3D Thermal-ADI
– An Efficient Chip-Level Transient Thermal Simulator[*]

Ting-Yuan Wang Yu-Min Lee
Electrical and Computer Engineering
University of Wisconsin-Madison
1415 Engineering Drive
Madison, WI 53706
{wangt, yu-min}@cae.wisc.edu

Charlie Chung-Ping Chen[†]
Graduate Institute of Electronics Engineering &
Department of Electrical Engineering
National Taiwan University
Taipei 106, Taiwan
cchen@cc.ee.ntu.edu.tw

ABSTRACT

Recent studies show that the nonuniform thermal distribution on the substrate and interconnects has impact on the circuit reliability and performance. Hence three-dimensional (3-D) thermal analysis is crucial to analyze these effects. In this paper, we present and develop an efficient 3-D transient thermal simulator based on the alternating direction implicit (ADI) method for large scale temperature estimation problems. Our 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 is also highly accurate and memory efficient. The temperature profile of steady state can be reached in few iterations. The software is avaiable on the web [1].

Categories and Subject Descriptors

J.6 [**COMPUTER-AIDED ENGINEERING**]: Computer-aided design (CAD); I.6 [**SIMULATION AND MODELING**]: Applications

General Terms

Algorithms, Design, Performance

Keywords

ADI, automation, design, finite difference methods, thermal simulation, temperature

1. INTRODUCTION

A comprehensive analysis of the thermal effects in high-performance VLSI has been discussed recently [2], [3], [4], [5]. Due to 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. Therefore, 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 vertical distance from the upper surface of the silicon substrate to the top metal level has been simulated [6]. As technology is scaling down, the maximum temperature increases and the temperature gradient between the top metal wires and the silicon substrate becomes larger. This is because the scaling trend, which increases interconnect levels, current density, and thermal coupling as well as the introduction of low-k materials, makes the thermal effects worse.

High temperature not only causes timing failures of the VLSI circuit design but also degrades chip reliability. For example, the temperature-induced logic fault occurs in a 10-bit adder because the large temperature gradient causes the arrival time of an input signal at the 10^{th} bit to become slower than expected [7].

Therefore, how to effectively analyze the chip-level 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 due to the fact that interconnects form a complicated 3-D array. Therefore 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

[*] This work was partially supported by the National Science Foundation under grants CCR-0093309 and CCR-0204468.

[†] This work was done when the author was at University of Wisconsin-Madison.

ISPD'03, April 6–9, 2003, Monterey, California, USA.
Copyright 2003 ACM 1-58113-650-1/03/0004 ...$5.00.

RC model has been presented [7], [8]. 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. A full-chip thermal analysis was presented in [9], but the problem was simplified with function blocks. A two-dimensional full-chip thermal simulation was also presented [10].

In this paper, we propose an efficient transient thermal simulator using the ADI method [11] to simulate the full-chip 3-D temperature profile. Basically the ADI method is an alternative solution method which, instead of solving the 3-D problems, solves a succession of three one-dimensional (1-D) problems as illustrated in Fig. 1. Our simulator, 3D Thermal-ADI, is not only unconditionally stable but also has a linear runtime and memory usage. The temperature profile of steady state can also be reached in few iterations.

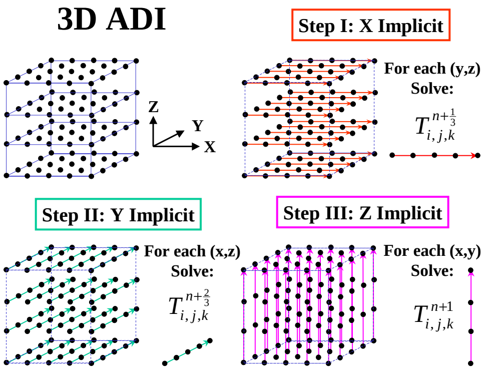

Figure 1: ADI Method.

The remainder of the paper is organized as follows. An overview of the simulator is presented in Section 2. Section 3 discusses the problem formulation using finite difference method. Section 4 deals with the ADI method. The implementation and experimental results are presented in Section 5, followed by the conclusion in Section 6. The proof of unconditional stability of the ADI method is in the appendix.

2. OVERVIEW OF THE SIMULATOR

An overview of the 3D Thermal-ADI simulator will be presented including the modeling strategy, temperature-power relation, and the main flowchart of the simulator.

2.1 Modeling Strategy

Consider a chip with package and heat sinks, e.g., the Motorola PowerPC microprocessors with controlled-collapsed-chip-connection/ceramic-ball-grid-array (C4/CBGA) single-chip package are shown in Fig. 2 [12]. The modeling strategy is illustrated in Fig. 3 with two parts [7]. First, the temperature distribution in a chip containing the interconnects and a portion of the silicon substrate is governed by the following 3-D heat conduction equation [13], and is formulated with the finite difference method alternatively solved by the ADI technique for a high degree of accuracy

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

subject to the thermal boundary conditions

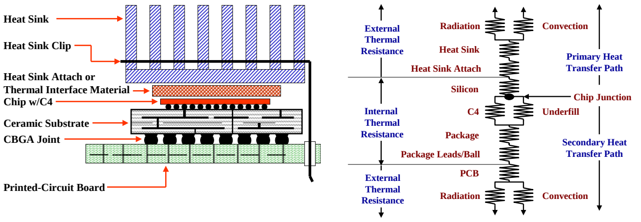

Figure 2: Cross-section view of the PowerPC with C4/CBGA package and heat sinks and the simplified thermal resistance network.

