NanoTherm: An Analytical Fourier-Boltzmann Framework for Full Chip Thermal Simulations

Shashank Varshney
Electrical Engg., IIT Delhi
shashankvarshney94@gmail.com

Hameedah Sultan
School of IT, IIT Delhi
hameedah@cse.iitd.ac.in

Palkesh Jain
Qualcomm
palkesh@qti.qualcomm.com

Smruti R. Sarangi
Computer Science, IIT Delhi
srsarangi@cse.iitd.ac.in

***Abstract*—Temperature simulation is a classic problem in EDA, and researchers have been working on it for at least the last 15 years. In this paper, we focus on fast Green's function based approaches, where computing the temperature profile is as simple as computing the convolution of the power profile with the Green's function. We observe that for many problems of interest the process of computing the Green's function is the most time consuming phase, because we need to compute it with the slower finite difference or finite element based approaches. In this paper we propose a solution, *NanoTherm*, to compute the Green's function using a fast analytical approach that exploits the symmetry in the thermal distribution.**

Secondly, conventional analyses based on the Fourier's heat transfer equation fail to hold at the nanometer level. To accurately compute the temperature at the level of a standard cell, it is necessary to solve the Boltzmann transport equation (BTE) that accounts for quantum mechanical effects. This research area is very sparse. Conventional approaches ignore the quantum effects, which can result in a 25 to 60% error in temperature calculation. Hence, we propose a fast analytical approach to solve the BTE and obtain an exact solution in the Fourier transform space.

Using our fast analytical models, we demonstrate a speedup of 7-668X over state of the art techniques with an error limited to 3% while computing the combined Green's function.

I. Introduction

For at least the last 15 years, the design community has viewed on-chip temperature as one of the most important criteria while designing a new SoC. High temperatures result in several adverse effects. The reliability of the device is negatively affected [1] and the carrier mobility is degraded, resulting in poorer performance [1]. Moreover, the chip temperature determines the leakage power. Finally, note that with increasing power and transistor densities, the problem of high on-chip temperatures is expected to get worse.

Different stages of the design process have different levels of information available, and the requirements for thermal optimization at each stage are different. For instance, at the architecture level, standard cell information or package level information such as the properties of the heat spreader and heat sink may not be available. Hence designers make assumptions about the missing information, and evaluate the design space from a thermal point of view. After synthesis and standard cell mapping, designers can conduct more accurate thermal analyses to determine the nature of packaging and expected on-chip temperatures for different workloads. The latter can be conveyed to software and systems designers such that they can optimize the system at their end. Over the entire design cycle, thousands of candidate designs have to be evaluated based on the information available at each stage to determine the optimal configuration. In such a case the speed of the thermal simulation becomes a bottleneck in the design process [2], [3]. As a result, **fast thermal estimation** at all stages of the design is necessary.

Many thermal simulators [4], [5], [6], [7] which are based on the classical Fourier heat transfer equation exist in this space. The fastest approaches use the Green's function, which is defined as the impulse response of a unit power source (Dirac delta function) [5], [6], [8]. The main drawback of many of the Green's function based approaches is that they rely on a traditional finite element or finite difference based simulator to compute the Green's functions [5], [6], [9]. If the geometry of the chip or the boundary conditions change, the Green's function will have to be recomputed, making it a very slow and time consuming process [3]. Moreover, as we move down to smaller dimensions, at the nanometer scale, the quantum effects become significant. Conventional approaches do not take the quantum effects into account, and do a regular analysis based on classical Fourier's heat transfer equations. It has been shown in [10], [11], [12] and in our analysis that this leads to a 25 to 60% error in estimation. Particularly, in the later stages of the design process, an accurate estimate of temperature is needed at the nanometer scale for two reasons: 1) to optimize the design of standard cells by taking thermal effects into account, and 2) to design mixed-signal blocks, where the analog functional units are highly sensitive to temperature.

We address the drawbacks of existing works by proposing *NanoTherm*.

1) We reduce an $O(N^2)$ problem to an $O(N)$ problem, and then use a Hankel transform based approach to solve the traditional heat conduction problem using Fourier equations.

2) Using transform based approaches we derive an exact analytical solution for the gray Boltzmann transport equation. Our approach, *NanoTherm*, is 7-688 times faster than the state of the art.

In Section II we introduce the relevant background and related work. Then we discuss our methodology in Section III. We proceed to Section IV to present the evaluation of our proposed approach and the results obtained, and finally conclude in Section V.

