3-D Thermal-ADI: A Linear-Time Chip Level Transient Thermal Simulator

Ting-Yuan Wang and Charlie Chung-Ping Chen

Abstract—**Recent study shows that the nonuniform thermal distribution not only has an impact on the substrate but also interconnects. Hence, three–dimensional (3-D) thermal analysis is crucial to analyze these effects. In this paper, the authors present and develop an efficient 3-D transient thermal simulator based on the alternating direction implicit (ADI) method for temperature estimation in a 3-D environment. Their simulator, 3D Thermal-ADI, not only has a linear runtime and memory requirement, but also is unconditionally stable. Detailed analysis of the 3-D nonhomogeneous cases and boundary conditions for on-chip VLSI applications are introduced and presented. Extensive experimental results show that our algorithm is not only orders of magnitude faster than the traditional thermal simulation algorithms but also highly accurate and memory efficient. The temperature profile of steady state can also be reached in several iterations. This software will be released via the web for public usage.**

Index Terms—**ADI, design automation, finite difference methods, thermal simulation, temperature.**

I. Introduction

Because of the rapid increase of power and packaging densities, thermal issues have become important factors of the reliability and performance concerns for advanced very large scale integration (VLSI) design and manufacturing. A comprehensive analysis of the thermal effects in high-performance VLSI has been discussed recently [1]–[4]. Management of thermal issues is becoming a key factor to success for the next-generation high-performance VLSI design.

In general, thermal effects are caused by the power distribution and dissipation. The primary power consumption in chips is associated with devices. However, the thermal effects in interconnects are becoming more serious even though the Joule heating only contributes a small part of the chip power consumption. The trend of temperature distribution along the vertical distance from upper surface of silicon substrate to top metal level is shown in Fig. 1 [5]. This is due to the fact that the scaling trends make the thermal effects worse because of the increase of interconnect levels, current density, and thermal coupling as well as the introducing of low-k dielectrics.

High temperature not only causes timing failures for both transistors and interconnects but also degrades chip reliability. For example, the temperature-induced logic fault occurs in a 10-bit adder because a large temperature gradient causes the arrival time of an input signal at the tenth bit to become slower than expected [6].

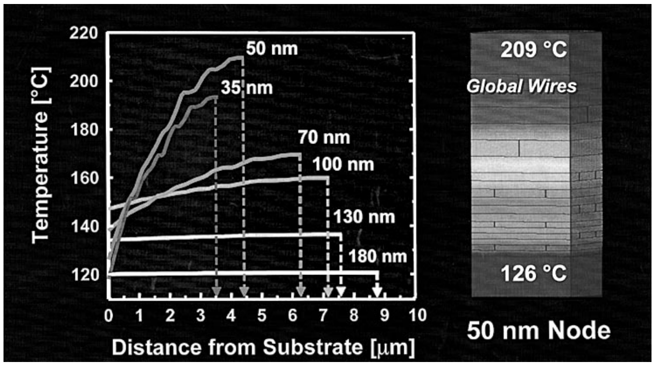


Fig. 1. The trend of maximum temperature distribution along vertical distance from upper surface of silicon substrate to top metal level [5].

Therefore, how to effectively analyze the chip-level three–dimensional (3-D) thermal distribution and hot-spot locations is important. There are some issues making the problem hard to deal with. The uniform heat distribution does not guarantee the uniform temperature profile due to the complex 3-D nature of heat spreading. This addresses another 3-D thermal issue. If the thermal analysis of interconnects is based on single isolated lines, this approach cannot solve the highly integrated VLSI chip. The reason is that the interconnects form a complicated 3-D array. The total heating in the interconnects could be more severe due to self-heating and thermal coupling.

Several approaches have been proposed to perform thermal analysis. The finite-difference method with equivalent RC model has been presented [6], [7]. However, due to the complexity of solving the large scale matrix, the existing direct matrix-solving algorithms suffer superlinear runtime and memory usage for large scale problems. Reference [8] presented a full-chip thermal analysis, but simplified the problem with function blocks. A two-dimensional (2-D) full-chip thermal simulation was also presented [9].

In this paper, we propose an efficient transient thermal simulator using the alternating direction implicit (ADI) method [10] to simulate the 3-D temperature profile. Basically, the ADI method is an alternative solution method which instead of solving the 3-D problem, solves a succession of three one-dimensional problems. Our simulator, 3D Thermal-ADI, is not only unconditionally stable but also has a linear runtime and a linear memory requirement. Extensive experimental results show that our algorithm is not only orders of magnitude faster than the traditional thermal simulation algorithms but

Manuscript received January 17, 2002; revised May 22, 2002. This work was supported in part by the National Science Foundation under Grant CCR-0093309, by Intel Corp., and by Faraday, Inc. This paper was recommended by Associate Editor Z. Yu.

The authors are with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA (e-mail: wangt@cae.wisc.edu and chen@engr.wisc.edu).

Digital Object Identifier 10.1109/TCAD.2002.804385

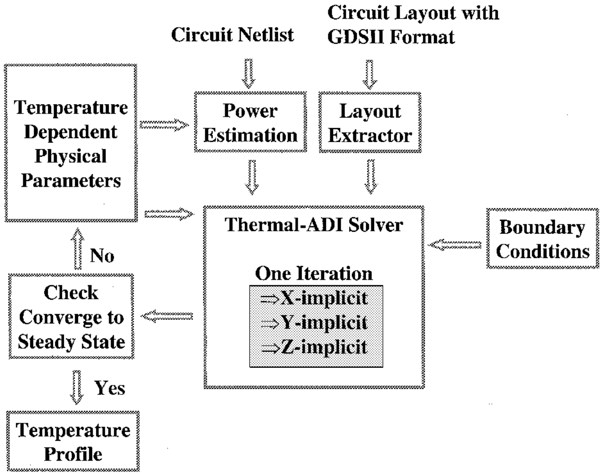


Fig. 2. The flowchart of the 3D Thermal-ADI simulator.

also highly accurate and memory efficient. The temperature profile of steady state can also be reached in several iterations.

The remainder of the paper is organized as follows. Overview of the simulator is presented in Section II. Section III discusses the problem formulation by finite-difference method. Section IV deals with the ADI method. The implementation and experimental results are presented in Section V, followed by the conclusion in Section VI. The proof of unconditional stability of the ADI method is in the Appendix.

II. Overview of the Simulator

3D Thermal-ADI is a linear time chip-level transient thermal simulator. It simulates the temperature profile by the finite-difference method based on the ADI method. The main procedure is described as follows and shown in Fig. 2.

The geometric information and locations of the interconnects and transistors are given. The simplified problem which has the geometry and locations of the function blocks can also be handled. These information may be extracted from the GDSII file or other sources. The corresponding power densities are given from the power estimation tools [11], [12]. The package information and chip size are given in order to apply the boundary conditions. The temperature sensitive physical parameters as well as the technology information are given. The system initializes the computation by combining all the information and begins the calculation by the ADI-based solver.

In order to converge to the thermal steady state, at least several transient iterations are needed. Each transient iteration repeats the following procedure. The system checks whether the temperature profile converges to steady state or not. If the results do not converge, the physical parameters and the power are updated by the current temperature solutions. Then the next iteration will be calculated according to the new power and physical values. If the results do converge, the temperature profile approaches a steady state. Some detailed discussions are described as follows.

A. Kernel of 3D Thermal-ADI Solver

The temperature distribution in a chip is governed by the following partial differential equation of heat conduction [13]:

$$\rho C_p \frac{\partial T(\vec{r},t)}{\partial t} = \bigtriangledown \cdot [\kappa(\vec{r},T) \bigtriangledown T(\vec{r},t)] + g(\vec{r},t) \qquad (1)$$

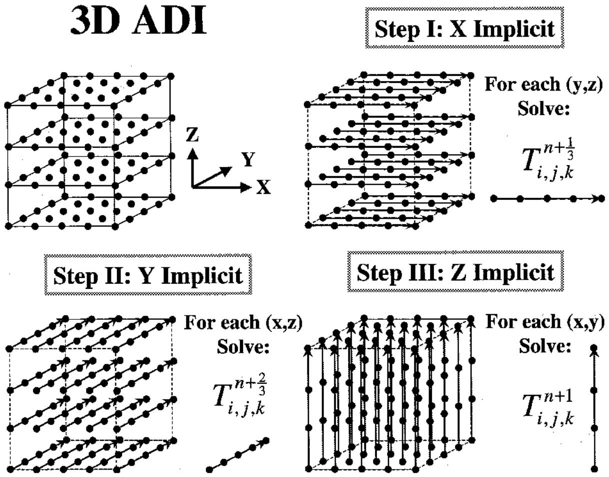


Fig. 3. ADI method.

subject to the thermal boundary conditions

$$\kappa(\vec{r},T)\frac{\partial T(\vec{r},t)}{\partial n_i} + h_i T(\vec{r},t) = f_i(\vec{r}_{s_i},t) \qquad (2)$$

where T is the time dependent temperature at any point, ρ is the density of the material, C_p is the specific heat, κ is the thermal conductivity as a function of temperature and position, g is the heat energy generation rate, h_i is the heat transfer coefficient on the boundary surface of the chip, $f_i(\vec{r}_{s_i},t)$ is an arbitrary function on the boundary surface s_i, and $\partial/\partial n_i$ is the differentiation along the outward direction normal to the boundary surface s_i.

The physical meaning of (1) is described as follows. Consider the energy-balance condition for a small control volume. The rate of energy stored in a control volume causing the temperature increase is $\int \rho C_p(\partial T/\partial t)dV$. Suppose that the rate of heat conduction through surface $d\vec{A}$ is $q = \kappa d\vec{A} \cdot \bigtriangledown T$. Then the rate of heat flow entering the control volume is $\int q = \int \kappa \bigtriangledown T \cdot d\vec{A} = \int \bigtriangledown \cdot [\kappa \bigtriangledown T]\, dV$. The power generated in a control volume is $\int g dV$. For the convection boundary conditions, the function f_i in (2) is $f_i = h_i T_\infty$, where T_∞ is the ambient temperature.