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

where T is the time dependent temperature, 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 $\int \rho C_p \frac{\partial T}{\partial t} dV$. Suppose that the rate of heat conduction through surface $d\vec{A}$ is $q = \kappa d\vec{A} \cdot \triangledown T$. Then the rate of heat flow entering the control volume is $\int q = \int \kappa \triangledown T \cdot d\vec{A} = \int \triangledown \cdot [\kappa \triangledown T]\ dV$. 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.

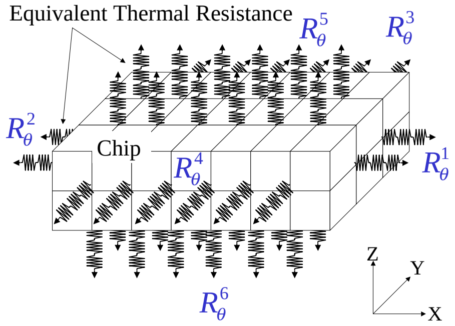

Figure 3: Modeling Strategy of 3D Thermal-ADI simulator.

Second, the complicated package and heat sink structures are modeled as 1-D equivalent thermal resistances network, e.g., Fig. 2. Suppose that the package surfaces are held isothermal. If the package surfaces are not isothermal, 3-D model is needed for better accuracy to include the contribution due to heat spreading within the package. The equivalent thermal resistances, R_θ^i, on the six sides of the chip boundary are applied to model the effective heat transfer coefficient, $h_i^e = \frac{1}{A^i R_\theta^i}$, where A^i is the effective 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 $f_i = h_i^e T_\infty$. Therefore the equation is

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

The finite difference method is used to solve (1), and efforts are made to keep the formulas having second-order accuracy in time and space. Since there is not only one material in the chip, the homogeneous and non-homogeneous cases are also considered.

However, the computational inefficiency due to the large size of the problem and the complexity of the non-homogeneous 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 4. A practical concern for power-temperature relation will be discussed in the next section.

2.2 Mutual Relation of Power-Temperature

The sources of power consumption in VLSI circuits are from devices and interconnects. There are four sources of power consumption from devices, which are switching power, short-circuit power, leakage circuit power, and static power [14]. The source of power consumption in interconnects is from self-heating. 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 at each time step.

Temperature and power are related to each other. In general, the physical parameters, e.g., resistivity and carrier mobility are sensitive to temperatures. This implies that power is sensitive to temperature. For example, the power generated from self-heating which is proportional to resistivity is specially 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. As shown in Fig. 4, 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 the 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.

2.3 Flowchart of the Simulator

The main procedure of the simulator is described as follows and shown in Fig. 4. The geometric information and locations of the interconnects and transistors are given. The simplified problem which gives only the geometry and locations of the function-blocks can also be handled. This information may be extracted from GDSII file or other sources. The corresponding power densities are given from the power estimation tools [15], [16]. The package and cooling system data and the chip size are given in order to apply the boundary conditions. The temperature sensitive physical parameters as well as the technology information are given.

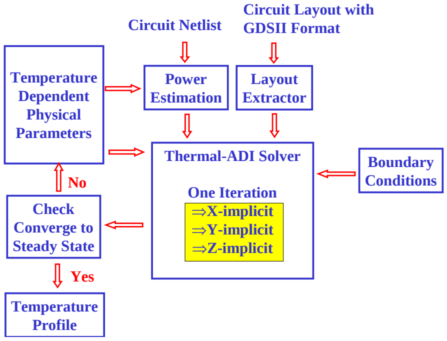

Figure 4: The flowchart of the 3D Thermal-ADI Simulator.

The system initializes the computation by combining all the information, and begins the calculation by the ADI based solver.

Several transient iterations are needed to converge to the thermal steady state. Each transient iteration repeats the following procedure. The ADI based solver calculates the temperature distribution with three alternating direction implicit steps. Then the system checks whether the temperature profile converges to steady state or not. If the results do not converge, the temperature sensitive 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.

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 non-homogeneous cases.

3. 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). For the homogeneous material, the term $\bigtriangledown \cdot [\kappa(\vec{r},T) \bigtriangledown T(\vec{r},t)]$ in (1) can be replaced by $\kappa(T) \bigtriangledown^2 T(\vec{r},t)$. We have

$$\frac{\partial T(\vec{r},t)}{\partial t} = \alpha[\frac{\partial^2 T(\vec{r},t)}{\partial x^2} + \frac{\partial^2 T(\vec{r},t)}{\partial y^2} + \frac{\partial^2 T(\vec{r},t)}{\partial z^2}] + \frac{g(\vec{r},t)}{\rho C_p} \qquad (4)$$

where $\alpha = \frac{\kappa}{\rho C_p}$. This equation is a second-order parabolic partial differential equation.

