Fast alternating direction implicit method for efficient transient thermal simulation of integrated circuits

Heh, Ding Yu; Tan, Eng Leong; Tay, Wei Choon

2015

Heh, D. Y., Tan, E. L., & Tay, W. C. (2015). Fast alternating direction implicit method for efficient transient thermal simulation of integrated circuits. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 29(1), 93-108.
doi:10.1002/jnm.2049

<https://hdl.handle.net/10356/137201>

<https://doi.org/10.1002/jnm.2049>

This is the peer reviewed version of the following article: Heh, D. Y., Tan, E. L., & Tay, W. C. (2015). Fast alternating direction implicit method for efficient transient thermal simulation of integrated circuits. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 29(1), 93-108, which has been published in final form at <https://dx.doi.org/10.1002/jnm.2049>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

Fast Alternating Direction Implicit Method for Efficient Transient Thermal Simulation of Integrated Circuits

Ding Yu Heh^{*,†}, Eng Leong Tan and Wei Choon Tay

Nanyang Technological University, School of EEE, 50 Nanyang Avenue, Singapore 639798

SUMMARY

This paper presents fast alternating direction implicit (FADI) method for efficient transient thermal simulation of integrated circuits. The FADI method is formulated from Peaceman-Rachford's ADI (PR-ADI) and Douglas-Gunn's ADI (DG-ADI) methods. The update procedure of the proposed method has basic implicit form which features derivative-free RHS and hence, better efficiency and conciseness. Subsequently, through the basic implicit form of FADI method, the relationship between classical PR-ADI and DG-ADI methods can be clarified and elucidated in detail. A unified boundary condition that can cater to common kinds of boundary conditions in thermal simulation is also introduced. To further accelerate FADI method, the graphics processing unit (GPU) is also utilized through Compute Unified Device Architecture (CUDA) implementation. It is shown that high efficiency gain can be achieved using the proposed FADI method through large time step size and data parallelism, while maintaining stability and good accuracy. As numerical illustration, an integrated circuit structure with microchannel cooling is demonstrated. Numerical results further ascertain the cooling effect of the microchannels.

Copyright © 2014 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Alternating direction implicit (ADI), finite-difference methods, implicit methods, heat transfer equation, thermal simulation, integrated circuits

1. INTRODUCTION

As technology increasingly pushes for higher speed and performance, power and packaging densities rise significantly. Since the power dissipated is directly translated into heat, the processors are getting hotter. High temperature causes the transistors to slow down due to the degradation of carrier mobility. The interconnect metal also results in higher resistivity, causing longer interconnect RC delays. All these will translate into degradation in performance for the otherwise high performance chips, while the reliability of the chip will also suffer as increased temperature will exponentially decrease the lifespan of the chip. Furthermore, layer by layer thermal floorplanning [1]–[4] has become increasingly important through proper placement of functional blocks for optimum heat distribution. In addition, as integrated circuit (IC) chips become smaller in size, power consumption increases significantly, which in turn causes the chips to become hotter. Rising temperature reduces the performance of the IC chips and therefore, growing attention has also been put into cooling of thermal problems. While heat dissipation becomes increasingly challenging with the stacking of multi-layer IC chips, conventional heat sink design faces difficulty in dissipating the heat to maintain relatively moderate temperature level across the chip. To solve this problem, one

^{*}Correspondence to: Nanyang Technological University, School of EEE, 50 Nanyang Avenue, Singapore 639798

[†]dyheh@ntu.edu.sg

promising solution for the cooling of chip is through on-chip integrated microchannel cooling [5]-[7].

In the early stage of temperature-aware design, a fast and reasonably accurate temperature prediction is desired. Thus, it is important for chip-level thermal analysis to effectively analyze the thermal distribution, not only for steady state but also for transient state. The finite-difference method [8]-[13] is a very flexible approach for handling complex structures. Its time domain stepping scheme also allows the transient temperature profile to be captured and analyzed effectively. Among them, the scheme based on alternating direction implicit (ADI) method [12], [13] is useful as it permits the chosen time step beyond the stability limit of the explicit finite difference method. By using the ADI method, a reasonable temperature prediction can be obtained using a large time step size, thus reducing the computation time. The ADI methods are generally based on the early works of Peaceman and Rachford's ADI (PR-ADI) [14] as well as that of Douglas and Gunn's ADI (DG-ADI) [15]. Nonetheless, these two ADI methods come at the expense of increased complexity in its implementation. Besides having to solve the tridiagonal system of equations, there are substantial arithmetic operations involved on the right hand side (RHS) of the update equations, and large amount of memory indexing operations incurred. These in turn increase the programming complexity and the CPU computation time. Furthermore, these ADI methods [12]-[15] generally include only conduction, while excluding convection due to fluid flow as a mode of heat transfer. They are thus unable to model microchannel fluid motion. Although there are other numerical methods that can deal with convection of fluid flow, they may involve large sparse matrix, preconditioner, iterative and neural network techniques, etc, and are not directly related to ADI method. Hence, these methods are omitted and beyond the scope of this paper.