Note that the thermal time constant of the heat conduction is much larger than the circuit clock period, which implies that the temperature variation caused by transient currents is small. Therefore, we are able to use the average power dissipation in calculating the heat generation rate.

First, the finite-difference method is used to solve (1), and efforts are made to keep the formulas with second-order accuracy in time and space. Since there is not only one material in the chip, the homogeneous and nonhomogeneous cases are also discussed. However, the computational inefficiency due to the large size of the problem and the complicated situation of the nonhomogeneous cases requires long runtime and large memory usage.

Therefore, we introduce the ADI method to alleviate the problems. By the ADI method, only the variables in one direction are implicit in each step, thus the matrix for solving the ADI method at each direction is tridiagonal. This implies that no matrix solving is needed, and runtime for solving the tridiagonal matrix is linear. The detailed method and the formulas we derived are discussed in Section IV.

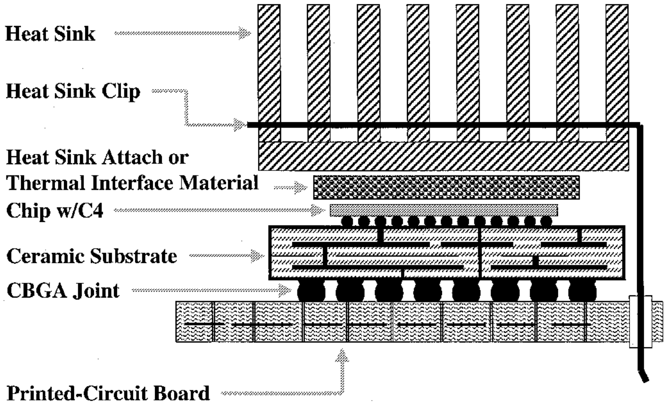


Fig. 4. The cross-sectional view of the PowerPC with C4/CBGA package and heat sink.

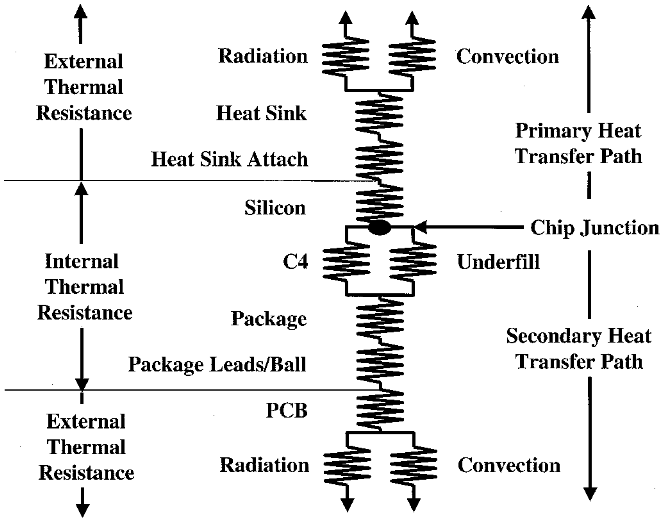


Fig. 5. The simplified thermal resistance network model for PowerPC.

B. Boundary Conditions

One of the critical factors under simulation is to determine the temperature distribution on the boundary conditions. In this paper, the complicated thermal problems of packaging and heat sink are modeled by generating a thermal resistance network. Then, the equivalent thermal resistance, R_θ, of the thermal resistance network is calculated.

For example, the Motorola PowerPC microprocessors which are available in a controlled-collapsed-chip-connection/ceramic-ball-grid-array (C4/CBGA) single-chip package are shown in Fig. 4 [14]. The simplified thermal network for PowerPCs with the C4/CBGA package mounted on a printed-circuit board is illustrated in Fig. 5 [15].

The equivalent thermal resistances, R_θ^i, on the six sides of the chip boundary are applied to model the effective heat transfer coefficient, $h_i^e = 1/A^i R_\theta^i$, where A^i is the chip area normal to the direction of heat flow. Then, (2) can be used to model the equivalent convection boundary conditions with $h_i = h_i^e$ and $f_i = h_i^e T_\infty$

$$\kappa(\vec{r},T)\frac{\partial T(\vec{r},t)}{\partial n_i} = h_i^e(T_\infty - T(\vec{r},t)). \tag{3}$$

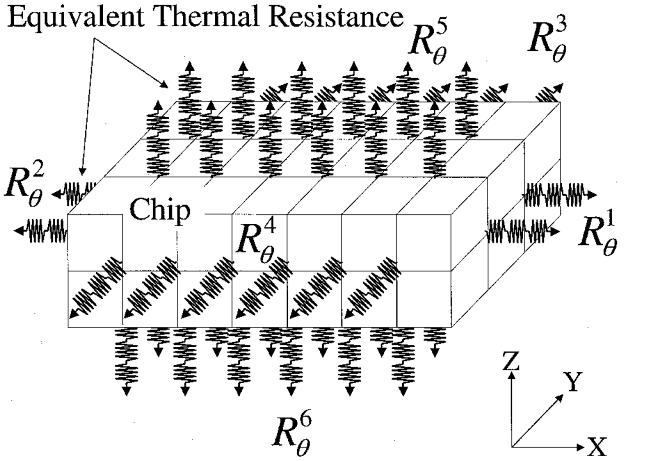


Fig. 6. The modeling strategy of 3D Thermal-ADI simulator. The chip is simulated with the ADI method governed by the 3-D heat equation. The package and the heat sinks are modeled as 1-D equivalent thermal resistances.

In the 3D Thermal-ADI simulator, the problem is modeled as illustrated in Fig. 6. First, the chip containing the interconnects and a portion of the silicon substrate is simulated with the ADI method governed by the 3-D heat equation, in (1), for a high degree of accuracy. The other part of the silicon substrate is included in the thermal resistance network. If the temperature distribution of the whole silicon substrate is interested, big discretization size, Δz, or a large discretization number in the z direction is needed.

Second, the complicated package and heat sink structures are modeled as 1-D equivalent thermal resistances, and the equivalent thermal resistances on each side are employed to simplify the boundary conditions as expressed in (3). Note that the temperature can not be known inside the equivalent thermal resistances, and the absolute magnitude of temperature in the chip is dependent on the external thermal resistance.

C. Mutual Relation of Power–Temperature

The temperature and the power are related to each other. In general, the physical parameters, e.g., resistivity, thermal conductivity, specific heat, density, and carrier mobility are sensitive to temperatures. This implies that the power caused by short-circuit and leakage currents in the gates and the power generated by self-heating in the interconnects are temperature sensitive. For example, the power generated from self-heating which is proportional to resistivity is especially sensitive to temperature. On the other hand, the temperature profile needs the accurate power estimation for heat sources, since the different power distribution may have a very different temperature profile.

Since the temperature and power are mutually related, the update of power and temperature are needed for the calculation in each transient iteration in the ADI method. As shown in Fig. 2, the first power estimation is calculated by the given initial temperature. After that, the estimated power for each transient iteration is dependent on the temperature profile of the previous iteration. Similarly, the calculation of temperature profile by the ADI method in each transient iteration is dependent on the previous power estimation. The process is stopped when the temperature profile converges to a steady state.

In the following section, we will discuss the problem formulation by the finite-difference method to have second-order accuracy both in time and space. We also derive the formulas for homogeneous and nonhomogeneous cases.

III. Problem Formulation by Finite-Difference Method

For a given chip, the temperature distribution is governed by (1) and is subject to the boundary conditions in (3). To solve (1) with the finite-difference method, discretization is needed both in time and space.

A. Discretization

For the homogeneous material, the term $\nabla\cdot[\kappa(\vec{r},T)\nabla T(\vec{r},t)]$ in (1) can be replaced by $\kappa(T)\ \nabla^2\ T(\vec{r},t)$. Then the heat conduction equation can be rewritten as

$$\frac{\partial T(x,y,z,t)}{\partial t} = \alpha\left[\frac{\partial^2 T(x,y,z,t)}{\partial x^2} + \frac{\partial^2 T(x,y,z,t)}{\partial y^2} + \frac{\partial^2 T(x,y,z,t)}{\partial z^2}\right] + \frac{1}{\rho c}g(x,y,z,t) \quad (4)$$

where $\alpha = \kappa/\rho C_p$.

This equation is a second-order parabolic partial differential equation. The first step to establish a finite-difference solution method of the partial differential equation is to discretize the continuous space domain into a grid with a finite number of grid points. At time step n, the temperature $T(x,y,z,t)$ at grid point (i,j,k) can be replaced by $T(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$ which is denoted as $T^n_{i,j,k}$ for the rest of the paper. According to central finite-difference discretization, the second-order partial derivative of T with respect to x can be expressed as

$$\begin{aligned}\frac{\partial^2 T}{\partial x^2}\ |^n_{i,j,k} &= \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} + O(\Delta x)^2 \\ &\approx \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} = \frac{\delta_x^2 T^n}{(\Delta x)^2} \end{aligned} \quad (5)$$

where the truncation error (TR) is $O[(\Delta x)^2]$, and $\delta_x^2 T^n = T^n_{i-1,j,k} - 2T^n_{i,j,k} + T^n_{i+1,j,k}$. A similar process can be applied to the y and z directions.

The next step is to consider the time discretization problem. Since (4) comes from energy conservation, the rate of the energy stored in a control volume equals the net rate of energy transferring into the control volume and power generated. Hence, the forward-difference with time on the left-hand side of (4) is the energy stored from time step n to $n+1$ in the control unit volume. Three time-marching methods will be considered to apply on the right-hand side of (4) with respect to accuracy and stability [16].