II. Background and Related Work

A. Background of Heat Transfer

The classical Fourier equation is used to solve heat transfer problems in solids. It does not model quantum effects and is meant to be used in scenarios where the geometry is orders of magnitude larger than the mean free path of phonons. It is given by:

$$\rho c \frac{\partial T}{\partial t} - k\nabla^2 T = q_{vol}, \tag{1}$$

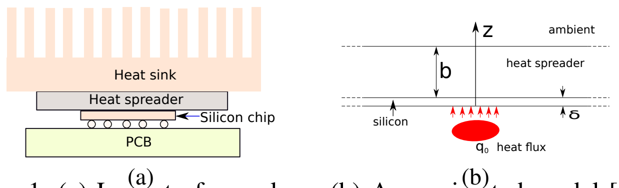
where, k is the thermal conductivity, ρ is the density, c is the specific heat, and q_{vol} is the volumetric heat. The temperature field is represented by T, and time is represented by t.

This equation is typically solved using either finite element (FEM) or finite difference methods (FDM). In the FEM technique, we divide a 3D region into small blocks, and solve the heat transfer equation for each small block by either finding an analytical solution, or by choosing a function from a set of many trial functions that minimize the residual error. These equations are then combined into a global system of equations, which are solved using regular matrix methods. In the case of the finite difference method, we replace the differential equations with a set of algebraic equations. They are similar to recurrence relations, and are solved using linear algebra techniques. For example, we replace $df(x)/dt$ with $(f(x+h) - f(x))/\Delta t$, where Δt and h tend to 0. A very important offshoot of finite difference methods comprise techniques that model a temperature estimation problem as an analogous electrical circuit simulation problem (HotSpot [4] and 3D-ICE [7]).

1) Green's Function based Techniques: Both the finite difference and finite element methods require matrix inversion, which is a slow process. A faster way of computing the thermal profile is the Green's function based technique [5], [6], [8]. A Green's function is defined as the impulse response of a unit power function. This can be obtained by applying 1 W of power to a very small area (approximating the Dirac delta function). The resultant temperature distribution is the Green's function, G. The advantage of this approach is that we can precompute and store the Green's functions, and then quickly use them at runtime to compute the temperature profile for a given power profile. This can be done as follows:

$$T = P \star G, \tag{2}$$

where, P is the power field, and $\star$ is the convolution operator. There are many proposals [6], [5] that use Green's functions to speed up thermal estimation. However, these techniques still rely on traditional FEM and FDM based techniques to compute the Green's function in the first place. This is a very slow process. In situations where thousands of geometries have to be evaluated, the time taken in computing the Green's function will dominate the total modeling time. Hence, the main aim in this paper *is to very quickly compute the Green's function for a given geometry.*



(a)
(b)

Fig. 1: (a) Layout of a package (b) Approximated model [8]

2) Geometry of the Chip: Let us now look at the geometry of a typical chip (shown in Figure 1a).

We have a layer of silicon that contains all the transistors. Over that, we have a heat spreader, which is made of a high thermal conductivity material. This helps spread the heat and reduce the formation of thermal hot spots. Above the heat spreader, we have a heat sink that has multiple fins to increase the surface area. We can use an approximate model where we remove the heat sink and substitute an isothermal layer in its place; this is a standard approximation made by other authors as well [8]. In any case, extending our model to include the heat sink is trivial.

B. Boltzmann Transport Equation (BTE)

Atoms in a silicon substrate are arranged as a lattice. Synchronized perturbation of groups of atoms from their equilibrium positions is known as a vibration. The propagation of this vibration is known as a lattice wave (also known as *phonons*). This vibrational wave has a wavelength and a velocity. From wave-particle duality, phonons also behave as particles in the quantum mechanical sense. At the nanometer scale, phonons play an important role in determining the temperature distribution. Phonons are created because of thermal fluctuations, and can be absorbed, or can get dispersed while propagating through the silicon lattice. Hence, modeling phonon creation and dispersion is crucial to estimate the temperature at the nanometer scale. The distance that phonons travel before losing their energy is of the order of several mean free paths ($\sim$40–300 nm) [12]. When the dimensions under consideration are smaller than the mean free path, the phonon effects become significant. Hence, in modern day devices, where the device feature size is lower than the mean free path of phonons, modeling these effects is necessary to estimate temperature accurately.