The procedure to establish a finite-difference method of the partial differential equation is to discretize the continuous space domain and time domain. 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:

$$\left.\frac{\partial^2 T}{\partial x^2}\right|^n_{i,j,k} = \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} + O[(\Delta x)^2] \approx \frac{\delta^2_x T^n}{(\Delta x)^2} \qquad (5)$$

where $\delta^2_x T = T_{i-1,j,k} - 2T_{i,j,k} + T_{i+1,j,k}$. The truncation error

(TR) is $O[(\Delta x)^2]$. Similar process can be applied to 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 sum of the net rate of energy transferring into the control volume and the power generated. Hence on the left-hand side of (4) is the energy stored from time step n to $n+1$ in a control volume. However, the choice of time step on the right-hand side affects the accuracy and stability of discretization [17]. In this paper, the Crank-Nicolson method is used by taking the average of energy at time step n (simple explicit method) and time step $n+1$ (simple implicit method). According to (4) and (5), we have:

$$\begin{aligned}\frac{T^{n+1}_{i,j,k}-T^{n}_{i,j,k}}{\Delta t} &= \alpha[\frac{\delta_x^2 T^{n+1}+\delta_x^2 T^n}{2(\Delta x)^2}+\frac{\delta_y^2 T^{n+1}+\delta_y^2 T^n}{2(\Delta y)^2}\\ &+\frac{\delta_z^2 T^{n+1}+\delta_y^2 T^n}{2(\Delta z)^2}]+\frac{g}{\rho C_p}. \qquad (6)\end{aligned}$$

After rearrangement, we get the difference equation as:

$$\begin{aligned}&-r_x T^{n+1}_{i-1,j,k}-r_y T^{n+1}_{i,j-1,k}-r_z T^{n+1}_{i,j,k-1}+2(1+r_x+r_y+r_z)T^{n+1}_{i,j,k}-r_x T^{n+1}_{i+1,j,k}\\ &\quad -r_y T^{n+1}_{i,j+1,k}-r_z T^{n+1}_{i,j,k+1} = r_x T^{n}_{i-1,j,k}+r_y T^{n}_{i,j-1,k}+r_z T^{n}_{i,j,k-1}+2(1-r_x\\ &\quad -r_y-r_z)T^{n}_{i,j,k}+r_x T^{n}_{i+1,j,k}+r_y T^{n}_{i,j+1,k}+r_z T^{n}_{i,j,k+1}+\frac{2\Delta t}{\rho C_p}g_{i,j,k}, \qquad (7)\end{aligned}$$

where $r_x=\frac{\alpha\Delta t}{(\Delta x)^2}$, $r_y=\frac{\alpha\Delta t}{(\Delta y)^2}$, $r_z=\frac{\alpha\Delta t}{(\Delta z)^2}$ and $i=1,2,\dots,I{-}1$, $j=1,2,\dots,J-1$, $k=1,2,\dots,K-1$. Note that there are seven unknowns in each equation at point (i,j,k) to solve the time step $n+1$. With this choose of discretization, the equation has the best accuracy with $TR=O[(\Delta t)^2,(\Delta x)^2,(\Delta y)^2,(\Delta z)^2]$. This method is unconditionally stable [17].

Basically, there are several different materials in a 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 different materials.

3.1 Non-homogeneous Case

Consider the nodes located between different materials, e.g., the geometry as shown in Fig. 5, 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 the control volume causing the temperature increase is $\frac{dE}{dt}=[\frac{3}{4}\Delta V\rho_1 C_{p1}+\frac{1}{4}\Delta V\rho_2 C_{p2}]\frac{\Delta T_{i,j,k}}{\Delta t}$ where $\Delta V=\Delta x\Delta y\Delta z$ is the control volume. The power generated in the control volume is $g=\frac{3}{4}\Delta V g_1+\frac{1}{4}\Delta V g_2$. The rate of heat conduction from $T_{i,j,k}$ to $T_{i-1,j,k}$ and $T_{i+1,j,k}$ are $q_1=\kappa_1\frac{\Delta y\Delta z}{\Delta x}(T_{i,j,k}-T_{i-1,j,k})$ and $q_2=\frac{\kappa_1+\kappa_2}{2}\frac{\Delta y\Delta z}{\Delta x}(T_{i,j,k}-T_{i+1,j,k})$, respectively. The others can be calculated similarly. Then we obtain $q_3=\frac{3\kappa_1+\kappa_2}{4}\frac{\Delta x\Delta z}{\Delta y}(T_{i,j,k}-T_{i,j-1,k})$, $q_4=\frac{3\kappa_1+\kappa_2}{4}\frac{\Delta x\Delta z}{\Delta y}(T_{i,j,k}-T_{i,j+1,k})$, $q_5=\kappa_1\frac{\Delta x\Delta y}{\Delta z}(T_{i,j,k}-T_{i,j,k-1})$, and $q_6=\frac{\kappa_1+\kappa_2}{2}\frac{\Delta x\Delta y}{\Delta z}(T_{i,j,k}-T_{i,j,k+1})$. From energy conservation, we have

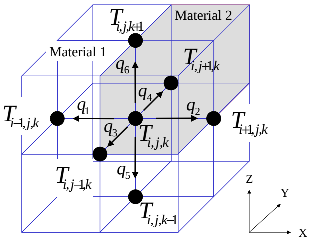

Figure 5: The non-homogeneous discretization example with node $T_{i,j,k}$ on the boundary surface of two different materials.