- *Simple Explicit Method*

  Applying the explicit update on the right-hand side of the discretized form of (4) at time step n, we obtain

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha\left[\frac{\delta_x^2 T^n}{(\Delta x)^2} + \frac{\delta_y^2 T^n}{(\Delta y)^2} + \frac{\delta_z^2 T^n}{(\Delta z)^2}\right] + \frac{g}{\rho C_p}. \quad (6)$$

  Noting that there is only one unknown, $T^{n+1}_{i,j,k}$, in each equation with respect to point (i,j,k), the solution is obvious. This method has second-order accuracy in space and first-order accuracy in time [16], i.e., $TR = O[(\Delta t),(\Delta x)^2,(\Delta y)^2,(\Delta z)^2]$. Nevertheless, there is a stability constraint γ[16] such that

$$\gamma = \alpha\Delta t\left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}\right) \leq \frac{1}{2}. \quad (7)$$

  This restricts the size of the time step, Δt, for the given space increments, Δx, Δy, and Δz.

- *Simple Implicit Method*

  Applying the simple implicit update on the right-hand side of (4) at time step n, we get

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha\left[\frac{\delta_x^2 T^{n+1}}{(\Delta x)^2} + \frac{\delta_y^2 T^{n+1}}{(\Delta y)^2} + \frac{\delta_z^2 T^{n+1}}{(\Delta z)^2}\right] + \frac{g}{\rho C_p} \quad (8)$$

  even though there are five unknowns in each equation at point (i,j,k) to solve the time step $n+1$. The simple implicit method is unconditionally stable. *TR* is the same as the simple explicit method [16].

- *Crank–Nicolson Method*

  Crank and Nicolson dealt with the time marching problem by taking the average of simple explicit and implicit methods. According to (4) and (5), we have

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha\left[\frac{\delta_x^2 T^{n+1} + \delta_x^2 T^n}{2(\Delta x)^2} + \frac{\delta_y^2 T^{n+1} + \delta_y^2 T^n}{2(\Delta y)^2} + \frac{\delta_z^2 T^{n+1} + \delta_y^2 T^n}{2(\Delta z)^2}\right] + \frac{g}{\rho C_p}. \quad (9)$$

  After rearrangement, we get the difference equation as

$$\begin{aligned} -r_x T^{n+1}_{i-1,j,k} &- r_y T^{n+1}_{i,j-1,k} - r_z T^{n+1}_{i,j,k-1} \\ &+ 2(1 + r_x + r_y + r_z)T^{n+1}_{i,j,k} \\ &- r_x T^{n+1}_{i+1,j,k} - r_y T^{n+1}_{i,j+1,k} - r_z T^{n+1}_{i,j,k+1} \\ &= r_x T^n_{i-1,j,k} + r_y T^n_{i,j-1,k} + r_z T^n_{i,j,k-1} \\ &+ 2(1 - r_x - r_y - r_z)T^n_{i,j,k} + r_x T^n_{i+1,j,k} \\ &+ r_y T^n_{i,j+1,k} + r_z T^n_{i,j,k+1} + \frac{2\Delta t}{\rho C_p} g_{i,j,k} \end{aligned} \quad (10)$$

  where $r_x = \alpha\Delta t/(\Delta x)^2$, $r_y = \alpha\Delta t/(\Delta y)^2$, $r_z = \alpha\Delta t/(\Delta z)^2$, and $i = 1,2,\ldots,I-1$, $j = 1,2,\ldots,J-1$, $k = 1,2,\ldots,K-1$. There are seven unknowns in each equation at point (i,j,k) to solve the time step $n+1$. The Crank–Nicolson method has the best accuracy with $TR = O[(\Delta t)^2,(\Delta x)^2,(\Delta y)^2,(\Delta z)^2]$. This method is also unconditionally stable [16].

The difference equations in (10) are for those nodes whose control volumes include only one material. However, there are several different materials in the chip such as silicon, polysilicon, silicon dioxide, silicon nitride, aluminum, copper, and others. In the next section, we will discuss the nodes having control volumes with two different materials.

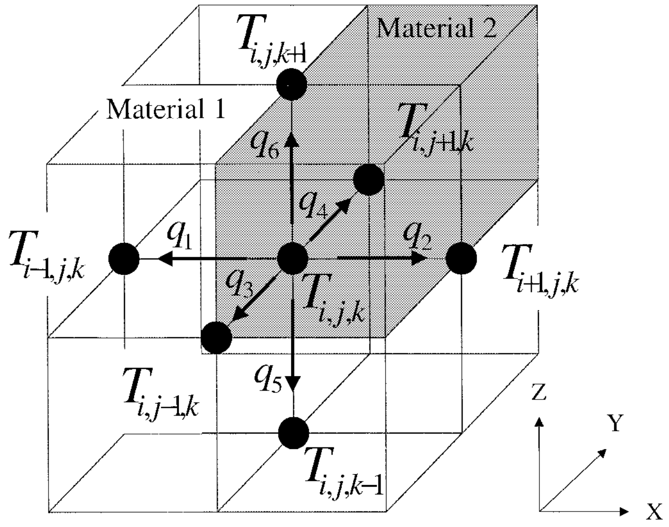


Fig. 7. The nonhomogeneous discretization example with node $T_{i,j,k}$ on the boundary surface of two different materials.

B. Nonhomogeneous Case

Consider the nodes located between two different materials; there are 27 types of possible combinations of layout extraction. For the geometry as shown in Fig. 7, for example, one-quarter of the control volume is material 2 and three-quarters of the control volume is material 1. The rate of energy stored in control volume causing the temperature increase is

$$\frac{dE}{dt} = \left(\frac{3}{4}\Delta V \rho_1 c_1 + \frac{1}{4}\Delta V \rho_2 c_2\right)\frac{\Delta T_{i,j,k}}{\Delta t}$$

where $\Delta V = \Delta x \Delta y \Delta z$ is the control volume. The power generated in control volume is $G = (3/4)\Delta V g_1 + (1/4)\Delta V g_2$. The rate of heat conduction from $T_{i,j,k}$ to $T_{i-1,j,k}$ and $T_{i+1,j,k}$ are $q_1 = \kappa_1((\Delta y \Delta z)/\Delta x)(T_{i,j,k} - T_{i-1,j,k})$ and $q_2 = ((\kappa_1 + \kappa_2)/2)((\Delta y \Delta z)/\Delta x)(T_{i,j,k} - T_{i+1,j,k})$, respectively. The others can be calculated similarly. Then we obtain $q_3 = ((3\kappa_1 + \kappa_2)/4)((\Delta x \Delta z)/\Delta y)(T_{i,j,k} - T_{i,j-1,k})$, $q_4 = ((3\kappa_1 + \kappa_2)/4)((\Delta x \Delta z)/\Delta y)(T_{i,j,k} - T_{i,j+1,k})$, $q_5 = \kappa_1((\Delta x \Delta y)/\Delta z)(T_{i,j,k} - T_{i,j,k-1})$, and $q_6 = ((\kappa_1 + \kappa_2)/2)((\Delta x \Delta y)/\Delta z)(T_{i,j,k} - T_{i,j,k+1})$. From energy conservation, we have

$$g = \frac{dE}{dt} + q_1 + q_2 + q_3 + q_4 + q_5 + q_6. \tag{11}$$

It can be rearranged as follows:

$$\begin{aligned}\frac{\Delta T_{i,j,k}}{\Delta t} =& \frac{1}{\frac{3}{4}\rho_1 C_{p1} + \frac{1}{4}\rho_2 C_{p2}} \\ &\times \left\{ \frac{\frac{\kappa_1+\kappa_2}{2}\frac{T_{i+1,j,k}-T_{i,j,k}}{\Delta x} - \kappa_1 \frac{T_{i,j,k}-T_{i-1,j,k}}{\Delta x}}{\Delta x} \right. \\ &+ \frac{\frac{3\kappa_1+\kappa_2}{4}\frac{T_{i+1,j,k}-T_{i,j,k}}{\Delta y} - \frac{3\kappa_1+\kappa_2}{4}\frac{T_{i,j,k}-T_{i,j-1,k}}{\Delta y}}{\Delta y} \\ &\left. + \frac{\frac{\kappa_1+\kappa_2}{2}\frac{T_{i,j,k+1}-T_{i,j,k}}{\Delta z} - \kappa_1 \frac{T_{i,j,k}-T_{i,j,k-1}}{\Delta z}}{\Delta z} \right\} \\ &+ \frac{3g_1 + g_2}{3\rho_1 C_{p1} + \rho_2 C_{p2}}.\end{aligned} \tag{12}$$

This expression implies that it is the same as taking central finite-difference discretization on (1) as discussed in Section III-A. Therefore, the space accuracy is still second order. The next step is using the Crank–Nicolson method to treat the time discretization resulting in difference equations with second-order accuracy in time. Note that the equations discussed so far are for those points inside the chip. The equations for those points on the boundary will be discussed later.

For a 3-D grid with size $l \times m \times n$, the number of degrees-of-freedom for this system is $N = lmn$, which requires a matrix A with size $N \times N$ to store the coefficients. To solve the equations $Ax = b$ by Cholesky decomposition with ordering the matrix by the minimum degree ordering algorithm, which is known as the fastest decomposition algorithm and the least fill-in ordering method for a grid structure, it is still not fast enough to solve the large size problem. The next section introduces the ADI method.

IV. ADI Method

Peaceman and Rachford [17], and Douglas and Gunn [10] developed a variation on the Crank–Nicolson approximation which is known as the ADI method. In this paper, we discuss only the Douglas–Gunn scheme applied to the thermal problems. The reason is that the Peaceman Rachford approach has second-order accuracy and is unconditionally stable for only the 2-D problems. It leads to conditionally stable and first-order accuracy in time for the 3-D problems.

