fig. 3). Take the  $k$ th grid cell in  $\Omega_2$  as an example; it overlaps several cells in  $\Omega_1$ . We convert the heat flow into cells in  $\Omega_1$  to the heat flow into cell  $k$ . Suppose cell  $k$  contains the cells in  $\Omega_1$  with numbers form a set  $S_c$ , while the cells in  $\Omega_1$  intersecting cell  $k$  has numbers in set  $S_i$  [Fig. 3(b)]. Then, the converted heat flow can be calculated as

$$q'_{V1,k} = \sum_{j \in S_c} q_{V1,j} + \sum_{j \in S_i} r_j \cdot q_{V1,j} \quad (5)$$

where  $q_{V1,j}$  is the heat flow into cell  $j$  in  $\Omega_1$ , and  $r_j$  denotes the percentage of the cell  $j$ 's area that falls in cell  $k$  of  $\Omega_2$ . Similar formula is used to convert the temperature from cells in  $\Omega_2$  to cells in  $\Omega_1$ . With the nonconformal discretization scheme, the number of unknowns in the heat spreader and heat sink will be substantially reduced, even to a number less than that in the IC region. Thus, for the high-resolution thermal simulation of IC, the computing expense for each iteration step of DDM becomes mainly dependent on the number of unknowns in IC region.

The initial value of the DDM iteration may affect the convergence rate of DDM. Our choice is to set the ambient temperature as the initial value of temperature. The initial heat flow that is needed for some DDM iteration schemes can be set to a constant with a reasonable order of magnitude. Our experiments show that the convergence rate of DDM is little affected by the detailed values in these initial conditions. Another approach is to set initial condition by the solution obtained through solving the whole domain with a coarse discretization grid. However, this results in an overhead in computational time. Except for the problem with very dense discretization, this strategy cannot be beneficial to the total computational time.

If each subdomain is solved with FPS, we can compare the DDM with the FPS-PCG algorithm [5]. The major computation of each DDM iteration step is solving the three subdomains with FPS. On the contrary, FPS-PCG solves the whole domain (actually a larger and modified domain) with FPS in each iteration step of PCG. So, the computational expense of DDM is cheaper for each iteration. If the number of iterations in DDM is not larger than that in FPS-PCG, the former will be more efficient than the latter. As for the space complexity, the DDM is bounded by the unknowns for the largest subdomain. For FPS-PCG, the major memory cost is for storing several vectors with the same size as the unknowns in the whole domain. So, the DDM has lower space complexity than FPS-PCG and is able to simulate larger problem with the same memory footprint.

TABLE I  
THICKNESS AND MATERIAL PARAMETERS

Thickness of	Case 1 ( $\mu\text{m}$ )	Case 2 ( $\mu\text{m}$ )	Thermal conductivity (W/(K·cm))	Thermal conduc- tivity of memory die in Case 3
Interconnect layer	10	50	0.3	
Silicon substrate	480	450	1.25	1.8 W/(K·cm)
Thermal interface	200	200	0.3	Heat convective coefficient $h$
Heat spreader	1500	1500	3.95	
Heat sink	5000	5000	3.95	

#### IV. NUMERICAL RESULTS

To validate the efficiency of the proposed DDM techniques, three chip structures with different power profiles are tested on the following.

- 1) **Case 1:** A 2-D chip imitating the POWER6 micro-processor [14]. The transverse dimensions of the die, spreader, and sink are 1.6 cm  $\times$  2 cm, 3 cm  $\times$  3 cm, and 7 cm  $\times$  7 cm, respectively. The thickness and material parameters are listed in Table I. The power map is similar to that in [5], with total power of 175 W.
- 2) **Case 2:** A four-core 2-D chip artificially generated, which is the Testcase no. 2 in [5]. Except that the die is of 1 cm  $\times$  1 cm, the transverse dimensions are the same as Case 1. The thickness and material parameters are listed in Table I. The power map (totally 176 W) models a scenario with one core idle, one core with peak load, and two others with median loads.
- 3) **Case 3:** A high-performance 3-D chip with three device dies. The processor die closest to heat sink is the same as that in Case 1. The other two dies are thinned static random-access memory (SRAM), with power of 10.6 W each, and similar power density and profile as [15]. The memory die uses thinned silicon with 50  $\mu\text{m}$  thickness.