In this paper, we shall present fast alternating direction implicit method for efficient transient thermal simulation of integrated circuits. The heat transfer equation first takes into account both conduction and convection due to fluid flow. The FADI method is then formulated from DG-ADI and PR-ADI methods, which will feature basic implicit form having derivative-free RHS. While both DG-ADI and PR-ADI methods have been in existence for quite some time, there actually exist subtle differences between these two methods, and their relationships still remain unclear. It will be shown that through the basic implicit form of FADI method, the relationship between classical PR-ADI and DG-ADI methods can be clarified and elucidated in detail. To further accelerate FADI method, the graphics processing unit (GPU) will also be utilized through Compute Unified Device Architecture (CUDA) implementation. It will shown that high efficiency gain can be achieved using the proposed FADI method through large time step size and data parallelism, while maintaining stability and good accuracy. As numerical illustration, an integrated circuit structure with microchannel cooling will be demonstrated. Numerical results will further ascertain the cooling effect of the microchannels.

2. HEAT TRANSFER EQUATION AND ALTERNATING DIRECTION IMPLICIT (ADI) METHOD

The heat transfer equation including both conduction and convection due to incompressible fluid flow can be derived as follows [16]:

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) - \rho C_p \mathbf{u} \cdot \nabla T + g \quad (1)$$

where T is the time-dependent temperature at any point, κ is the thermal conductivity (W/m·K), ρ is the density of the material (kg/m³), C_p is the specific heat capacity (J/kg·K), $\mathbf{u}$ is the fluid velocity vector (m/s) and g is the heat energy generation rate (W/m³). The material parameters here are assumed to be temperature-independent as the temperature variation considered is not great or extreme.

2.1. Peaceman and Rachford's ADI Method

Based on Peaceman and Rachford's ADI (PR-ADI) method [14], (1) can be solved in two successive substeps as

$$\begin{aligned} T^{n+\frac{1}{2}} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{2} \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \\ = T^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T^n}{\partial y} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (2a)$$

$$\begin{aligned} T^{n+1} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T^{n+1}}{\partial y} \\ = T^{n+\frac{1}{2}} + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (2b)$$

It should be noted that (2) has been extended to include convection term in the heat transfer equation. By applying finite-differencing to the spatial derivatives accordingly along with appropriate boundary conditions, the left hand side (LHS) of each substep results in tridiagonal matrix to be solved implicitly. The PR-ADI method has second-order temporal accuracy.

2.2. Douglas and Gunn's ADI Method

Equation (1) can also be solved in two successive substeps through another scheme, known as Douglas and Gunn's ADI (DG-ADI) method [15]. In DG-ADI method, the two successive substeps are

$$\begin{aligned} T^{n+\frac{1}{2}} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{2} \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \\ = T^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^n}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T^n}{\partial x} + \frac{\Delta t}{\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T^n}{\partial y} \right) - \Delta t u_y \frac{\partial T^n}{\partial y} + \frac{\Delta t}{\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (3a)$$

$$\begin{aligned} T^{n+1} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T^{n+1}}{\partial y} \\ = T^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^n}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T^n}{\partial x} + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T^n}{\partial y} \\ + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (3b)$$

Similarly, applying finite-differencing to the spatial derivatives accordingly along with appropriate boundary conditions results in tridiagonal matrix on the LHS of each substep. The DG-ADI method also has second-order temporal accuracy.

3. FAST ALTERNATING DIRECTION IMPLICIT (FADI) METHOD

One can see from (2) and (3) that the right hand side (RHS) of update procedure consists of many spatial derivatives. Applying finite-differencing to these spatial derivatives may lead to considerable arithmetic operations (flops) count and memory indexing, which in turn results in lower algorithm efficiency. Hereby, we propose a fast alternating direction implicit (FADI) method which seeks to reduce the update procedure to basic implicit form to feature derivative-free RHS with better efficiency and conciseness. The original FADI concept is extended from [17] within the electromagnetics community. Furthermore, through the basic implicit form of FADI method, the relationship between PR-ADI and DG-ADI methods can be clarified and elucidated in detail.

3.1. Formulations

We now reduce the update procedure of PR-ADI method in (2) into FADI method. Exploiting the use of auxiliary variable $\tilde{V}$, (2) is first rewritten as

$$\tilde{V}^n = T_{PR}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{PR}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{PR}^n}{\partial y} \quad (4a)$$

$$T_{PR}^{n+\frac{1}{2}} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{2} \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} = \tilde{V}^n + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \quad (4b)$$

$$\tilde{V}^{n+\frac{1}{2}} = T_{PR}^{n+\frac{1}{2}} + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \quad (4c)$$

$$T_{PR}^{n+1} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{PR}^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{PR}^{n+1}}{\partial y} = \tilde{V}^{n+\frac{1}{2}} \quad (4d)$$

where T_{PR} is defined here as the temperature variable of PR-ADI method. Based on (4d) at one time step backward

$$\tilde{V}^{n-\frac{1}{2}} = T_{PR}^n - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{PR}^n}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{PR}^n}{\partial y}, \quad (5)$$

it follows that (4a) is reducible to

$$\begin{aligned} \tilde{V}^n &= T_{PR}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{PR}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{PR}^n}{\partial y} \\ &= 2T_{PR}^n - \left[T_{PR}^n - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{PR}^n}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{PR}^n}{\partial y} \right] \\ &= 2T_{PR}^n - \tilde{V}^{n-\frac{1}{2}}. \end{aligned} \quad (6)$$