A. Douglas–Gunn Approach

Douglas and Gunn developed a general ADI scheme that is unconditionally stable and retains second-order accuracy when applied to the 3-D problem. We rewrite (9) as follows:

$$\begin{aligned}T^{n+1} - T^n = & r_x \frac{\delta_x^2}{2}(T^{n+1} + T^n) + r_y \frac{\delta_y^2}{2}(T^{n+1} + T^n) \\ & + r_z \frac{\delta_z^2}{2}(T^{n+1} + T^n) + \frac{\Delta t}{\rho C_p} g.\end{aligned} \tag{13}$$

Instead of directly solving (13) at every time step n, we solve the same equations by three subtime steps at each time step n

$$\begin{aligned}\textbf{Step I}: T^{n+(1/3)} - T^n =& \frac{r_x \delta_x^2}{2}(T^{n+(1/3)} + T^n) \\ & + r_y \delta_y^2 T^n + r_z \delta_z^2 T^n + \frac{\Delta t}{\rho C_p} g\end{aligned} \tag{14}$$

$$\begin{aligned}\textbf{Step II}: T^{n+(2/3)} - T^n =& \frac{r_x \delta_x^2}{2}(T^{n+(1/3)} + T^n) \\ & + \frac{r_y \delta_y^2}{2}(T^{n+(2/3)} + T^n) \\ & + r_z \delta_z^2 T^n + \frac{\Delta t}{\rho C_p} g\end{aligned} \tag{15}$$

$$\begin{aligned}\textbf{Step III}: T^{n+1} - T^n =& \frac{r_x \delta_x^2}{2}(T^{n+(1/3)} + T^n) \\ & + \frac{r_y \delta_y^2}{2}(T^{n+(2/3)} + T^n) \\ & + \frac{r_z \delta_z^2}{2}(T^{n+1} + T^n) + \frac{\Delta t}{\rho C_p} g.\end{aligned} \tag{16}$$

Theorem 1: The ADI method in (14)–(16) is unconditionally stable. The detailed proof is in the Appendix .

This method splits the time march from n to $n+1$ into three steps: from n to $n+(1/3)$, $n+(1/3)$ to $n+(2/3)$, and $n+(2/3)$ to $n+1$ as shown in Fig. 3. In *Step I*, the x direction is implicit, but the y and z directions are explicit. For each (y,z) row of grid points, there are $I-1$ equations of the corresponding (x,y,z) point from (14), the other two equations of the corresponding point from boundary conditions, which will be discussed in Section IV-B. Since each point (x,y,z) is related to three unknown variables $T^{n+(1/3)}_{i-1,j,k}$, $T^{n+(1/3)}_{i,j,k}$, and $T^{n+(1/3)}_{i+1,j,k}$, it is a tridiagonal system of equations which must be solved for each (y,z) row of grid points. The tridiagonal matrix can be solved by the Thomas algorithm [13] with $O(n)$ time, where n is the discretization node number in the x direction. Similarly, it applies to *Step II* and *Step III*.

The detailed difference equations for these three steps with the homogeneous and nonhomogeneous cases can be derived. For example, the three steps of the homogeneous case can be derived as follows:

$$\begin{aligned}\textbf{Step I}:& -r_xT^{n+(1/3)}_{i-1,j,k}+2(1+r_x)T^{n+(1/3)}_{i,j,k}-r_xT^{n+(1/3)}_{i+1,j,k}\\ &=\{r_xT^n_{i-1,j,k}+2r_yT^n_{i,j-1,k}\\ &\quad+2r_zT^n_{i,j,k-1}\\ &\quad+2(1-r_x-2r_y-2r_z)T^n_{i,j,k}\\ &\quad+r_xT^n_{i+1,j,k}+2r_yT^n_{i,j+1,k}\\ &\quad+2r_zT^n_{i,j,k+1}\}+\frac{2\Delta t}{\rho C_p}g_{i,j,k} \qquad (17)\end{aligned}$$

$$\begin{aligned}\textbf{Step II}:& -r_yT^{n+(2/3)}_{i,j-1,k}+2(1+r_y)T^{n+(2/3)}_{i,j,k}-r_yT^{n+(2/3)}_{i,j+1,k}\\ &=\left\{r_xT^{n+(1/3)}_{i-1,j,k}-2r_xT^{n+(1/3)}_{i,j,k}\right.\\ &\quad\left.+r_xT^{n+(1/3)}_{i+1,j,k}\right\}+\{r_xT^n_{i-1,j,k}\\ &\quad+r_yT^n_{i,j-1,k}+2r_zT^n_{i,j,k-1}\\ &\quad+2(1-r_x-r_y-2r_z)T^n_{i,j,k}\\ &\quad+r_xT^n_{i+1,j,k}+r_yT^n_{i,j+1,k}\\ &\quad+2r_zT^n_{i,j,k+1}\}+\frac{2\Delta t}{\rho C_p}g_{i,j,k} \qquad (18)\end{aligned}$$

$$\begin{aligned}\textbf{Step III}:& -r_zT^{n+1}_{i,j,k-1}+2(1+r_z)T^{n+1}_{i,j,k}-r_zT^{n+1}_{i,j,k+1}\\ &=\left\{r_xT^{n+(1/3)}_{i-1,j,k}-2r_xT^{n+(1/3)}_{i,j,k}\right.\\ &\quad\left.+r_xT^{n+(1/3)}_{i+1,j,k}\right\}+\left\{r_yT^{n+(2/3)}_{i,j-1,k}\right.\\ &\quad\left.-2r_yT^{n+(2/3)}_{i,j,k}+r_yT^{n+(2/3)}_{i,j+1,k}\right\}\\ &\quad+\{r_xT^n_{i-1,j,k}+r_yT^n_{i,j-1,k}\\ &\quad+r_zT^n_{i,j,k-1}\\ &\quad+2(1-r_x-r_y-r_z)T^n_{i,j,k}\\ &\quad+r_xT^n_{i+1,j,k}+r_yT^n_{i,j+1,k}\\ &\quad+r_zT^n_{i,j,k+1}\}+\frac{2\Delta t}{\rho C_p}g_{i,j,k}. \qquad (19)\end{aligned}$$

In *Step I* as shown in (17), there are $I+1$ equations for each (j,k) value. Also there are three unknown variables $T^{n+(1/3)}_{i-1,j,k}$, $T^{n+(1/3)}_{i,j,k}$, and $T^{n+(1/3)}_{i+1,j,k}$ for each equation. Then this system can be expressed as

$$\begin{bmatrix}\diamond & \diamond & & & & \\ \star & \star & \star & & & \\ & \star & \star & \star & & \\ & & \star & \star & \star & \\ & & & \vdots & & \\ & & & \star & \star & \star \\ & & & & \diamond & \diamond\end{bmatrix}\begin{bmatrix}T^{n+(1/3)}_{0,j,k}\\ T^{n+(1/3)}_{1,j,k}\\ \vdots\\ T^{n+(1/3)}_{I-1,j,k}\\ T^{n+(1/3)}_{I,j,k}\end{bmatrix}=\begin{bmatrix}b_0\\ b_1\\ \vdots\\ b_{I-1}\\ b_I\end{bmatrix} \qquad (20)$$

where b_i are the values on the right-hand side of the equal sign in (17). Note the coefficients denoted with diamonds will be determined by the boundary conditions. We will discuss this in the next section. The tridiagonal matrix can be solved by the Thomas algorithm [16] efficiently. For the 27 types of nonhomogeneous cases, the difference equations by the ADI method can be calculated with the same rule.

B. Boundary Conditions

So far the equations we have discussed are only for the points inside the chip. In this section, the equations related to the boundary conditions will be discussed. These complete the system to be solved, with the number of the unknown variables equal to the number of the equations.

As mentioned in Section II-B, the complicated package and heat sink problems can be modeled by (3). Suppose that we have following boundary conditions:

$$\begin{aligned}-\kappa\frac{\partial T}{\partial x}+h^e_{x-}T&=h^e_{x-}T_\infty \text{ at surface } x=0\\ \kappa\frac{\partial T}{\partial x}+h^e_{x+}T&=h^e_{x+}T_\infty \text{ at surface } x=L\\ -\kappa\frac{\partial T}{\partial y}+h^e_{y-}T&=h^e_{y-}T_\infty \text{ at surface } y=0\\ \kappa\frac{\partial T}{\partial y}+h^e_{y+}T&=h^e_{y+}T_\infty \text{ at surface } y=M\\ -\kappa\frac{\partial T}{\partial z}+h^e_{z-}T&=h^e_{z-}T_\infty \text{ at surface } z=0\\ \kappa\frac{\partial T}{\partial z}+h^e_{z+}T&=h^e_{z+}T_\infty \text{ at surface } z=N \qquad (21)\end{aligned}$$

where h^e_{x-}, h^e_{x+}, h^e_{y-}, h^e_{y+}, h^e_{z-}, and h^e_{z+} are the effective heat transfer coefficients calculated from the equivalent thermal resistance on the boundary $x=0$, $x=L$, $y=0$, $y=M$, $z=0$, and $z=N$, respectively.