To model the nanometer scale phonon effects, we typically use the molecular dynamics method, ballistic-diffusive method, or the Boltzmann transport equation. We shall focus on the Boltzmann transport equation (BTE) because it is relatively less computationally intensive and more accurate than other methods [13]. They model the heat transfer by modeling the scattering of phonons [14]. Specifically, we consider the gray BTE model that assumes a single mean frequency of phonons:

$$\frac{\partial e_w}{\partial t} + \vec{\mathbf{v}}_g . \nabla e_w - \frac{Q}{4\pi} = \left(\frac{\partial e_w}{\partial t}\right)_{collision}, \tag{3}$$

where, e_w is the energy density function per unit solid angle, $\vec{\mathbf{v}}_g$ is the group velocity of phonons, t is the time, and Q is the volumetric heat generation. The term on the RHS models the scattering of phonons [14].

C. Related Work

1) Green's Functions: The most influential work in analytically computing the Green's function has been done by Zhan et al. [8]. They compute the Green's function by dividing a chip into multiple layers and solving the Fourier equation. They assume that the Green's function consists of a sum of cosine based basis functions. Then they find the parameters of these basis functions for different settings. This takes $O(N^2 log(N))$ time primarily because the representation of the Green's function is generic, and the isotropic nature of heat spreading is not exploited. Also, they have not modeled the transient temperature profile. *NanoTherm* instead uses the Hankel transform to solve the Fourier equation. This reduces the complexity under consideration to $O(N)$ by leveraging the symmetry of the heat distribution. Our technique is also capable of modeling the transient temperature distribution. Other analytical Green's function based techniques are not capable of computing the transient temperature profile.

2) Fourier Analysis: In HotSpot [4], the authors divide the volume into small blocks and create an equivalent electrical circuit, and then solve it using matrix solvers. similar approach; and also models microchannels. All of these popular tools solve the Fourier equation only.

3) Solutions of the BTE: Hua et al. [11] solve a different variant of the BTE equation analytically, where they assume that the relaxation time and the specific heat are dependent on the frequency of phonons. We did not use this approach because this increases the simulation time significantly, and does not have commensurate gains in accuracy. Zahiri et al. solve the gray BTE model by transforming the BTE equation into a set of ordinary differential equations. Our approach gives an exact solution in the Fourier transform space, and thus is more efficient than solving a system of differential equations. ThermalScope [10] is the most related work because it takes into account both the Fourier and BTE models. It solves the gray BTE model (similar to *NanoTherm*) at the nanometer scale, and solves the Fourier equation at the level of the chip. They solve the gray BTE model using FEM and the discrete ordinate method (DOM). The slowest part of the algorithm is the FEM-based analysis, making it orders of magnitude slower than our approach.

III. Modeling Methodology

A. Fourier Analysis

1) Steady state: The Fourier's heat equation is given by Equation 1. Since we are solving at steady state and the problem under consideration is symmetric, the Fourier equation will reduce to Equation 4 in cylindrical coordinates. We are assuming that transistors are modeled as heat sources placed at the bottom of a silicon sheet (same assumption as [8]).

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial T_l}{\partial r}\right)+\frac{\partial}{\partial z}\left(\frac{\partial T_l}{\partial z}\right)=0, \tag{4}$$

where l is the layer number. In our case we have two layers. The subscript 1 represents the silicon layer and 2 represents the heat spreader layer (see Figure 1b). Our solution can be easily extended to include more layers. Next we describe the boundary conditions that we have considered for the chip.

Boundary conditions:

1) For a circular source of radius r_o, the heat flux for $|r| \le r_o$ is q_o, and for $|r| > r_o$ the heat flux is zero.

$$-k_1\frac{\partial T_1}{\partial z}\Big|_{z=0} = \begin{cases} q_o, & \text{for } |r| \le r_o, \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

2) Heat flux at the interface of the chip and the heat spreader is equal.

$$-k_1\frac{\partial T_1}{\partial z}\Big|_{z=\delta} = -k_2\frac{\partial T_2}{\partial z}\Big|_{z=\delta} \tag{6}$$

3) The temperature at the interface of the chip and the heat spreader is equal.

$$T_1(r,z)\Big|_{z=\delta} = T_2(r,z)\Big|_{z=\delta} \tag{7}$$

4) The temperature at the top of the heat spreader is uniform and it is maintained at the ambient temperature (T_a).

$$T_2(r,z)\Big|_{z=\delta+b} = T_a \tag{8}$$

5) At large distances from the center, the temperature rise is infinitesimally small, and can be considered as zero.