Similarly, using (4b), (4c) is also reducible to

$$\begin{aligned} \tilde{V}^{n+\frac{1}{2}} &= T_{PR}^{n+\frac{1}{2}} + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \\ &= 2T_{PR}^{n+\frac{1}{2}} - \left[T_{PR}^{n+\frac{1}{2}} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{2} \frac{\partial T_{PR}^{n+\frac{1}{2}}}{\partial x} - \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \right] \\ &= 2T_{PR}^{n+\frac{1}{2}} - \tilde{V}^n. \end{aligned} \quad (7)$$

Through redefinition of auxiliary variable $V = \frac{1}{2}\tilde{V}$, the update procedure of the FADI method can be obtained sequentially from (6), (4b), (7) and (4d) as

$$V^n = T_F^n - V^{n-\frac{1}{2}} \quad (8a)$$

$$\frac{1}{2}T_F^{n+\frac{1}{2}} - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_F^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{4} \frac{\partial T_F^{n+\frac{1}{2}}}{\partial x} = V^n + \frac{\Delta t}{4\rho C_p} g^{n+\frac{1}{2}} \quad (8b)$$

$$V^{n+\frac{1}{2}} = T_F^{n+\frac{1}{2}} - V^n \quad (8c)$$

$$\frac{1}{2}T_F^{n+1} - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_F^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{4} \frac{\partial T_F^{n+1}}{\partial y} = V^{n+\frac{1}{2}} \quad (8d)$$

with initialization

$$V^{-\frac{1}{2}} = \frac{1}{2}T_F^0 - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_F^0}{\partial y} \right) + \frac{\Delta t u_y}{4} \frac{\partial T_F^0}{\partial y}. \quad (9)$$

T_F now represents the temperature variable of FADI method and V serves as auxiliary variable. It should be emphasized now that the RHS of (8) has been free of spatial derivatives completely, which leads to more efficiency and conciseness. Also, it is interesting to see that the source term is only required once [18] as opposed to twice in (2).

Note that the relationship of temperature variables between PR-ADI and FADI methods remain unchanged at all time indices, i.e.

$$T_F^n = T_{PR}^n \quad (10a)$$

$$T_F^{n+\frac{1}{2}} = T_{PR}^{n+\frac{1}{2}} \quad (10b)$$

$$T_F^{n+1} = T_{PR}^{n+1} \quad (10c)$$

Next, we proceed to reduce the update procedure of DG-ADI method in (3). For the first substep, we again define an auxiliary variable $\tilde{V}^n$ similar to that of (4a) as

$$\tilde{V}^n = T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y}. \quad (11)$$

where T_{DG} is now defined as the temperature variable of DG-ADI method. Using (11), (3a) is rewritten as

$$\begin{aligned} T_{DG}^{n+\frac{1}{2}} &= T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{2} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \\ &= T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{DG}^n}{\partial x} + \frac{\Delta t}{\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) - \Delta t u_y \frac{\partial T_{DG}^n}{\partial y} + \frac{\Delta t}{\rho C_p} g^{n+\frac{1}{2}} \\ &= 2\tilde{V}^n - T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{DG}^n}{\partial x} + \frac{\Delta t}{\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (12)$$

For the second substep, $\tilde{V}^{n+\frac{1}{2}}$ is defined to be the RHS of (3b) as

$$\begin{aligned} \tilde{V}^{n+\frac{1}{2}} &= T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{DG}^n}{\partial x} + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y} \\ &\quad + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{2} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (13)$$

and hence (3b) is rewritten as

$$T_{DG}^{n+1} - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^{n+1}}{\partial y} = \tilde{V}^{n+\frac{1}{2}}. \quad (14)$$

We now wish to remove the spatial derivative in y -direction from (13). To do this, (3a) is first manipulated into

$$\begin{aligned} &\frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y} \\ &= \frac{1}{2} T_{DG}^{n+\frac{1}{2}} - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \\ &\quad - \frac{1}{2} T_{DG}^n - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) + \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^n}{\partial x} - \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (15)$$

Substituting (15) into (13), $\tilde{V}^{n+\frac{1}{2}}$ is now

$$\begin{aligned} \tilde{V}^{n+\frac{1}{2}} &= \frac{1}{2} T_{DG}^n + \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) - \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^n}{\partial x} \\ &\quad + \frac{1}{2} T_{DG}^{n+\frac{1}{2}} + \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}}. \end{aligned} \quad (16)$$

Note that more manipulations and simplifications are required here for DG-ADI method due to presence of spatial derivatives in both x and y -directions on the RHS of each substep.