In order to achieve second-order accuracy, the central-difference approximation will be used to discretize the boundary condition equations. First, we introduce the virtual temperature nodes $T^n_{-1,j,k}$, $T^n_{I+1,j,k}$, $T^n_{i,-1,k}$, $T^n_{i,J+1,k}$, $T^n_{i,j,-1}$, and $T^n_{i,j,K+1}$ by expanding the distance Δx, Δy, and Δz to external boundary. Then we apply the central-difference approximation to discretize the boundary condition equations, e.g., at $i=0$ and time step n in (21), we have

$$-\kappa\frac{T^n_{1,j,k}-T^n_{-1,j,k}}{2\Delta x}+h^e_{x-}T^n_{0,j,k}=h^e_{x-}T_\infty. \qquad (22)$$

Thus the virtual point can be expressed as

$$T^n_{-1,j,k}=T^n_{1,j,k}+\frac{2\Delta x h^e_{x-}}{\kappa}(T_\infty-T^n_{0,j,k}). \qquad (23)$$

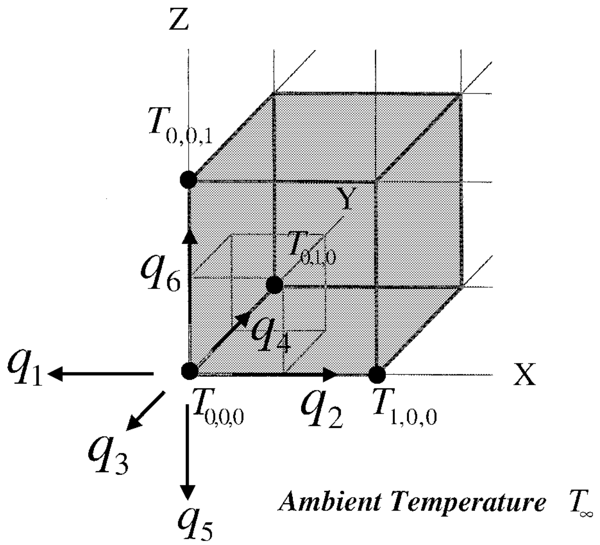


Fig. 8. The boundary conditions for node at point (0,0,0).

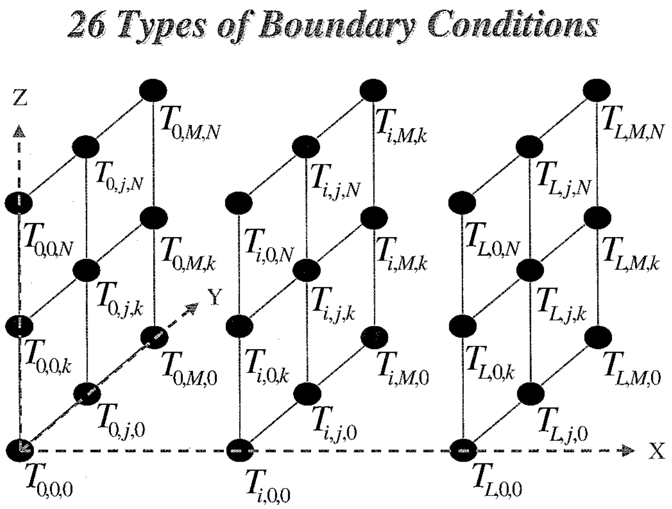


Fig. 9. There are 26 different locations of the point $T_{i,j,k}$ on the boundary surface. $T_{i,0,N}$, for example, shows that the location of the points is on intersection of the surface $y = 0$ and surface $z = N$.

The other virtual points, $T^n_{I+1,j,k}, T^n_{i,-1,k}, T^n_{i,J+1,k}, T^n_{i,j,-1}$, and $T^n_{i,j,K+1}$, can be derived in the same way. Then, these derived virtual points can be used to eliminate the virtual points occurring on the boundary points.

For instance, consider the point (0,0,0) on the boundary as shown in Fig. 8. The energy generated in the control volume is $G = (1/8)\Delta V g$. The rate of energy stored causing the temperature increase is $(dE/dt) = (1/8)\Delta V \rho C_p(\Delta T_{0,0,0}/\Delta t)$. The rates of heat transfer as shown in Fig. 8 are $q_1 = h^e_{x-}(\Delta y \Delta z/4)(T_{0,0,0} - T_\infty)$, $q_2 = \kappa_1(\Delta y \Delta z/4\Delta x)(T_{0,0,0} - T_{1,0,0})$, $q_3 = h^e_{y-}(\Delta x \Delta z/4)(T_{0,0,0} - T_\infty)$, $q_4 = \kappa_1(\Delta x \Delta z/4\Delta y)(T_{0,0,0} - T_{0,1,0})$, $q_5 = h^e_{z-}(\Delta x \Delta y/4)(T_{0,0,0} - T_\infty)$, and $q_6 = \kappa_1(\Delta x \Delta y/4\Delta z)(T_{0,0,0} - T_{0,0,1})$. Then, the difference equation of the Crank–Nicolson method derived by energy conservation at point (0,0,0) is

$$G = \frac{dE}{dt} + q_1 + q_2 + q_3 + q_4 + q_5 + q_6. \tag{24}$$

After rearranging, we have

$$\begin{aligned}
&\frac{1}{4}(1 + r_x\beta_{x-} + r_y\beta_{y-} + r_z\beta_{z-})T^{n+1}_{0,0,0}\\
&\quad - \frac{r_x}{4}T^{n+1}_{1,0,0} - \frac{r_y}{4}T^{n+1}_{0,1,0} - \frac{r_z}{4}T^{n+1}_{0,0,1}\\
&=\frac{1}{4}(1 - r_x\beta_{x-} - r_y\beta_{y-} - r_z\beta_{z-})T^{n}_{0,0,0}\\
&\quad + \frac{r_x}{4}T^{n}_{1,0,0} + \frac{r_y}{4}T^{n}_{0,1,0} + \frac{r_z}{4}T^{n}_{0,0,1}\\
&\quad + \frac{1}{2}(r_x r_{hx-} + r_y r_{hy-} + r_z r_{hz-}) + \frac{\Delta t}{4\rho C_p}g_{0,0,0}
\end{aligned} \tag{25}$$

where $r_{hx-} = (h^e_{x-}\Delta x T_\infty)/\kappa$, $r_{hy-} = (h^e_{y-}\Delta y T_\infty)/\kappa$, $r_{hz-} = (h^e_{z-}\Delta z T_\infty)/\kappa$, $\beta_{x-} = 1 + (h^e_{x-}\Delta x/\kappa)$, $\beta_{y-} = 1 + (h^e_{y-}\Delta y/\kappa)$, $\beta_{z-} = 1 + (h^e_{z-}\Delta z/\kappa)$. This equation is the same as substituting virtual points, $T^{n+1}_{-1,0,0}$, $T^{n+1}_{0,-1,0}$, $T^{n+1}_{0,0,-1}$, $T^{n}_{-1,0,0}$, $T^{n}_{0,-1,0}$, and $T^{n}_{0,0,-1}$ into the Crank–Nicolson equation at point (0,0,0) in (10). Note that the coefficients in (25) are very important to maintain the symmetry of the matrix when solving the problem. If both sides of (25) are multiplied by 4, the symmetry of the matrix will be destroyed. There are 26 different locations of the point $T_{i,j,k}$ on the boundary surface as shown in Fig. 9. All 26 types can be derived in the same way. Furthermore, the difference equations of the 26 different locations on the boundary conditions of the ADI method can similarly be derived by substituting virtual points for each step from (17) to (19).

V. Experimental Results and Discussion

The proposed simulator, 3D Thermal-ADI, was implemented with C++ language and executed on a PC with a 1.4-GHz Pentium 4 processor and 1 GB of memory. Note that the matrix in the Crank–Nicolson method is solved by the Choleskey decomposition with ordering of the matrix by the minimum degree ordering method in order to make a fair comparison.

The runtime comparison of the simulator with the Crank–Nicolson method and the 3D Thermal-ADI method per iteration is illustrated in Fig. 10. Note that the scale in the y axis is a logarithm on the top. As can be seen in Fig. 10, the runtime of 3D Thermal-ADI is linearly proportional to the number of the grid nodes. The time complexity can be analyzed as follows. Suppose that the grid size is $l \times m \times n$, then the total node count is $N = lmn$. Three subroutine steps are executed in each transient iteration, and each step needs $O(lmn)$. For example, in *Step I*, we need to solve the tridiagonal matrix with runtime $O(m)$ for each (y, z) value. Therefore, the total time complexity is $O(itr \times 3 \times lmn) = O(N)$, where itr is the number of transient iterations. However, the runtime of the Crank–Nicolson method increases dramatically.

The comparison of memory usages of the simulator with the Crank–Nicolson method and the 3D Thermal-ADI method is illustrated in Fig. 11. On the top figure, the y axis is also a logarithm. The memory usage of the 3D Thermal-ADI is linear with respect to the node number. However, the memory usage of the

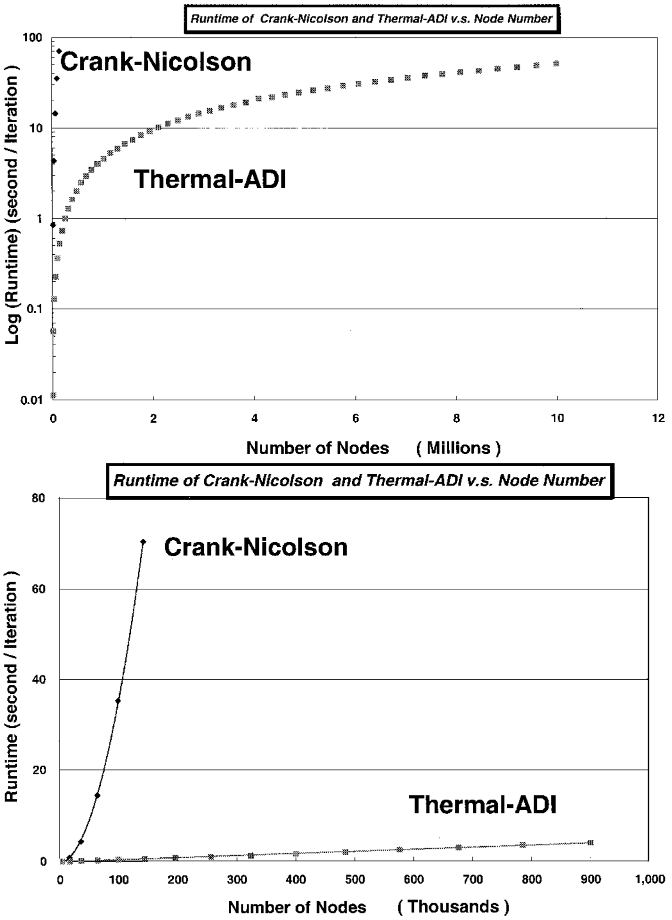


Fig. 10. Runtime comparison of the simulations with the Crank–Nicolson and the 3D Thermal-ADI approaches.