The DDM algorithms have been implemented in a MATLAB program *ddmThermal*, which invokes FPS coded in C. The fastest Fourier transform in the West (FFTW) [16] is used in the FPS to perform FFT. The problems of thermal simulation (2) are also solved with the MATLAB “\” operator [12], the PCG with incomplete Cholesky factorization with drop tolerance (ICT-PCG) [8], and PowerRush with the algebraic multigrid PCG (AMG-PCG) algorithm [9]. For all DDM iteration schemes,  $\left\| \tilde{\mathbf{T}}_{V1}^{(i)} - \mathbf{T}_{V1}^{(i-1)} \right\|_{\infty} < 2 \times 10^{-4}$  is set as the convergence criterion. This guarantees same result accuracy as the iterative equation solvers.

All experiments are carried out with serial computing on a PC with 2.70 GHz dual-core Pentium CPU, 6-GB memory.

##### A. Experiments With Conformal Discretization Grids

In this subsection, the results regarding the conformal discretization grids among all subdomains are presented. We first test different DDM iteration orders. The number of iterations ranges from 10 to 12 for various iteration orders and cases. Usually, the top-to-bottom DDM order has the fewest iteration steps. We then test the influence of the relaxation factor  $\omega$ . We find out that the number of iteration varies between 8 and 13. As far as is known, there is no rule about how to choose the best  $\omega$ . So, we simply use  $\omega=0.9$  and the top-to-bottom order (Algorithm 1) in the following experiments.

For each case, with different resolution of discretization, we obtained multiple equation systems (2). For each of them, the DDM algorithm and other equation solvers are used to perform thermal simulation. In Table II, the computational results of

TABLE II  
COMPARISON OF MATLAB “\,” ICT-PCG, AMG-PCG, AND DDM

Matrix	n	“\”		ICT-PCG		AMG-PCG		DDM	
		Time(s)	Iter.	Time(s)	Iter.	Time(s)	Mem.	Time(s)	Mem.
M1-1	1.45e5	<b>35.4</b>	75	<b>16.7</b>	16	<b>1.15</b>	14MB	<b>1.20</b>	< 8MB
M1-2	1.51e5	<b>38.9</b>	75	<b>16.8</b>	13	<b>1.01</b>	33MB	<b>1.28</b>	< 8MB
M1-3	3.42e5	<b>331.5</b>	110	<b>43.1</b>	14	<b>2.63</b>	72MB	<b>2.48</b>	< 8MB
M1-4	5.47e6	--	--	--	15	<b>51.1</b>	1.1GB	<b>32.3</b>	80MB
M1-5	1.51e7	--	--	--	14	<b>142.6</b>	3.0GB	<b>90.7</b>	218MB
M1-6	3.42e7	--	--	--	--	--	--	<b>207.2</b>	494MB
M2-1	8.97e4	<b>18.4</b>	66	<b>6.88</b>	12	<b>0.52</b>	14MB	<b>0.82</b>	< 10MB
M2-2	1.40e5	<b>43.4</b>	78	<b>16.7</b>	12	<b>0.89</b>	31MB	<b>1.13</b>	< 10MB
M3-3	3.59e5	<b>359.3</b>	103	<b>204.9</b>	12	<b>2.68</b>	78MB	<b>2.21</b>	< 10MB
M2-4	3.51e6	--	--	--	12	<b>26.63</b>	718MB	<b>20.5</b>	55MB
M2-5	1.40e7	--	--	--	13	<b>124.6</b>	2.8GB	<b>80.1</b>	217MB
M2-6	3.31e7	--	--	--	--	--	--	<b>185.2</b>	493MB
M3-1	1.50e5	<b>39.6</b>	77	<b>16.8</b>	13	<b>1.08</b>	18MB	<b>1.28</b>	< 8MB
M3-2	1.55e5	<b>40.8</b>	77	<b>16.9</b>	13	<b>1.11</b>	37MB	<b>1.32</b>	< 8MB
M3-3	3.46e5	<b>493.4</b>	112	<b>33.3</b>	14	<b>2.81</b>	78MB	<b>2.61</b>	< 8MB
M3-4	5.54e6	--	--	--	14	<b>49.1</b>	1.1GB	<b>33.5</b>	80MB
M3-5	1.55e7	--	--	--	15	<b>160.8</b>	3.1GB	<b>92.4</b>	218MB
M3-6	3.46e7	--	--	--	--	--	--	<b>209.1</b>	494MB

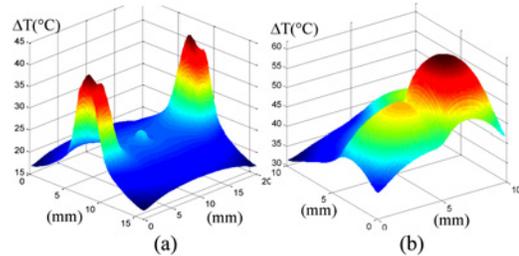


Fig. 4. On-chip profiles of temperature rise. (a) Case 1. (b) Case 2.