$$g=\frac{dE}{dt}+q_1+q_2+q_3+q_4+q_5+q_6. \qquad (8)$$

It can be rearranged as follows:

$$\begin{aligned}\frac{\Delta T_{i,j,k}}{\Delta t} &= \frac{1}{\frac{3}{4}\rho_1 C_{p1}+\frac{1}{4}\rho_2 C_{p2}}\{\frac{\frac{\kappa_1+\kappa_2}{2}\frac{T_{i+1,j,k}-T_{i,j,k}}{\Delta x}-\kappa_1\frac{T_{i,j,k}-T_{i-1,j,k}}{\Delta x}}{\Delta x}\\ &+\frac{\frac{3\kappa_1+\kappa_2}{4}\frac{T_{i,j+1,k}-T_{i,j,k}}{\Delta y}-\frac{3\kappa_1+\kappa_2}{4}\frac{T_{i,j,k}-T_{i,j-1,k}}{\Delta y}}{\Delta y}\\ &+\frac{\frac{\kappa_1+\kappa_2}{2}\frac{T_{i,j,k+1}-T_{i,j,k}}{\Delta z}-\kappa_1\frac{T_{i,j,k}-T_{i,j,k-1}}{\Delta z}}{\Delta z}\}+\frac{3g_1+g_2}{3\rho_1 C_{p1}+\rho_2 C_{p2}}. \qquad (9)\end{aligned}$$

This expression implies that it is the same as taking central finite-difference discretization on (1). Therefore, the accuracy of the space domain 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.

4. ADI METHOD

Peaceman and Rachford [18], and Douglas and Gunn [11] developed a variation on the Crank-Nicolson approximation which is known as the ADI method. In this paper, we adopt the Douglas-Gunn technique applied to the thermal problems. The Peaceman-Rachford approach has second-order accuracy and unconditional stability for only two-dimensional problems. It leads to conditionally stable and first-order accuracy in time for 3-D problems.

4.1 Douglas-Gunn Approach

First rewrite (6) as follows:

$$\begin{aligned}T^{n+1}-T^{n} &= r_x\frac{\delta_x^2}{2}(T^{n+1}+T^n)+r_y\frac{\delta_y^2}{2}(T^{n+1}+T^n)\\ &+r_z\frac{\delta_z^2}{2}(T^{n+1}+T^n)+\frac{\Delta t}{\rho C_p}g. \qquad (10)\end{aligned}$$

Instead of directly solving (10) at every time step, we solve the same equations by three sub-time steps at each time march $n\to n+1$.

Step I :

$$\begin{aligned}T^{n+\frac{1}{3}}-T^{n} &= \frac{r_x\delta_x^2}{2}(T^{n+\frac{1}{3}}+T^n)+r_y\delta_y^2 T^n+r_z\delta_z^2 T^n\\ &+\frac{\Delta t}{\rho C_p}g \qquad (11)\end{aligned}$$

Step II :

$$
\begin{aligned}
T^{n+\frac{2}{3}} - T^{n} &= \frac{r_x\delta_x^2}{2}(T^{n+\frac{1}{3}} + T^{n}) + \frac{r_y\delta_y^2}{2}(T^{n+\frac{2}{3}} + T^{n}) \\
&\quad + r_z\delta_z^2 T^{n} + \frac{\Delta t}{\rho C_p} g \qquad (12)
\end{aligned}
$$

Step III :

$$
\begin{aligned}
T^{n+1} - T^{n} &= \frac{r_x\delta_x^2}{2}(T^{n+\frac{1}{3}} + T^{n}) + \frac{r_y\delta_y^2}{2}(T^{n+\frac{2}{3}} + T^{n}) \\
&\quad + \frac{r_z\delta_z^2}{2}(T^{n+1} + T^{n}) + \frac{\Delta t}{\rho C_p} g. \qquad (13)
\end{aligned}
$$

THEOREM 1. *The ADI method in (11)-(13) 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 $n+\frac{1}{3}$, $n+\frac{1}{3}$ to $n+\frac{2}{3}$, and $n+\frac{2}{3}$ to $n+1$ as shown in Fig. 1. 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 (11), the other two equations of the corresponding point are from boundary conditions, which will be discussed in Section 4.2. Since each point in 11 is related to three unknown variables $T^{n+\frac{1}{3}}_{i-1,j,k}$, $T^{n+\frac{1}{3}}_{i,j,k}$, and $T^{n+\frac{1}{3}}_{i+1,j,k}$, there is a tridiagonal system of each (y,z) value. For the grid size $l \times m \times n$ with total node count $N = lmn$, the tridiagonal matrix of each (y,z) is solved by the Thomas Algorithm [17] with time complexity $O(l)$. Similarly, it applies to *Step II* and *Step III*. Therefore the total time complexity is $O(itr \times 3 \times lmn) = O(N)$, where itr is the number of transient iterations.

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

Step I :

$$
\begin{aligned}
&-r_x T^{n+\frac{1}{3}}_{i-1,j,k} + 2(1+r_x)T^{n+\frac{1}{3}}_{i,j,k} - r_x T^{n+\frac{1}{3}}_{i+1,j,k} = \\
&\quad \{\, r_x T^{n}_{i-1,j,k} + 2r_y T^{n}_{i,j-1,k} + 2r_z T^{n}_{i,j,k-1} + 2(1-r_x-2r_y-2r_z)T^{n}_{i,j,k} \\
&\quad + r_x T^{n}_{i+1,j,k} + 2r_y T^{n}_{i,j+1,k} + 2r_z T^{n}_{i,j,k+1}\} + \frac{2\Delta t}{\rho C_p} g_{i,j,k} \qquad (14)
\end{aligned}
$$