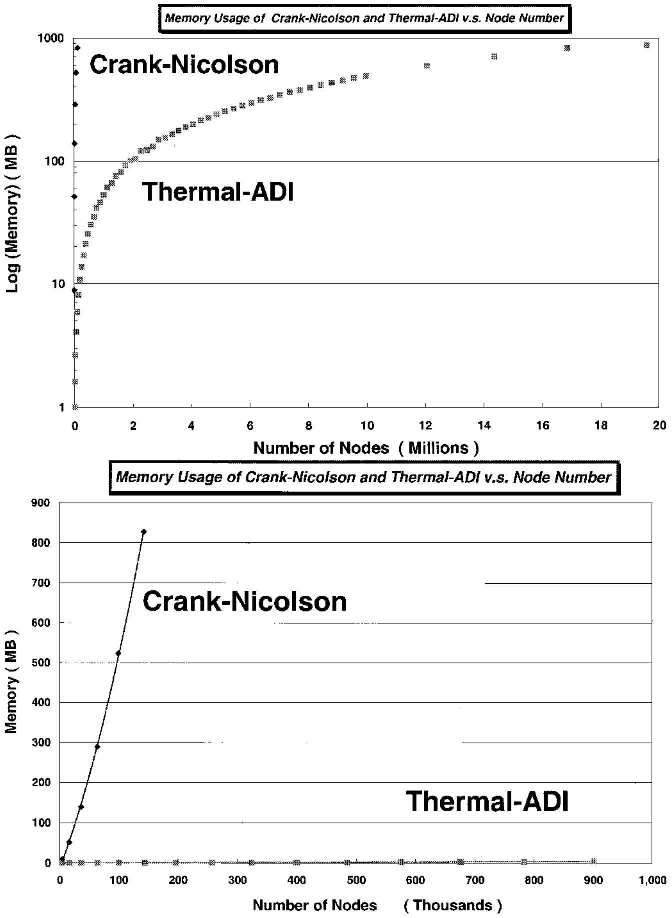


Fig. 11. Comparison of the memory usage of the simulations with the Crank–Nicolson and the 3D Thermal-ADI approaches.

Crank–Nicolson method increases quickly, which limits the size of the problem which can be solved on a given machine.

Next consider a chip whose layout is illustrated with hierarchical function blocks in Fig. 12. The size of the chip is 11.3 mm × 14.4 mm. The power in each block is shown in Fig. 13. The chip is discretized in the x direction with $\Delta x = 20$ μm, the y direction with $\Delta y = 20$ μm, and the z direction with $\Delta z = 20$ μm. Therefore, the discretization size is 565 × 720 × 7. Here, we only consider the volume near the substrate surface, and the substrate is included in the equivalent thermal resistance as discussed in Section II-B. The effective heat transfer coefficients, h_i^e, are supposed to be 7×10^3 W/m^2 K in the primary heat transfer path, 8×10^4 W/m^2 K in the secondary heat transfer path, and 5×10^3 W/m^2 K around the side faces. The size of the time increment is $\Delta t = 10^{-4}$ s, and the simulation runs 1200 iterations. The runtime is about 311 min. The results of the temperature profile are shown in Fig. 14. The highest temperature is about 180 °C which is influenced by the effective heat transfer coefficients. There are three main parameters that affect the temperature: the board-level component population (thermal loading), the heat sink style and design, and the air velocity on the components and/or the heat sink [15].

Let us observe a random chosen point $T_{i,j,k}$ with $i = 60, j = 500$, and $k = 3$, which is on the substrate surface. The transient temperature profile at this point is shown in Fig. 15. From the figure we can find that the thermal time constant in this problem is about 0.07 s. The temperature at point $(i, j, k) = (60{,}500{,}3)$ reaches steady state at time 0.07 s or at iteration 700. The temperature difference between time 0.07 and 0.12 s is 0.3%.

However, we need 700 iterations to reach the steady state in this circuit. In order to decrease the number of iterations to approach the steady state, we can increase the size of time step Δt. Is there any limit for the maximum Δt? For the problem solved by the Crank–Nicolson method, there is a criterion for the maximum Δt[18]. If the Δt is bigger than the criterion, then the simulation begins to oscillate. After careful analysis, the estimate Δt of the critical time step can be expressed as follows:

$$\Delta t_c = \min\left\{\frac{1}{2((1-s)\alpha_1 + s\alpha_2)} \times \left[\frac{1}{\frac{b_x}{\Delta x^2} + \frac{b_y}{\Delta y^2} + \frac{b_z}{\Delta z^2}}\right]\right\}_{(i,j,k)} \qquad (26)$$

where s depends on the location of the point $T_{i,j,k}$ as discussed in the nonhomogeneous case, and b_x, b_y, and b_z depend on the location of the point $T_{i,j,k}$ in the boundary conditions. The meaning of (26) is as follows. Calculate the value of each discretization point $T_{i,j,k}$ on the right-hand side of (26), then find the minimum value which is the criterion for the maximum Δt among all points. For example, the case shown in Fig. 7 has one-quarter of material 2 and three-quarters of material 1. Therefore, the value of s is $1/4$. We have $b_x = (1/2)(1 + \beta_{\pm x})$,

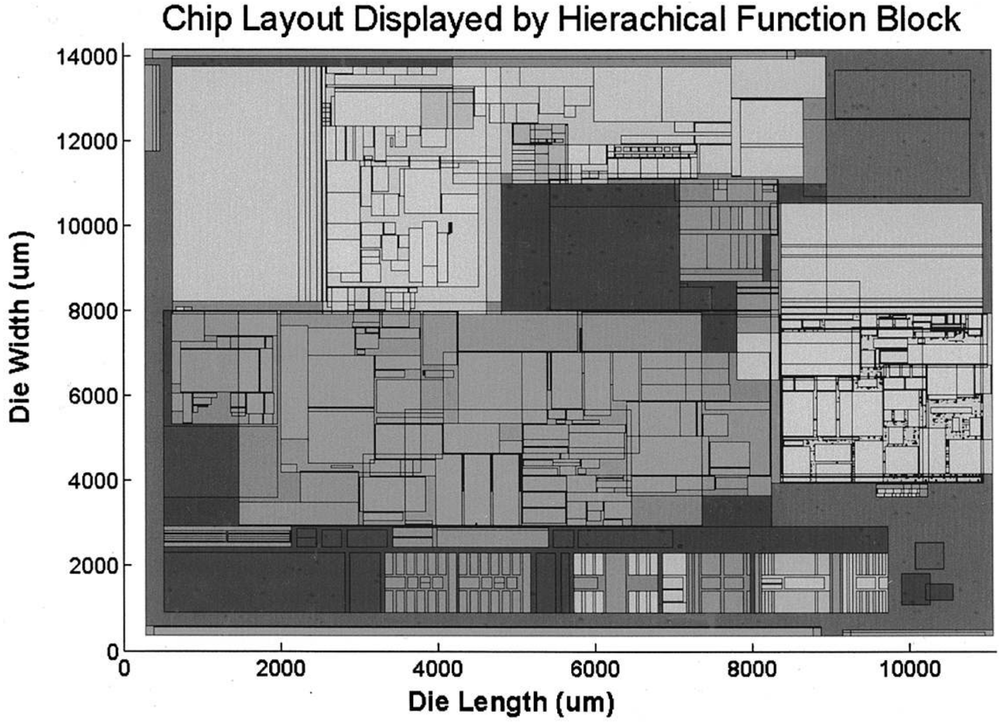


Fig. 12. A chip layout example is displayed with hierarchical function blocks.

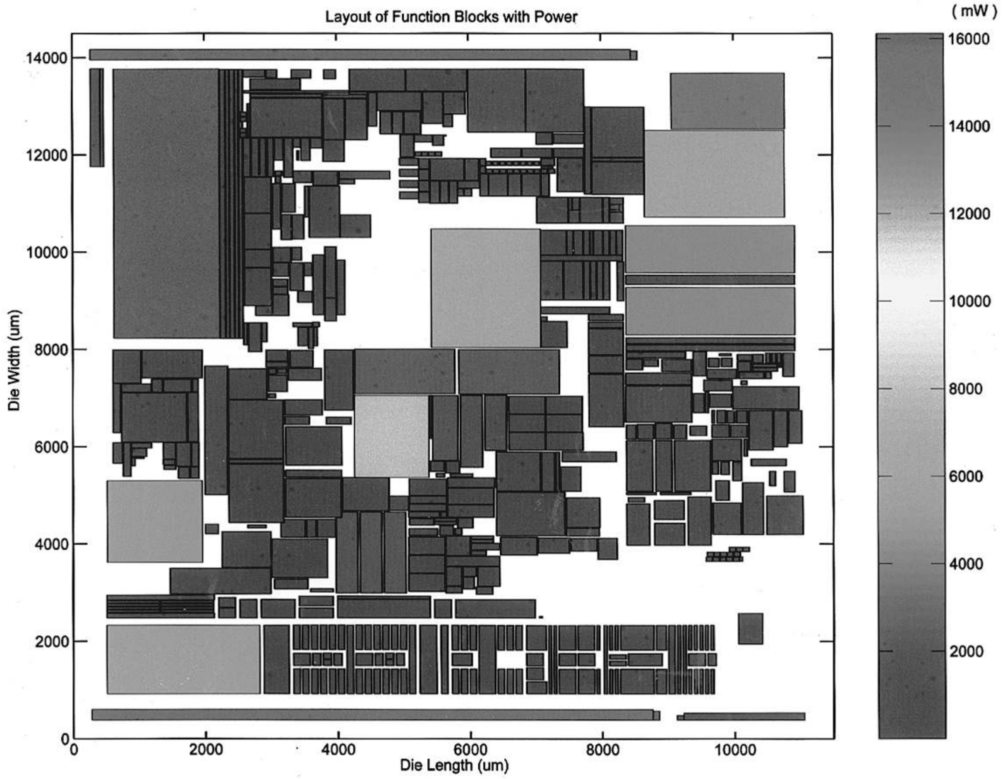


Fig. 13. The illustration of the power in each function block.