Based on (14) at one time step backward,

$$\tilde{V}^{n-\frac{1}{2}} = T_{DG}^n - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y}, \quad (17)$$

(11) can be reduced to

$$\begin{aligned} \tilde{V}^n &= T_{DG}^n + \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) - \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y} \\ &= 2T_{DG}^n - \left[T_{DG}^n - \frac{\Delta t}{2\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^n}{\partial y} \right) + \frac{\Delta t u_y}{2} \frac{\partial T_{DG}^n}{\partial y} \right] \\ &= 2T_{DG}^n - \tilde{V}^{n-\frac{1}{2}}. \end{aligned} \quad (18)$$

Similarly, using (12), (16) is also reducible to

$$\begin{aligned} \tilde{V}^{n+\frac{1}{2}} &= \frac{1}{2} T_{DG}^n + \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) - \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^n}{\partial x} \\ &\quad + \frac{1}{2} T_{DG}^{n+\frac{1}{2}} + \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) - \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} + \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \\ &= T_{DG}^{n+\frac{1}{2}} + T_{DG}^n - \left[\frac{1}{2} T_{DG}^{n+\frac{1}{2}} - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right) + \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^{n+\frac{1}{2}}}{\partial x} \right. \\ &\quad \left. + \frac{1}{2} T_{DG}^n - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial T_{DG}^n}{\partial x} \right) + \frac{\Delta t u_x}{4} \frac{\partial T_{DG}^n}{\partial x} - \frac{\Delta t}{2\rho C_p} g^{n+\frac{1}{2}} \right] \\ &= T_{DG}^{n+\frac{1}{2}} + T_{DG}^n - \tilde{V}^n. \end{aligned} \quad (19)$$

Setting $V = \frac{1}{2} \tilde{V}$ and grouping the like terms in (12), (19), we obtain the following update procedure sequentially from (18), (12), (19) and (14) as

$$V^n = T_{DG}^n - V^{n-\frac{1}{2}} \quad (20a)$$

$$\begin{aligned} \frac{1}{4} \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right) - \frac{\Delta t}{8\rho C_p} \frac{\partial}{\partial x} \left(\kappa \frac{\partial \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right)}{\partial x} \right) + \frac{\Delta t u_x}{8} \frac{\partial \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right)}{\partial x} \\ = V^n + \frac{\Delta t}{4\rho C_p} g^{n+\frac{1}{2}} \end{aligned} \quad (20b)$$

$$V^{n+\frac{1}{2}} = \frac{1}{2} \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right) - V^n \quad (20c)$$

$$\frac{1}{2} T_{DG}^{n+1} - \frac{\Delta t}{4\rho C_p} \frac{\partial}{\partial y} \left(\kappa \frac{\partial T_{DG}^{n+1}}{\partial y} \right) + \frac{\Delta t u_y}{4} \frac{\partial T_{DG}^{n+1}}{\partial y} = V^{n+\frac{1}{2}} \quad (20d)$$

Through careful examination, one can observe that (20) can be transformed into FADI method in (8) via the relationship as follows:

$$T_F^n = T_{DG}^n \quad (21a)$$

$$T_F^{n+\frac{1}{2}} = \frac{1}{2} \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right) \quad (21b)$$

$$T_F^{n+1} = T_{DG}^{n+1} \quad (21c)$$

Since both PR-ADI and DG-ADI methods are reducible to a common FADI method, we can now establish their relationship in a clear manner. Combining (21) and (10), it can be deduced that the

Figure 1. Simulation domain and locations of virtual points.

temperature variables between PR-ADI and DG-ADI methods satisfy the following relationship:

$$T_{PR}^n = T_{DG}^n \quad (22a)$$

$$T_{PR}^{n+\frac{1}{2}} = \frac{1}{2} \left(T_{DG}^{n+\frac{1}{2}} + T_{DG}^n \right) \quad (22b)$$

$$T_{PR}^{n+1} = T_{DG}^{n+1} \quad (22c)$$

It should be pointed out that the temperatures of both PR-ADI and DG-ADI methods are the same at T^n and T^{n+1} while the intermediate temperatures at $T^{n+\frac{1}{2}}$ are different via (22b).

3.2. Boundary Conditions

Due to tridiagonal matrices of DG-ADI, PR-ADI and FADI methods, additional treatments are required at the boundary of the simulation domain. Figure 1 illustrates a computation domain size of $ie \times je$ cells, where the grids start from 0 to ie in the x -direction and 0 to je in the y -direction. The simulation domain is bounded by the bold lines and the circled group of points are identified as the virtual points. It can be recognized that problem will arise at the boundaries $i=0$, $i=ie$, $j=0$ and $j=je$ during formation of tridiagonal matrix entries because they involve virtual points which are out of simulation domain.

In order to have a complete thermal simulation, the virtual points need to be replaced with an appropriate boundary condition. Hereby, we introduce a unified boundary condition given as

$$\kappa \frac{\delta T}{\delta n_\xi} = h_{\xi_\infty} T_\infty - h_\xi T \quad (23)$$

where T_∞ is a constant temperature, h_{ξ_∞} and h_ξ are the equivalent heat-transfer coefficients and n_ξ is the direction normal to the boundary. The unified boundary condition can cater to three different boundary conditions, namely:

- Specified Temperature

The first type of boundary is to assign a specific temperature along the boundary. Therefore, by

substituting $\kappa=0$ and $h_{\xi\infty}=h_{\xi}$, we can achieve

$$T = T_{\infty} \quad (24)$$

where T_{∞} can be set to any temperature desired.