Step II :

$$
\begin{aligned}
&-r_y T^{n+\frac{2}{3}}_{i,j-1,k} + 2(1+r_y)T^{n+\frac{2}{3}}_{i,j,k} - r_y T^{n+\frac{2}{3}}_{i,j+1,k} = \\
&\quad \{\, r_x T^{n+\frac{1}{3}}_{i-1,j,k} - 2r_x T^{n+\frac{1}{3}}_{i,j,k} + r_x T^{n+\frac{1}{3}}_{i+1,j,k}\} + \{\, r_x T^{n}_{i-1,j,k} + r_y T^{n}_{i,j-1,k} \\
&\quad + 2r_z T^{n}_{i,j,k-1} + 2(1-r_x-r_y-2r_z)T^{n}_{i,j,k} + r_x T^{n}_{i+1,j,k} + r_y T^{n}_{i,j+1,k} \\
&\quad + 2r_z T^{n}_{i,j,k+1}\} + \frac{2\Delta t}{\rho C_p} g_{i,j,k} \qquad (15)
\end{aligned}
$$

Step III :

$$
\begin{aligned}
&-r_z T^{n+1}_{i,j,k-1} + 2(1+r_z)T^{n+1}_{i,j,k} - r_z T^{n+1}_{i,j,k+1} = \\
&\quad \{r_x T^{n+\frac{1}{3}}_{i-1,j,k} - 2r_x T^{n+\frac{1}{3}}_{i,j,k} + r_x T^{n+\frac{1}{3}}_{i+1,j,k}\} + \{r_y T^{n+\frac{2}{3}}_{i,j-1,k} - 2r_y T^{n+\frac{2}{3}}_{i,j,k} \\
&\quad + r_y T^{n+\frac{2}{3}}_{i,j+1,k}\} + \{r_x T^{n}_{i-1,j,k} + r_y T^{n}_{i,j-1,k} + r_z T^{n}_{i,j,k-1} + 2(1-r_x-r_y \\
&\quad - r_z)T^{n}_{i,j,k} + r_x T^{n}_{i+1,j,k} + r_y T^{n}_{i,j+1,k} + r_z T^{n}_{i,j,k+1}\} + \frac{2\Delta t}{\rho C_p} g_{i,j,k}. \qquad (16)
\end{aligned}
$$

In *Step I* as shown in (14), there are $I+1$ equations for each (y,z) value. Also there are three unknown variables $T^{n+\frac{1}{3}}_{i-1,j,k}$, $T^{n+\frac{1}{3}}_{i,j,k}$, and $T^{n+\frac{1}{3}}_{i+1,j,k}$ for each equation. Then this can be expressed as a tridiagonal system:

$$
\begin{bmatrix}
\diamond & \diamond & & & & & \\
\star & \star & \star & & & & \\
& \star & \star & \star & & & \\
& & \star & \star & \star & & \\
& & & & \vdots & & \\
& & & & \star & \star & \star \\
& & & & & \diamond & \diamond
\end{bmatrix}
\begin{bmatrix}
T^{n+\frac{1}{3}}_{0,j,k} \\
T^{n+\frac{1}{3}}_{1,j,k} \\
\vdots \\
T^{n+\frac{1}{3}}_{I-1,j,k} \\
T^{n+\frac{1}{3}}_{I,j,k}
\end{bmatrix}
=
\begin{bmatrix}
b_0 \\ b_1 \\ \vdots \\ b_{I-1} \\ b_I
\end{bmatrix}
\qquad (17)
$$

where b_i are the values on the right-hand side of the equal sign in (14). Note that the coefficients denoted with diamonds will be determined by the boundary conditions. For the non-homogeneous cases, the difference equations of the ADI method can be derived with the same rule.

4.2 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 2.1, the complicated package and heat sinks can be modeled by (3). Suppose that we have the following boundary conditions in x-direction:

$$
\begin{aligned}
-\kappa\frac{\partial T}{\partial x} + h^e_{x-}T &= h^e_{x-}T_\infty \qquad \textit{at surface} \;\; x = 0 \\
\kappa\frac{\partial T}{\partial x} + h^e_{x+}T &= h^e_{x+}T_\infty \qquad \textit{at surface} \;\; x = L \qquad (18)
\end{aligned}
$$

where h^e_{x-} and h^e_{x+} are the effective heat transfer coefficients calculated from the equivalent thermal resistance on the boundary $x = 0$ and $x = L$, 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 $T^n_{-1,j,k}$ and $T^n_{I+1,j,k}$ by expanding the distance Δx 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 (18), we have

$$
-\kappa\frac{T^n_{1,j,k} - T^n_{-1,j,k}}{2\Delta x} + h^e_{x-}T^n_{0,j,k} = h^e_{x-}T_\infty \quad at \quad i = 0. \qquad (19)
$$

Thus the virtual point can be expressed as:

$$
T^n_{-1,j,k} = T^n_{1,j,k} + \frac{2\Delta x h^e_{x-}}{\kappa}(T_\infty - T^n_{0,j,k}). \qquad (20)
$$