$b_y = (1/2)(1 + \beta_{\pm y})$, and $b_z = (1/2)(1 + \beta_{\pm z})$ if the point is located on the boundary surface x, y, and z, respectively. Otherwise, we have $b_x = 1$, $b_y = 1$, and $b_z = 1$. Let us look at the example in Fig. 8. The point $T_{0,0,0}$ is located on the left plane at $x = 0$, the front plane at $y = 0$, and the bottom plane at $z = 0$. Hence, we have $b_x = (1/2)(1 + \beta_{+x})$, $b_y = (1/2)(1 + \beta_{+y})$, and $b_z = (1/2)(1 + \beta_{+z})$. The value of Δt_c in this example is 8.293×10^{-7} s.

However, for the problem solved by the Thermal-ADI method, there is no such restriction. The transient results for

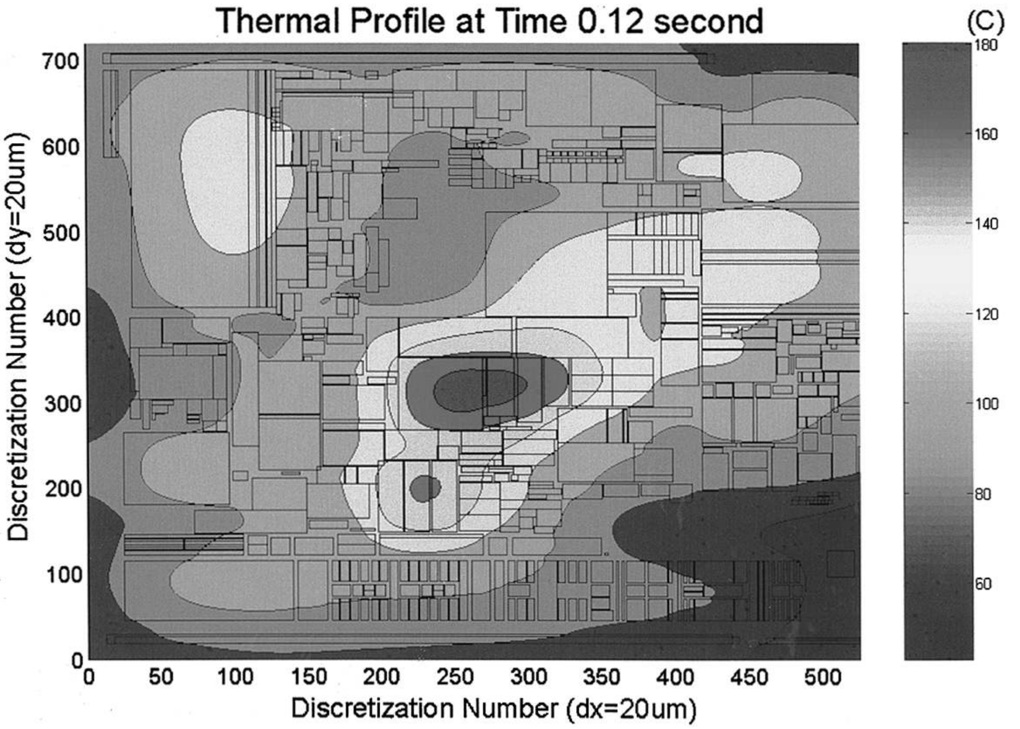


Fig. 14. Thermal profile at $t = 0.12$ s.

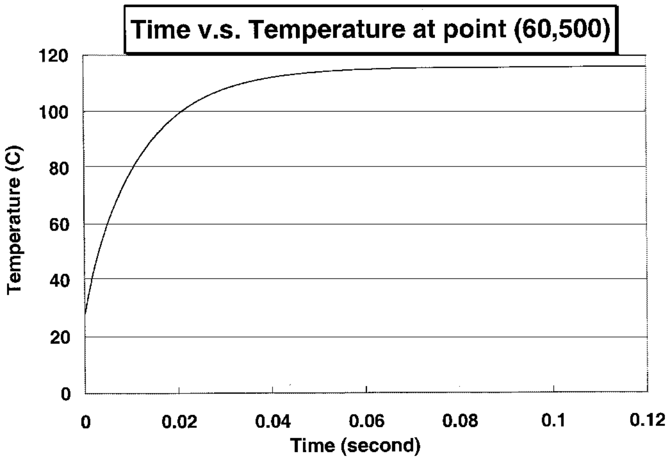


Fig. 15. The transient results for a point at (60,500,3) with $\Delta t = 10^{-4}$ s.

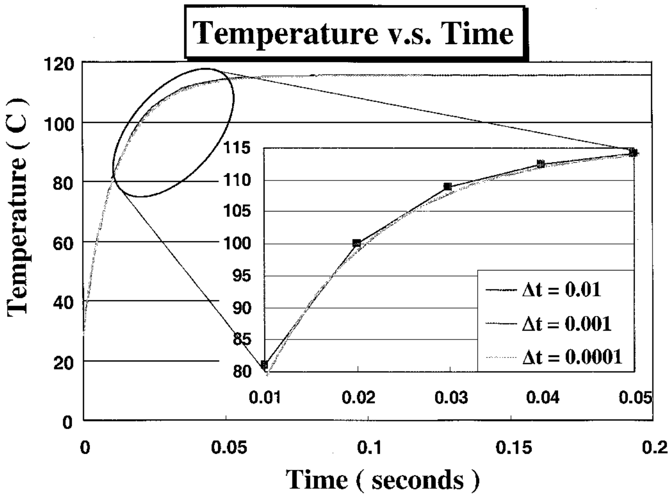


Fig. 16. The transient results for a point at (60,500,3) with $\Delta t = 10^{-4}$ s, $\Delta t = 10^{-3}$ s, and $\Delta t = 10^{-2}$ s.

a point at $(i = 60, j = 500, k = 3)$ with $\Delta t = 10^{-4}$ s, $\Delta t = 10^{-3}$ s, and $\Delta t = 10^{-2}$ s are shown in Fig. 16. Obviously, there is no oscillation for Δt bigger than 8.293×10^{-7} s. If the time increment Δt is 10^{-2} s, it only takes six iterations to reach the thermal time constant. Therefore, ten iterations are enough to reach the steady state. However, it takes 700 iterations for $\Delta t = 10^{-4}$ s and 70 iterations for $\Delta t = 10^{-3}$ s to reach the steady state, respectively.

From Fig. 16, there are deviations between the transient thermal results. How big are the differences between the results? The differences between the transient temperature results by comparing $\Delta t = 10^{-4}$ to $\Delta t = 10^{-3}$ are from 0.007% to 0.037% in 0.1 s. Comparing $\Delta t = 10^{-4}$ with $\Delta t = 10^{-2}$, the differences are from 0.03% to 2.96% in 0.1 s. Therefore, the error of $\Delta t = 10^{-2}$ is about 100 times larger than $\Delta t = 10^{-4}$. It satisfies that the Thermal-ADI has second-order accuracy in time. Therefore, a tradeoff between the runtime (i.e., iteration number) and accuracy depends on the designer's requirements.

If the time increment is too big, the Thermal-ADI method still can convergence to steady state. For instance, the time increments are $\Delta t = 0.05$ s and $\Delta t = 0.1$ s as shown in Fig. 17. Even though we cannot know the transient results because of the oscillation, the results still converge to steady state in ten iterations.

VI. Conclusion

In this paper, an efficient transient 3-D thermal simulator based on the ADI method, 3D Thermal-ADI, has been presented

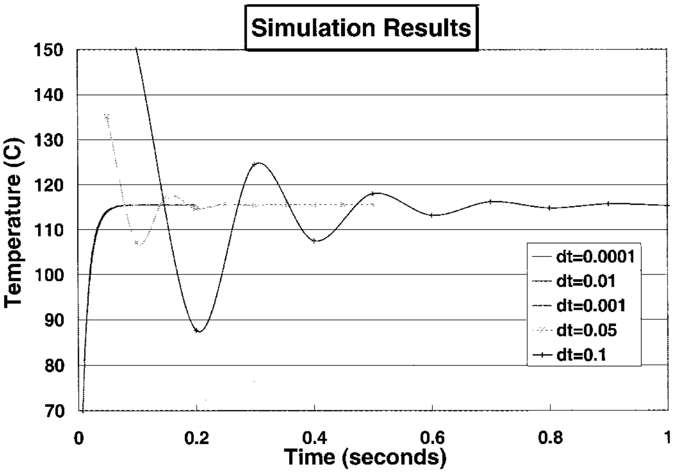


Fig. 17. The transient results for a point at (60,500,3) with $\Delta t = 10^{-4}$ s, $\Delta t = 10^{-3}$ s, $\Delta t = 5 \times 10^{-2}$ s, $\Delta t = 10^{-2}$ s, and $\Delta t = 0.1$ s.

and developed. Detailed analyses of the 3-D nonhomogeneous material and boundary conditions for on-chip VLSI applications are also presented. The experimental results show that our simulator not only has a linear runtime and memory requirement, but also is unconditionally stable. The temperature profile of steady state can also be reached in ten iterations.