- Heat Flux

Next, the second type of boundary is the heat flux along the boundary. Replacing $h_{\xi}=0$ and $h_{\xi\infty}T_{\infty}=q_{s_{\xi}}$, the specific heat flux along the boundary can be obtained as

$$\kappa \frac{\delta T}{\delta n_{\xi}} = q_{s_{\xi}} \quad (25)$$

where $q_{s_{\xi}}$ is the heat flux on the boundary surface s_{ξ} .

- Convection Boundary Condition

The third type of boundary condition is the convection boundary condition where heat is transferred from the boundary to the surroundings. By setting T_{∞} as the ambient temperature and $h_{\xi\infty}=h_{\xi}$, we can obtain the convection boundary condition as

$$\kappa \frac{\delta T}{\delta n_{\xi}} = h_{\xi}T_{\infty} - h_{\xi}T \quad (26)$$

It can be seen that the unified boundary condition is able to accommodate all three different kinds of boundary conditions. By simply substituting a set of variables, a specific boundary condition can be implemented conveniently.

To implement the unified boundary condition, (23) is discretized using finite-differencing at the boundary to obtain the update equation for the virtual point. For instance, at boundary $i = 0$, the update equations for the virtual point of $T|_{-1,j}$ read:

$$T|_{-1,j}^n = T|_{1,j}^n + \frac{2\Delta x}{\kappa_{0,j}}(h_{x\infty}T|_{\infty} - h_x T|_{0,j}^n) \quad (27a)$$

$$T|_{-1,j}^{n+\frac{1}{2}} = T|_{1,j}^{n+\frac{1}{2}} + \frac{2\Delta x}{\kappa_{0,j}}(h_{x\infty}T|_{\infty} - h_x T|_{0,j}^{n+\frac{1}{2}}) \quad (27b)$$

$$T|_{-1,j}^{n+1} = T|_{1,j}^{n+1} + \frac{2\Delta x}{\kappa_{0,j}}(h_{x\infty}T|_{\infty} - h_x T|_{0,j}^{n+1}) \quad (27c)$$

where $T|_{i,j}$ and $\kappa_{i,j}$ represent the temperature and thermal conductivity at discrete points $(i\Delta x, j\Delta y)$, respectively. Other virtual points on the boundaries $i = ie$, $j = 0$ and $j = je$ can be derived in similar manner. Note that (27) is applicable for all PR-ADI, DG-ADI and FADI methods. For consistency, (27) needs to preserve the relationship among the ADI methods in (10), (21) and (22). In fact, it can be shown that (27) indeed preserves these relationships. For instance, to prove that (22) is preserved, T is first replaced with T_{DG} . Upon adding (27a) and (27b), we obtain

$$T_{DG}|_{-1,j}^n + T_{DG}|_{-1,j}^{n+\frac{1}{2}} = (T_{DG}|_{1,j}^n + T_{DG}|_{1,j}^{n+\frac{1}{2}}) + \frac{2\Delta x}{\kappa_{0,j}}(2h_{x\infty}T_{\infty} - h_x(T_{DG}|_{0,j}^n + T_{DG}|_{0,j}^{n+\frac{1}{2}})). \quad (28)$$

Dividing both sides by two, it can be immediately recognized that (28) is reducible to

$$T_{PR}|_{-1,j}^{n+\frac{1}{2}} = T_{PR}|_{1,j}^{n+\frac{1}{2}} + \frac{2\Delta x}{\kappa_{0,j}}(h_{x\infty}T_{PR}|_{\infty} - h_x T_{PR}|_{0,j}^{n+\frac{1}{2}}) \quad (29)$$

through preservation of (22). One can see that (29) is the same as that obtained directly from (27b) by replacing T with T_{PR} . Other relationships of (10) and (21) can be proven in similar fashion.

Figure 2. Flowchart of GPU-accelerated FADI method.

3.3. GPU Acceleration

To further accelerate FADI method, GPU can be utilized through the implementation of CUDA [19]–[21]. CUDA is Nvidia’s parallel computing architecture which increases computing performance by harnessing the power of the GPU. For our simulation, we make use of a low cost Nvidia GTS 250. This GPU card has a set of 16 built-in streaming multiprocessors, where each streaming multiprocessor consists of a shared memory, a set of 32-bits registers and read-only caches. As each multiprocessor contains 8 streaming processors, this sums up to 128 streaming processors, where every streaming processor supports parallel executing model of the single-instruction multiple-data (SIMD). There is also a built-in 1 GB Random Access Memory (RAM) which acts as the global (read and write) and constant (read only) memory for the storage of datas.