The other virtual points 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. 6. The energy generates in the control volume is $G = \frac{1}{8}\Delta V g$. The rate of energy stored causing the temperature increase is $\frac{dE}{dt} = \frac{1}{8}\Delta V \rho_1 C_{p1}\frac{\Delta T_{0,0,0}}{\Delta t}$. The rates of heat transfer are $q_1 = h^e_{x-}\frac{\Delta y\Delta z}{4}(T_{0,0,0} - T_\infty)$, $q_2 = \kappa_1\frac{\Delta y\Delta z}{4\Delta x}(T_{0,0,0} - T_{1,0,0})$, $q_3 = h^e_{y-}\frac{\Delta x\Delta z}{4}(T_{0,0,0} - T_\infty)$, $q_4 = \kappa_1\frac{\Delta x\Delta z}{4\Delta y}(T_{0,0,0} - T_{0,1,0})$, $q_5 = h^e_{z-}\frac{\Delta x\Delta y}{4}(T_{0,0,0} - T_\infty)$, and $q_6 = \kappa_1\frac{\Delta x\Delta y}{4\Delta z}(T_{0,0,0} - T_{0,0,1})$. Then the difference equation of the Crank-Nicolson method derived by energy conservation at point $(0,0,0)$ is:

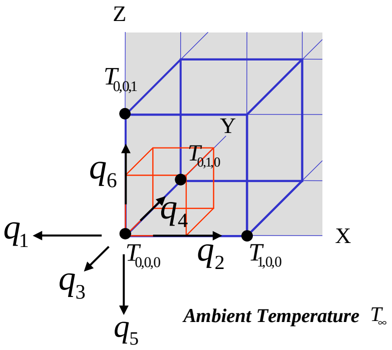

Figure 6: The boundary conditions for node at point $(0,0,0)$.

$$G = \frac{dE}{dt} + q_1 + q_2 + q_3 + q_4 + q_5 + q_6. \tag{21}$$

After rearranging, we have

$$\frac{1}{4}(1+r_x\beta_{x-}+r_y\beta_{y-}+r_z\beta_{z-})T^{n+1}_{0,0,0}-\frac{r_x}{4}T^{n+1}_{1,0,0}-\frac{r_y}{4}T^{n+1}_{0,1,0}-\frac{r_z}{4}T^{n+1}_{0,0,1} = \frac{1}{4}(1-r_x\beta_{x-}-r_y\beta_{y-}-r_z\beta_{z-})T^{n}_{0,0,0}+\frac{r_x}{4}T^{n}_{1,0,0}+\frac{r_y}{4}T^{n}_{0,1,0}+\frac{r_z}{4}T^{n}_{0,0,1}+ \frac{1}{2}(r_x r_{hx-}+r_y r_{hy-}+r_z r_{hz-})+\frac{\Delta t}{4\rho C_p}g_{0,0,0} \tag{22}$$

where $r_{hx-} = \frac{h^e_{x-}\Delta x T_\infty}{\kappa}$, $r_{hy-} = \frac{h^e_{y-}\Delta y T_\infty}{\kappa}$, $r_{hz-} = \frac{h^e_{z-}\Delta z T_\infty}{\kappa}$, $\beta_{x-} = 1+\frac{h^e_{x-}\Delta x}{\kappa}$, $\beta_{y-} = 1+\frac{h^e_{y-}\Delta y}{\kappa}$, $\beta_{z-} = 1+\frac{h^e_{z-}\Delta z}{\kappa}$. 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 (7). Note that the coefficients in (22) are very important to maintain the symmetry of the matrix when solving the problem. If both sides of (22) 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. All 26 types can be derived with the same way. Furthermore, the difference equations of the ADI method can similarly be derived by substituting virtual points for each step from (14) to (16).

5. 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 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. 7. Note that the scale in y axis is a logarithm on the top. As can be seen in Fig. 7, the runtime of 3D Thermal-ADI is linearly proportional to the number of the grid nodes. 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. 8. 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 Crank-Nicolson method increases quickly, which limits the size of the problem which can be solved on a given machine.

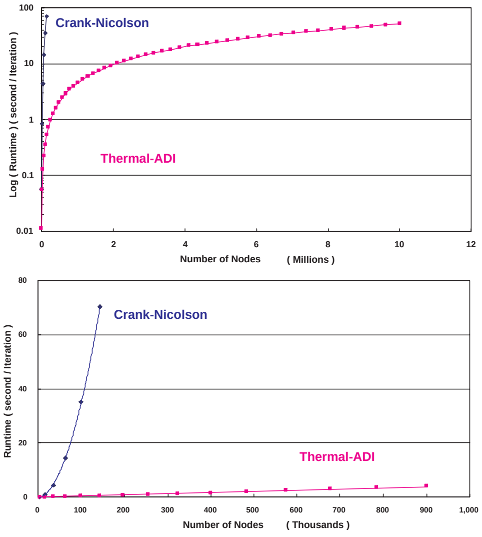

Figure 7: Runtime comparison of the simulations with the Crank-Nicolson and the 3D Thermal-ADI approaches.

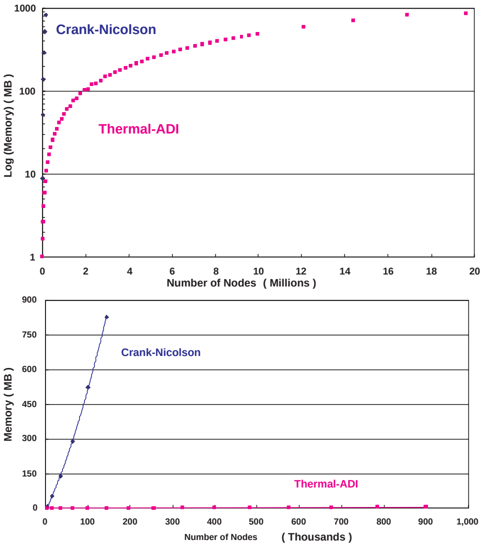

Figure 8: Comparison of the memory usage of the simulations with the Crank-Nicolson and the 3D Thermal-ADI approaches.