six equations per test case are listed. For larger matrices, the results of MATLAB “\,” ICT-PCG, and AMG-PCG are not available, due to over 12 hours runtime or memory limitation. For the cases, the DDM converges with eight or nine steps. From the table, we can see that the DDM is much faster than ICT-PCG, and about 1.6 $\times$  faster than AMG-PCG for large cases. The memory usage of DDM is only 1/10 of that by AMG-PCG. Compared with the results of the direct solver MATLAB “\,” the maximum and average errors of DDM on the chip temperature are 0.003  $^{\circ}\text{C}$  and 0.002  $^{\circ}\text{C}$ , respectively, for Case 1. And, the errors are also less than 0.01  $^{\circ}\text{C}$  for the other cases. Although there are more variables in the IC region of Case 3, the memory cost of DDM is the same as for Case 1. The reason is that the largest subdomain is the heat sink due to the conformal discretization.

Fig. 4 shows the chip temperature profiles for Cases 1 and 2. The results are obtained from the finest resolution in Table II, both with the step size of 100  $\mu\text{m}$  along the  $x$ -axis and  $y$ -axis. The maximum rise in temperature are 41.9  $^{\circ}\text{C}$  and 61.1  $^{\circ}\text{C}$ , respectively. For Case 3, the maximum temperature rise is 42.5  $^{\circ}\text{C}$ . Cases 2 and 3 have higher temperature than Case 1, because they have larger power density. Since Case 2 is the same as Testcase No. 2 in [5], we can compare the runtimes of DDM, AMG-PCG, and FPS-PCG. The trend curves in Fig. 5 reveal that the DDM is over 2 $\times$  faster than FPS-PCG.

##### B. Experiments With Nonconformal Discretization Grids

With the nonconformal discretization technique, the runtime of DDM can be accelerated further. We set the step size for the heat spreader to be 2.5 $\times$  of that for IC, and the step size for the heat sink 4 $\times$  of that for heat spreader. Using

TABLE III

COMPARISON OF DDMs WITH CONFORMAL AND NONCONFORMAL GRIDS

Matrix	Conformal grid			Non-conformal grid				
	n	Time(s)	T <sub>max</sub> (°C)	n	Time(s)	T <sub>max</sub> (°C)	Err <sub>max</sub> (°C)	Speedup
M1-4	5.47e6	32.3	61.89	5.73e5	3.86	61.90	0.11	<b>8.4</b>
M1-5	1.51e7	90.7	61.97	1.65e6	8.98	61.95	0.03	<b>10.1</b>
M1-6	3.42e7	203.2	61.88	1.99e6	11.2	61.85	0.04	<b>18</b>
M1-10	5.41e7	310.1	61.85	3.22e6	18.5	61.84	0.02	<b>17</b>

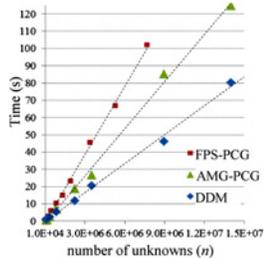
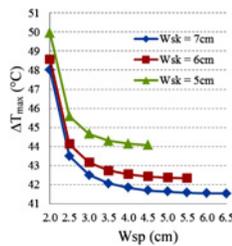


Fig. 5. Runtime of FPS-PCG, AMG-PCG, and DDM for Case 2.

Fig. 6. Trends of maximum temperature rise in Case 3, with varied sizes of heat spreader ( $W_{sp}$ ) and heat sink ( $W_{sk}$ ).

Case 1 with different discretization resolutions, we compare the nonconformal grid and the conformal grid in Table III. In addition to several matrices from Table II, a larger matrix M1-10 is also tested. In Table III,  $T_{max}$  means the maximum temperature rise, and  $Err_{max}$  means the maximum error of node temperatures in the IC region induced by the nonconformal discretization. We see that the error of temperature is within 0.13 °C (i.e., 0.3%) for all nodes, and within 0.02 °C for the hot spot. It is also found out that with the nonconformal discretization the hot spot is accurately located.

From Table III, it is revealed that the nonconformal discretization brings about 10× speedup to the DDM-based thermal simulation. Note that the nonconformal discretization hardly affects the convergence rate of DDM; for these cases, the DDM always converges in eight iteration steps.

As for the memory usage, the DDM with nonconformal grid consumes 6.3 MB, 24 MB, 24 MB, and 37 MB memory for the four matrices. Compared with Table II, more than 10× reduction is observed. Similar speedup brought by the nonconformal discretization can be observed with the other test cases. And, the induced temperature error in the IC region is within 0.5%. Therefore, with negligible error the proposed DDM algorithm is able to achieve 16× speedup and 100× memory reduction, as compared with the AMG-PCG method.