Figure 2 shows the flowchart of GPU-accelerated FADI method for thermal simulation, where ie and je represent the number of cells in the x - and y -directions respectively. In the first procedure, three kernels need to be invoked. The first kernel involves the explicit update of (8a) where data can be parallelized in both x - and y -directions. For (8b), it is an implicit update equation involving tridiagonal matrix on the left-hand side (LHS). Inverting tridiagonal matrix in (8b) involves LU factorization where forward elimination is first required, followed by backward substitution, both along the x -direction. Therefore, the second kernel corresponds to forward elimination while the third kernel corresponds to backward substitution. It should be noted that since both forward elimination and backward substitution are performed along the x -direction, data can be parallelized in the y -direction. In the second procedure, the same kernel is invoked for (8c) as in (8a), while the forward elimination and backward substitution kernels for (8d) are performed along y -direction. Hence, data can be parallelized in the x -direction.

4. NUMERICAL RESULTS

4.1. Efficiency and Accuracy

To verify the efficiency of our proposed thermal FADI simulation, we compute the CPU time needed to run using the DG-ADI, PR-ADI and FADI method over a range of computation domain size from 500×500 up to 4500×4500 cells. The programs have been compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS), and run on a platform of Intel Dual Core 2.66 GHz processor. The thermal simulation consists of a chip with size of $16 \text{ mm} \times 16 \text{ mm}$, thermal conductivity $\kappa = 100 \text{ W/m-K}$, density $\rho = 2500 \text{ kg/m}^3$ and specific heat $C_p = 700 \text{ J/kg-K}$. The chip is discretized in the x -direction with $\Delta x = 0.016/ie \text{ m}$ and y -direction with $\Delta y = 0.016/je \text{ m}$. The numerical simulation is performed for 1000 time steps at $\gamma = 2$, where $\gamma \leq 0.5$ is the stability constraint of the explicit method in thermal simulation expressed as

$$\gamma = \frac{\kappa \Delta t}{\rho C_p} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \leq \frac{1}{2}. \quad (30)$$

Table I. CPU efficiency gain of FADI over DG-ADI and PR-ADI methods for different computation domain ($\gamma=2$, 1000 time steps)

Domain Size	Efficiency Gain	
	FADI vs PR-ADI	FADI vs DG-ADI
500×500	1.397	2.144
1000×1000	1.306	2.125
1500×1500	1.369	2.333
2000×2000	1.363	2.497
2500×2500	1.548	3.393
3000×3000	2.635	5.669
3500×3500	2.660	6.477
4000×4000	4.057	7.027

Table II. GPU efficiency gain of FADI over DG-ADI and PR-ADI methods for different computation domain ($\gamma=2$, 1000 time steps)

Domain Size	Efficiency Gain	
	GPU FADI vs CPU PR-ADI	GPU FADI vs CPU DG-ADI
500×500	1.962	3.013
1000×1000	2.816	4.581
1500×1500	2.860	4.849
2000×2000	2.842	5.209
2500×2500	3.112	6.820
3000×3000	5.438	11.700
3500×3500	5.534	13.477
4000×4000	9.734	16.861

Table I shows the CPU efficiency gain of FADI over DG-ADI and PR-ADI methods for different computation domain. The efficiency gain for FADI over PR-ADI method increases from 1.3 to 4 as the computation domain increases. If FADI method is further compared to DG-ADI method, the efficiency gain increases from 2.1 to 7, which is a significant gain. It can be seen that DG-ADI method is the slowest among all three methods, due to its more complicated RHS which involves more spatial derivatives, c.f. (3). PR-ADI method is faster compared to DG-ADI method because its RHS is less complicated with fewer spatial derivatives, c.f. (2). FADI method is the fastest among all methods, evident from its derivative-free RHS, c.f. (8), which is simpler and more concise.

To maximise the efficiency gain, the FADI method is further incorporated into the GPU to exploit data parallelism. Table II now shows the GPU efficiency gain of FADI over DG-ADI and PR-ADI method for different computation domain. The GPU run time of FADI method is compared against the CPU run time of PR-ADI and DG-ADI methods. Although it can certainly be obtained, the GPU run time for PR-ADI and DG-ADI methods are not presented because it is very laborious to implement DG-ADI and PR-ADI on GPU due to their complex RHS. The efficiency gain for GPU FADI over CPU PR-ADI method increases approximately from 2 to 10 times. If GPU FADI method is compared to CPU DG-ADI method, the efficiency gain can increase up to 17 times. It is thus ascertained that substantial gain in the overall efficiency is achievable for the FADI method with GPU.

To test the stability feature of FADI method beyond the time step stability limit of the explicit method, we now consider a 129×129 cells domain with spatial steps $\Delta x = \Delta y = 0.125$ mm and the stability constraint γ in (30) is varied from 0.5 to 50. Figure 3 shows the transient temperature results at a point ($i = 60, j = 60$) with various γ . From Figure 3, it can be seen that the FADI method still remains stable beyond $\gamma > \frac{1}{2}$. It can also be noted that as the γ gets larger, the FADI method deviates

Figure 3. Transient temperature results for a point at (60,60) with various γ . Figure 4. Relative norm error of FADI method with various γ .

away from the explicit method curve. This is as expected due to the trade off between efficiency (use of larger time step) and accuracy.