Consider a chip with size 11.3 $mm \times 14.4$ mm. The power in each function block is shown in Fig. 9. The discretization size is 565 × 720 × 7 with $\Delta x = 20$ μm, $\Delta y = 20$ μm, and $\Delta z = 20$ μm. Here we only consider the volume near the substrate surface, and the substrate is included in the equivalent thermal resistance. The effective heat transfer

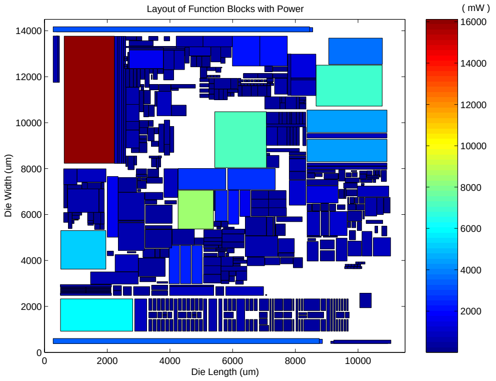

Figure 9: The illustration of the power in each function block.

coefficients, h_i^e, are supposed to be 8×10^4 W/m^2K in the primary heat transfer path and 7×10^3 W/m^2K in the secondary heat transfer path. The simulation runs 1200 iterations with time increment $\Delta t = 10^{-4}$ s. The runtime is about 311 min. The results of the temperature profile are shown in Fig. 10. The highest temperature is about 180 ^oC 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 [19].

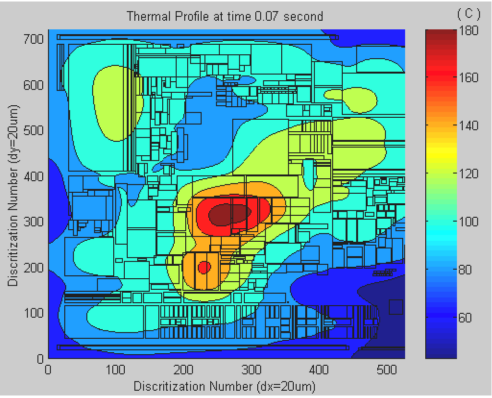

Figure 10: Thermal Profile at $t = 0.07$ s.

The transient temperature profile of the average temperature is shown in Fig. 11. From the figure we can find that the average temperature reaches steady state at time 0.07 s or at iteration 700. The temperature difference between time 0.07 s 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 Crank-Nicolson method, there is a criterion for the maximum Δt [20]. 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:

$$\Delta t_c = \min\ \{\frac{1}{2((1-s)\alpha_1 + s\alpha_2)}[\frac{1}{\frac{b_x}{\Delta x^2} + \frac{b_y}{\Delta y^2} + \frac{b_z}{\Delta z^2}}]\}_{(i,j,k)} \quad (23)$$

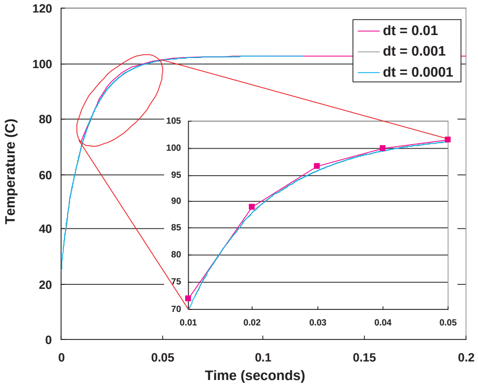

Figure 11: The transient results for the average temperature with $\Delta t = 10^{-4}s$, $\Delta t = 10^{-3}s$, and $\Delta t = 10^{-2}s$.

where s depends on the location of the point $T_{i,j,k}$ as discussed in the non-homogeneous 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 (23) is as follows. Calculate the value of each discretization point $T_{i,j,k}$ on the right-hand side of (23), then find the minimum value which is the criterion for the maximum Δt among all points. For example, the case shown in Fig. 5 has one quarter of material 2 and three quarters of material 1. Therefore the value of s is $\frac{1}{4}$. We have $b_x = \frac{1}{2}(1+\beta_{\pm x})$, $b_y = \frac{1}{2}(1+\beta_{\pm y})$, and $b_z = \frac{1}{2}(1+\beta_{\pm z})$ 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. 6. 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 $b_x = \frac{1}{2}(1+\beta_{+x})$, $b_y = \frac{1}{2}(1+\beta_{+y})$, and $b_z = \frac{1}{2}(1+\beta_{+z})$. The value of Δt_c in this example is 8.293×10^{-7} s.

For the problem solved by the Thermal-ADI method, there is no such restriction. The transient results for the average temperature with $\Delta t = 10^{-4}$ s, $\Delta t = 10^{-3}$ s, and $\Delta t = 10^{-2}$ s are shown in Fig. 11. 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 seven iterations to reach the thermal time constant. 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. 11, there are deviations between the transient thermal 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 period. 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. 12. Even though we can not know the transient results because of the oscillation, the results still converge to steady state in several iterations.