In the future, we will extend the 3D Thermal-ADI as a general tool to develop an electrothermal simulator as well as a thermal reliability diagnosis tool.

Appendix

In this section the unconditional stability of the ADI method in (14)-(16) is proved.

A. Proof

After rearranging (14)-(16), we have the following form:

$$\text{Step 1}: \left(1 - \frac{1}{2} r_x \delta_x^2\right) \Delta T^{n+(1/3)} = (r_x \delta_x^2 + r_y \delta_y^2 + r_z \delta_z^2) T^n + \frac{\Delta t}{\rho C_p} g$$

$$\text{Step 2}: \left(1 - \frac{1}{2} r_y \delta_y^2\right) \Delta T^{n+(2/3)} = \Delta T^{n+(1/3)}$$

$$\text{Step 3}: \left(1 - \frac{1}{2} r_z \delta_z^2\right) \Delta T^{n+1} = \Delta T^{n+(2/3)}$$

where

$$\begin{aligned} \Delta T^{n+(1/3)} &\equiv T^{n+(1/3)} - T^n \\ \Delta T^{n+(2/3)} &\equiv T^{n+(2/3)} - T^n \\ \Delta T^{n+1} &\equiv T^{n+1} - T^n. \end{aligned}$$

In order to see the ADI method is unconditionally stable, we take the discrete Fourier transform of the nonhomogeneous case of equations to have

$$\begin{aligned} (1 + r_x(1 - \cos k_x))\hat{T}_{i,j,k}^{n+(1/3)} &= (1 - r_x(1 - \cos k_x) \\ &\quad -2r_y(1 - \cos k_y) \\ &\quad -2r_z(1 - \cos k_z))\, \hat{T}_{i,j,k}^{n} \\ (1 + r_y(1 - \cos k_y))\hat{T}_{i,j,k}^{n+(2/3)} &= \hat{T}_{i,j,k}^{n+(1/3)} \\ &\quad + r_y(1 - \cos k_y)\hat{T}_{i,j,k}^{n} \\ (1 + r_z(1 - \cos k_z))\hat{T}_{i,j,k}^{n+1} &= \hat{T}_{i,j,k}^{n+(2/3)} \\ &\quad + r_z(1 - \cos k_z)\hat{T}_{i,j,k}^{n}. \end{aligned}$$

Solving the equations for $\hat{T}_{i,j,k}^{n+1}$ as a function of $\hat{T}_{i,j,k}^{n}$, we have the equation shown at the bottom of the page. From that equation, we have the second equation shown at the bottom of the page. Therefore, the ADI method is unconditionally stable.

$$\begin{aligned} \hat{T}_{i,j,k}^{n+1} =& \frac{1}{1 + r_z(1 - \cos k_z)} T_{i,j,k}^{n+(2/3)} + \frac{r_z(1 - \cos k_z)}{1 + r_z(1 - \cos k_z)} \hat{T}_{i,j,k}^{n} \\ =& \frac{1}{[1 + r_z(1 - \cos k_z)][1 + r_y(1 - \cos k_y)]} \hat{T}_{i,j,k}^{n+(1/3)} \\ &+ \frac{r_y(1 - \cos k_y)}{[1 + r_z(1 - \cos k_z)][1 + r_y(1 - \cos k_y)]} T_{i,j,k}^{n} \\ &+ \frac{r_z(1 - \cos k_z)}{1 + r_z(1 - \cos k_z)} \hat{T}_{i,j,k}^{n} \\ =& \left\{ \frac{1 - r_x(1 - \cos k_x) - 2r_y(1 - \cos k_y) - 2r_z(1 - \cos k_z)}{[1 + r_x(1 - \cos k_x)][1 + r_y(1 - \cos k_y)][1 + r_z(1 - \cos k_z)]} \right. \\ &\left. + \frac{r_y(1 - \cos k_y)}{[1 + r_y(1 - \cos k_y)][1 + r_z(1 - \cos k_z)]} + \frac{r_z(1 - \cos k_z)}{1 + r_z(1 - \cos k_z)} \right\} \hat{T}_{i,j,k}^{n} \\ =& \frac{[1 - r_x(1 - \cos k_x)][1 - r_y(1 - \cos k_y)][1 - r_z(1 - \cos k_z)]}{[1 + r_x(1 - \cos k_x)][1 + r_y(1 - \cos k_y)][1 + r_z(1 - \cos k_z)]} \hat{T}_{i,j,k}^{n} \end{aligned}$$

$$\frac{[1 - r_x(1 - \cos k_x)][1 - r_y(1 - \cos k_y)][1 - r_z(1 - \cos k_z)]}{[1 + r_x(1 - \cos k_x)][1 + r_y(1 - \cos k_y)][1 + r_z(1 - \cos k_z)]} < 1.$$

REFERENCES

[1] K. Banerjee, M. Pedram, and A. H. Ajami, "Analysis and optimization of thermal issues in high-performance vlsi," in *ACM/SIGDA Int. Symp. Physical Design (ISPD)*, Apr. 2001, pp. 230–237.

[2] K. Banerjee and A. Mehrotra, "Global (interconnect) warming," *IEEE Circuits Devices Mag.*, vol. 17, pp. 16–32, Sept. 2001.

[3] K. Banerjee, A. Mehrotra, A. Sangiovanni-Vincentelli, and C. Hu, "On thermal effects in deep sub-micron vlsi interconnects," in *36th ACM/IEEE Design Automation Conf.*, 1999, pp. 885–891.

[4] V. Székely, "Tracing the thermal behavior of ics," *IEEE Design Test Comput.*, vol. 15, pp. 14–21, Apr.-June 1998.

[5] S. Im and K. Banerjee, "Full chip thermal analysis of planar (2-d) and vertically integrated (3-d) high performance ics," *Tech. Dig. IEEE Int. Electron Devices Meeting (IEDM)*, pp. 727–730, Dec. 2000.

[6] Y.-K. Cheng, P. Raha, C.-C. Teng, E. Rosenbaum, and S.-M. Kang, "Illiads-t: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of cmos vlsi chips," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 668–681, Aug. 1998.

[7] D. Chen, E. Li, E. Rosenbaum, and S.-M. Kang, "Interconect thermal modeling for accurate simulation of circuit timing and reliability," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 197–205, Feb. 2000.

[8] Z. Yu, D. Yergeau, and R. W. Dutton, "Full chip thermal simulation," in *Proc. Int. Symp. Quality Electronic Design*, Mar. 2000, pp. 145–149.

[9] T.-Y. Wang and C. C.-P. Chen, "Thermal-adi: A linear-time chip-level dynamic thermal simulation algorithm based on alternating-direction-implicit (adi) method," in *Proc. 2001 Int. Symp. Physical Design*, Apr. 2001, pp. 238–243.

[10] J. Douglas Jr and J. E. Gunn, "A general formulation of alternating direction methods-Part I. Parabolic and hyperbolic problems," *Numerische Mathematik*, vol. 6, pp. 428–453, 1964.

[11] L.-P. Yuan, C.-C. Teng, and S.-M. Kang, "Statistical estimation of average power dissipation," in *Proc. 34th ACM/IEEE Design Automation Conf.*, 1997.

[12] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A monte carlo approach for power estimation," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 63–71, Mar. 1993.

[13] M. N. Ozisik, *Boundary Value Problems of Heat Conduction*. New York: Dover, 1968.

[14] J. Parry, H. Rosten, and G. B. Kromann, "The development of component-level thermal compact models of a c4/cbga interconnect technology: The motorola powerpc 603 and powerpc 604 risc microprocessors," *IEEE Trans. Components, Packaging, and Manufacturing Technol.*, pt. A, vol. 21, no. 1, pp. 104–112, Mar. 1998.

[15] B. B. Kromann, "Thermal management of a c4/ceramic-ball-grid array: The motorola powerpc 603 and powerpc 604 risc microprocessors," in *Proc. Semiconductor Thermal Measurement and Management Symp.*, 1996, pp. 36–42.

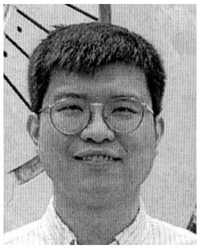
[16] M. N. Ozisik, *Finite Difference Methods in Heat Transfer*. New York: CRC, 1994.

[17] D. W. Peaceman and H. H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Indust. Appl. Math.*, vol. 3, pp. 28–41, 1955.

[18] G. E. Myers, *Analytical Methods in Conduction Heat Transfer*, 2nd ed: AMCHT, 1987.

Ting-Yuan Wang received the B.S. degree in physics from the University of Cheng-Kung University, Tainan, Taiwan, R.O.C., in 1990, the M.S. degrees in physics and electrical and computer engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1992, and the University of Wisconsin-Madison, in 1998, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of Wisconsin-Madison.

Since 1999, he has been a Research Assistant in the VLSI-EDA research group at the University of Wisconsin, Madison. In fall 2000, spring 2001, and spring 2002, he was a Teaching Assistant in the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison. His research interests include the areas of computer-aided design on VLSI circuits and systems, with emphasis on thermal simulation, IC reliability, and circuit optimization.



Charlie Chung-Ping Chen received the B.S degree in computer science and information engineering from the National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 1990, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Austin, in 1996 and 1998.

From 1996 to 1999, he was with Intel Corporation as a Senior CAD Engineer with Strategic CAD Labs. He was in charge of several interconnect and circuit synthesis projects in the microprocessor group. Since 1999, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison. His research interests include the areas of computer-aided design and microprocessor circuit design with an emphasis on interconnect and circuit optimization as well as signal integrity analysis and optimization.

Dr. Chen received the D2000 Award from Intel Corp. and the National Sciences Foundation Faculty Early Career Development Award (CAREER) in 1999 and 2001, respectively. He also received the 2002 SIGDA/ACM Outstanding Young Faculty Award and 2002 Peter Schneider Faculty Development Award.