To investigate the error for the FADI method with different γ , we now compute the relative norm error defined as

$$\text{error} = \frac{\sqrt{\sum_{i=0}^{ie} \sum_{j=0}^{je} |T_{FADI_{i,j}} - T_{explicit_{i,j}}|^2}}{\sqrt{\sum_{i=0}^{ie} \sum_{j=0}^{je} |T_{explicit_{i,j}}|^2}} \quad (31)$$

where T_{FADI} is the temperature recorded using FADI method with various γ and $T_{explicit}$ is the temperature recorded by the explicit method with $\gamma = 0.5$. Figure 4 shows the relative norm error of FADI method with various γ for 1200 time steps. At $\gamma = 2.5$, the FADI enables us to choose Δt 5 times larger than that of the explicit method, at the expense of only 4.3×10^{-4} of relative norm error. Even at $\gamma = 50$, the Δt of FADI can be as high as 100 times than that of the explicit method,

Figure 5. Five-layer silicon structure.

Figure 6. Power density of the functional blocks layer

at the expense of only 0.011 relative norm error. This implies a significantly huge potential of FADI method to exploit large time step while maintaining stability and accuracy.

4.2. Simulation Example

For numerical illustration, we consider a five-layer silicon structure of dimension $9.3 \text{ mm} \times 9.3 \text{ mm} \times 1.2 \text{ mm}$ with microchannel cooling shown in Figure 5. The functional blocks layer has thermal conductivity $\kappa = 315.82 \text{ W/m}\cdot\text{K}$, density $\rho = 8954.5 \text{ kg/m}^3$, specific heat $C_p = 383.65 \text{ J/kg}\cdot\text{K}$. For the microchannels layer, it consists of four microchannels along the x -direction with fluid velocity u_x , channel width and height set as 1.2 m/s , $700 \text{ }\mu\text{m}$ and $300 \text{ }\mu\text{m}$ [7], respectively. The fluid is water with thermal conductivity $\kappa = 0.613 \text{ W/m}\cdot\text{K}$, density $\rho = 997 \text{ kg/m}^3$ and specific heat $C_p = 4179 \text{ J/kg}\cdot\text{K}$. These two layers are separated by a silicon layer with thermal conductivity $\kappa = 131 \text{ W/m}\cdot\text{K}$, density $\rho = 2329 \text{ kg/m}^3$ and specific heat $C_p = 700 \text{ J/kg}\cdot\text{K}$. The power density of the functional blocks layer is shown in Figure 6, along with an observation point marked by "X" at the functional block with the highest power density. The top, bottom layers and all four side walls are terminated by convection boundary conditions. For top and bottom layers, the effective heat transfer coefficients are $7 \times 10^3 \text{ W/m}^2\cdot\text{K}$ and $6 \times 10^3 \text{ W/m}^2\cdot\text{K}$, respectively, while for the four side walls, the effective heat transfer coefficients are $5 \times 10^3 \text{ W/m}^2\cdot\text{K}$.

To include the transfer of thermal energy across multiple layers, we use the change of heat flux in the vertical z -direction as part of the heat generation rate g . Assuming no microchannel flowing

Figure 7. Temperature profile of the functional blocks layer at 0.2 s

Figure 8. Transient temperature results for a point at (170,120,4) with various γ

across vertical layers, the specific heat flux q in the z -direction is given by

$$q = \kappa \frac{\partial T}{\partial z}. \quad (32)$$

The heat generation rate g can then be obtained from the change of heat flux as

$$g = \frac{\partial q}{\partial z} = \frac{\partial}{\partial z} \left(\kappa \frac{\partial T}{\partial z} \right). \quad (33)$$

Using this approach, the domain is discretized with spatial steps $\Delta x = \Delta y = 39.9 \mu\text{m}$, while the vertical height between layers are set as $\Delta z = 300 \mu\text{m}$.

Figure 9. Relative norm error of FADI method with various γ for the 3D structure in Figure 5

Figure 10. Transient temperature results for a point at (170,120,4) with 0, 2, 4 and 8 microchannels

Figure 7 shows the temperature profile of the functional blocks layer at 0.2 s. The highest temperature is about 83°C. Taking an observation point at $i = 170$, $j = 120$, $k = 4$ around the maximum temperature (marked by "X" in Figure 6 earlier), the transient temperature results for various γ are shown in Figure 8. It can be seen again that the FADI method is stable for $\gamma > 0.5$. The results are compared against the explicit method and they agree well even for high $\gamma = 50$. The relative norm error for the 3D structure is computed by

$$error = \frac{\sqrt{\sum_{i=0}^{ie} \sum_{j=0}^{je} \sum_{k=0}^{ke} |T_{FADI_{i,j,k}} - T_{explicit_{i,j,k}}|^2}}{\sqrt{\sum_{i=0}^{ie} \sum_{j=0}^{je} \sum_{k=0}^{ke} |T_{explicit_{i,j,k}}|^2}} \quad (34)$$

where ke is the number of layers in the structure. Figure 9 plots the relative norm error for the 3D structure and it can be seen that the errors with increasing γ are still relatively low. Compared to Figure 4, the relative norm errors in Figure 9 have lower values due to smaller and finer spatial steps used. To investigate the effect of the number of microchannels on temperature, we further plot the transient temperature at observation point $i = 170$, $j = 120$, $k = 4$ with 0, 2, 4 and 8 microchannels in Figure 10. It can be observed that as the number of microchannels increases, the steady state temperature decreases. The time taken to reach steady state also decreases. Furthermore, it is worth nothing that the structure experiences the most significant drop in temperature when the number of microchannels is added from zero to two, which further ascertains its cooling effect.