6. CONCLUSION

In this paper, an efficient transient 3-D thermal simula-

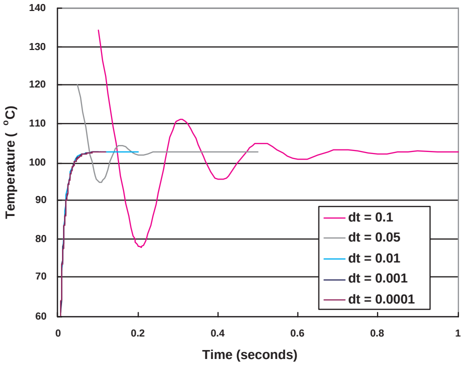

Figure 12: The transient results for the average temperature with $\Delta t = 10^{-4}s$, $\Delta t = 10^{-3}s$, $\Delta t = 5 \times 10^{-2}s$, $\Delta t = 10^{-2}s$, and $\Delta t = 0.1s$.

tor based on the ADI method, 3D Thermal-ADI, has been presented and developed for the analyses of VLSI chip thermal distribution. Detailed discussions of the 3D 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 few 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.

7. APPENDIX

In this section the unconditional stability of the ADI method in (11)-(13) is proved.

Proof: After rearranging (11)-(13), we have the following form

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

where

$$\Delta T^{n+\frac{1}{3}} \equiv T^{n+\frac{1}{3}} - T^n \quad \Delta T^{n+\frac{2}{3}} \equiv T^{n+\frac{2}{3}} - T^n \quad \Delta T^{n+1} \equiv T^{n+1} - T^n$$

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

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

Solving the equations for $\hat{T}^{n+1}_{i,j,k}$ as a function of $\hat{T}^n_{i,j,k}$

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

In the above equation, we have

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

Therefore, the ADI method is unconditionally stable.

8. REFERENCES

[1] http://vlsi.ece.wisc.edu/3d_thermal_adi.htm.

[2] Kaustav Banerjee, Massound Pedram, and Amir H. Ajami. Analysis and optimization of thermal issues in high-performance vlsi. In *ACM/SIGDA International Symposium on Physical Design (ISPD)*, pages 230–237, April 2001.

[3] Kaustav Banerjee and Amit Mehrotra. Global (interconnect) warming. *IEEE Circuits and Devices Magazine*, 17(5):16–32, September 2001.

[4] Kaustav Banerjee, Amit Mehrotra, Alberto Sangiovanni-Vincentelli, and Chenming Hu. On thermal effects in deep sub-micron vlsi interconnects. In *36th ACM/IEEE Design Automation Conference*, pages 885–891, 1999.

[5] Vladimir Székely. Tracing the thermal behavior of ics. *IEEE Design and Test of Computers*, 15(2):14–21, April-June 1998.

[6] Sungjun Im and Kaustav Banerjee. Full chip thermal analysis of planar (2-d) and vertically integrated (3-d) high performance ics. In *Technical Digest IEEE International Electron Devices Meeting (IEDM)*, pages 727–730, San Francisco, CA, December 2000.

[7] Yi-Kan Cheng, Prasun Raha, Chin-Chi Teng, Elyse Rosenbaum, and Sung-Mo Kang. Illiads-t: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of cmos vlsi chips. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 17(8):668–681, Aug. 1998.

[8] Danqing Chen, Erhong Li, Elyse Rosenbaum, and Sung-Mo Kang. Interconect thermal modeling for accurate simulation of circuit timing and reliability. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 19(2):197–205, Feb. 2000.

[9] Zhiping Yu, Dan Yergeau, and Robert W. Dutton. Full chip thermal simulation. In *International Symposium on Quality Electronic Design, 2000*, pages 145–149, March 2000.

[10] Ting-Yuan Wang and Charlie Chung-Ping Chen. Thermal-adi: a linear-time chip-level dynamic thermal simulation algorithm based on alternating-direction-implicit (adi) method. In *Proceedings of the 2001 international symposium on Physical design*, pages 238–243, April 2001.

[11] Jr. Jim Douglas and James E. Gunn. A general formulation of alternating direction methods–part i. parabolic and hyperbolic problems. *Numerische Mathematik*, 6:428–453, 1964.

[12] John Parry, Harvey Rosten, and Gray 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 Transactions on Components, Packaging, and Manufacturing Technology, Part A*, 21(1):104–112, March 1998.

[13] M. Necati Ozisik. *Boundary Value Problems of Heat Conduction*. Dover Publications Inc., 1968.

[14] Anatha P. Chandrakasan and Robert W. Brodersen. *Low Power Digital CMOS Design*. Kluwer Academic Publishers, 1995.

[15] Li-Pen Yuan, Chih-Chi Teng, and Sung-Mo Kang. Statistical estimation of average power dissipation. In *34th ACM/IEEE Design Automation Conference*, 1997.

[16] R. Burch, F. N. Najm, P. Yang, and T. N. Trick. A monte carlo approach for power estimation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1(1):63–71, March 1993.

[17] M. Necati Ozisik. *Finite Difference Methods in Heat Transfer*. CRC Press, 1994.

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

[19] Bray B. Kromann. Thermal management of a c4/ceramic-ball-grid array: the motorola powerpc 603 and powerpc 604 risc microprocessors. In *Semiconductor Thermal Measurement and Management Symposium*, pages 36 –42, 1996.

[20] Glen E. Myers. *Analytical Methods in Conduction Heat Transfer*. AMCHT Publications, second edition, 1987.