With the nonconformal DDM techniques, high-resolution thermal simulation can be performed easily. For example, we have made a fine-granularity mesh on Case 3 and simulated it with the proposed DDM solver, where  $1.05 \times 10^7$  unknown nodes are included in the IC region. The nonconformal DDM solver costs only 72 s (in nine iteration steps) and 194 MB

memory. We also change the widths of heat spreader and heat sink in Case 3, and study the trends of the maximum temperature rise with varied sizes of heat spreader and sink. The results are plotted in Fig. 6. For the total 24 configurations, the simulation procedure costs only 7 minutes.

## V. CONCLUSION

In this letter, the DDM techniques are presented for the thermal simulation with chip, heat spreader, and heat sink regions. Suitably set with subdomain boundaries and iteration schemes, the relaxed DDM algorithm converges quickly. For only the temperature in chip region desired, a nonconformal discretization strategy with interpolation at subdomain interface is proposed to reduce the number of unknowns in heat sink components, which achieves about 10× speedup with only 0.5% or less temperature error. Experiments with test cases of 2-D and 3-D ICs demonstrate the advantages of the proposed DDM algorithm over the FPS-PCG [5] and AMG-PCG [9] algorithms, and its efficiency for the thermal simulation with very high discretization resolution.

## ACKNOWLEDGMENT

The authors would like to thank Dr. J. Chen and Mr. J. Yang, Tsinghua University, Beijing, China, for helpful discussions.

## REFERENCES

- [1] S.-C. Lin and K. Banerjee, "Thermal challenges of 3D ICs," in *Wafer-Level 3D ICs Process Technology*. Berlin, Germany: Springer, 2008, pp. 307–332.
- [2] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC thermal simulation and modeling via efficient multigrid-based approaches," *IEEE Trans. Comput.-Aided Design*, vol. 25, no. 9, pp. 1763–1776, Sep. 2006.
- [3] Y. Zhan and S. S. Sapatnekar, "High efficiency Green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Design*, vol. 26, no. 9, pp. 1661–1675, Sep. 2007.
- [4] V. M. Heriz, J. Park, T. Kemper, S. Kang, and A. Shakouri, "Method of images for the fast calculation of temperature distributions in packaged VLSI chips," in *Proc. THERMINIC*, 2007, pp. 18–25.
- [5] H. Qian, S. S. Sapatnekar, and E. Kursun, "Fast Poisson solvers for thermal analysis," *ACM Trans. Design Autom. Electron. Syst.*, vol. 17, no. 3, pp. 32:1–32:23, Jun. 2012.
- [6] D. Oh, C. C. P. Chen, and Y. H. Hu, "Efficient thermal simulation for 3-D IC with thermal through-silicon vias," *IEEE Trans. Computer-Aided Design*, vol. 31, no. 11, pp. 1767–1771, Nov. 2012.
- [7] C.-W. Pan, Y.-M. Lee, P.-Y. Huang, C.-P. Yang, C.-T. Lin, C.-H. Lee, Y.-F. Chou, and D.-M. Kwai, "I-LUTSim: An iterative look-up table based thermal simulator for 3-D ICs," in *Proc. ASP-DAC*, 2013, pp. 151–156.
- [8] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2003.
- [9] J. Yang, Z. Li, Y. Cai, and Q. Zhou, "PowerRush: A linear simulator for power grid," in *Proc. ICCAD*, 2011, pp. 482–487.
- [10] K. Sun, Q. Zhou, K. Mohanram, and D. C. Sorensen, "Parallel domain decomposition for simulation of large-scale power grids," in *Proc. ICCAD*, 2007, pp. 54–59.
- [11] H. Peng and C.-K. Cheng, "Parallel transistor level circuit simulation using domain decomposition methods," in *Proc. ASP-DAC*, 2009, pp. 397–402.
- [12] T. A. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2006.
- [13] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*. Berlin, Germany: Springer, 2005.
- [14] H. Q. Le, W. J. Starke, J. S. Fields, F. P. O'Connell, D. Q. Nguyen, B. J. Ronchetti, W. M. Sauer, E. M. Schwarz, and M. T. Vaden, "IBM POWER6 microarchitecture," *IBM J. Res. Devel.*, vol. 51, no. 6, pp. 639–662, 2007.
- [15] S. Borkar, "3D integration for energy efficient system design," in *Proc. DAC*, Jun. 2011, pp. 214–219.
- [16] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.