5. CONCLUSIONS

This paper has presented FADI method for efficient thermal simulation of integrated circuits. The FADI method has been formulated from PR-ADI and DG-ADI methods. The update procedure of the proposed method has basic implicit form which features derivative-free RHS and hence, better efficiency and conciseness. Subsequently, through the basic implicit form of FADI method, the relationship between classical PR-ADI and DG-ADI methods has been clarified and elucidated in detail. A unified boundary condition that can cater to common kinds of boundary conditions in thermal simulation has also been introduced. To further accelerate FADI method, the GPU has also been utilized through CUDA implementation. It has been shown that high efficiency gain can be achieved using the proposed FADI method through large time step size and data parallelism, while maintaining stability and good accuracy. As numerical illustration, an integrated circuit structure with microchannel cooling has been demonstrated. Numerical results have further ascertained the cooling effect of the microchannels. The proposed method herein can also be applied to other implicit method such as the locally one dimensional (LOD) method, with additional spatial derivative with respect to z -direction in the 3-D heat transfer equation (and possibly including fluid flow). These extensions may be further addressed in other publications.

REFERENCES

1. Frantz F, Labrak L, O'Connor I. 3D IC floorplanning: Automating optimization settings and exploring new thermal-aware management techniques. *Microelectronics Journal* 2012; **43**(6): 423–432.
2. Cuesta D, Risco-Martin JL, Ayala JL. 3D thermal-aware floorplanner using a MILP approximation. *Microprocessors and Microsystems* 2012; **36**(5): 344–354.
3. Gracia NRD, Rajaram S, Nivethitha N, Sudarsan A. Thermal aware modern VLSI floorplanning. *International Conference on Devices, Circuits and Systems* 2012; 187–190.
4. Skadron K, Stan MR, Huang W, Velusamy S, Sankaranarayanan K, Tarjan D. Temperature-aware microarchitecture. *Proceedings of the 30th Annual International Symposium on Computer Architecture* 2003; 2–13.
5. Brunschweiler T, Michel B, Rothuizen H, Kloter U, Wunderle B, Oppermann H, Reichl H. Interlayer cooling potential in vertically integrated packages. *Microsystem Technologies* 2009; **15**(1): 57–74.
6. Bakir MS, King C, Sekar D, Thacker H, Dang B, Huang G, Naemi A, Meindl JD. 3D heterogeneous integrated systems: liquid cooling, power delivery, and implementation. *Proceedings IEEE Custom Integrated Circuits Conference* 2008; 663–670.
7. Koo JM, Im S, Jiang L, Goodson KE. Integrated microchannel cooling for three-dimensional electronic circuit architectures. *ASME Journal of Heat Transfer* 2005; **127**(1): 49–58.
8. Ozisik MN. *Finite Difference Methods in Heat Transpor*. CRC Press: New York, 1994.
9. Zhang HX, Webb PW. A modified finite difference method for modelling diffusion processes. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 1996; **9**(3): 251–223.
10. Wong CC, Lee C. A hybrid TLM-FDTD method for the modelling of diffusion. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 2000; **13**(1): 1–10.
11. Dehghan M. A finite difference method for a non-local boundary value problem for two-dimensional heat equation. *Applied Mathematics and Computation* 2000; **112**: 133–142.
12. Dehghan M. Alternating direction implicit methods for twodimensional diffusion with a non-local boundary condition. *International Journal of Computer Mathematics* 1999; **72**: 349–366.
13. Wang TY, Chen CCP. Thermal-ADI - A linear-time chip-level dynamic thermal-simulation algorithm based on alternating-direction-implicit (ADI) method. *IEEE Transactions on VLSI Systems* 2003; **11**(4): 691–700.
14. Peaceman DW, Rachford Jr. HH. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics* 1955; **3**(1): 28–41.

15. Douglas Jr. J, Gunn JE. A general formulation of alternating direction methods Part I: parabolic and hyperbolic problems. *Numerische Mathematik* 1964; **6**: 428–453.
16. Kaviany M. *Principles of Heat Transfer*, John Wiley & Sons: New York, 2002.
17. Tan EL. Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods. *IEEE Transactions on Antennas and Propagation* 2008; **56**(1): 170–177.
18. Tan EL. Concise current source implementation for efficient 3-D ADI-FDTD method. *IEEE Microwave and Wireless Components Letters* 2007; **17**(11): 748–750.
19. Pharr M, Fernando R. *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley: New Jersey, 2005.
20. Owens JD, Houston M, Leubke D, Green S, Stone JE, Phillips JC, GPU computing. *Proceedings of the IEEE* 2008; **96**(5): 879–899.
21. Nvidia Corporation. *NVIDIA CUDA Programming Guide*, Version 2.3, 2009.