$$T_l(r,z)|_{r\to\infty} - T_a = 0 \tag{9}$$

Hankel Transform: To solve Equation 4, let us use the Hankel transform. A 1D Hankel transform is equivalent to a 2D Fourier transform of an isotropic function in polar coordinates. This reduces the complexity of the problem from $O(n^2)$ to $O(n)$. Now, let us simplify Equation 4.

$$\nabla_o T_l + \frac{\partial^2}{\partial z^2}T_l = 0, \tag{10}$$

where ∇_o is zero order Bessel differential operator. Since we are interested in the temperature rise, we subtract the ambient temperature (T_a) from $T_l(r,z)$, and set it equal to $\phi_l(r,z)$:

$$\phi_l(r,z) = T_l(r,z) - T_a \tag{11}$$

Combining Equations 10 and 11 we get:

$$\nabla_o\phi_l + \frac{\partial^2}{\partial z^2}\phi_l = 0 \tag{12}$$

We compute the zero order Hankel transform of both sides:

$$\frac{\partial^2\overline{\phi_l}}{\partial z^2} = \sigma^2\overline{\phi_l} \tag{13}$$

Here, $\overline{}$ represents the Hankel transform. The solution for the silicon plate and the heat spreader respectively are as follows:

$$\overline{\phi_1} = C_1e^{\sigma z} + C_2e^{-\sigma z} \tag{14}$$

$$\overline{\phi_2} = C_3e^{\sigma z} + C_4e^{-\sigma z} \tag{15}$$

We apply the zero order Hankel transform to the boundary conditions. Next we apply these transformed boundary conditions to Equations 14 and 15 and solve for the constants. The final solution is given by Equation 16.

$$\phi_1(r,z) = \frac{q_o r_o}{k_1}\int_0^\infty \frac{J_1(r_o\sigma)}{\sigma}\frac{(e^{-\sigma z}+f(\sigma)e^{\sigma z})}{1-f(\sigma)}J_o(\sigma r)d\sigma,$$
$$f(s) = e^{-2\sigma\delta}\frac{k_1\tanh(b\sigma)-k_2}{k_1\tanh(b\sigma)+k_2}, \tag{16}$$

where J_o and J_1 are the zero and first order Bessel functions of the first kind respectively.

2) Transient Analysis: The analogous Fourier equation in cylindrical coordinates is as follows:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial T_l}{\partial r}\right)+\frac{\partial}{\partial z}\left(\frac{\partial T_l}{\partial z}\right)=\frac{\rho_l c_l}{k_l}\frac{\partial T_l}{\partial t} \tag{17}$$

To remove the derivative with respect to time, we first compute the Laplace transform of both sides of Equation 17, and after that we compute the Hankel transform (similar to the steady state analysis). We end up with a simple ordinary differential equation (ODE).

$$\frac{\partial^2\overline{\boldsymbol{\phi_l}}}{\partial z^2} = p_l^2(s,\sigma)\overline{\boldsymbol{\phi_l}},$$
$$p_l(s,\sigma) = \sqrt{\sigma^2+\frac{\rho_l c_l}{k_1}s} \tag{18}$$

Here, boldface ($\boldsymbol{\phi}$) represents Laplace transform. The solutions for the silicon plate and the heat spreader respectively are as follows:

$$\overline{\boldsymbol{\phi_1}} = C_5 e^{p_1(s,\sigma)z} + C_6 e^{-p_1(s,\sigma)z} \tag{19}$$
$$\overline{\boldsymbol{\phi_2}} = C_7 e^{p_2(s,\sigma)z} + C_8 e^{-p_2(s,\sigma)z} \tag{20}$$

Applying the boundary conditions to Equations 19 and 20 and solving for constants, we get the temperature distribution function for the silicon chip as given by Equation 21.

$$\overline{\boldsymbol{\phi}_1}(s,z,\sigma) = \frac{q_o r_o}{k_1}\frac{J_1(r_o s)}{p_1(s,\sigma)s\sigma}\frac{1}{1-f(s,\sigma)}$$
$$\times\left(e^{-p_1(s,\sigma)z}+f(s,\sigma)e^{p_1(s,\sigma)z}\right),$$
$$f(s,\sigma) = e^{-2p_1(s,\sigma)\delta}\frac{k_1 p_1(s,\sigma)\tanh(p_2(s,\sigma)b)-k_2 p_2(s,\sigma)}{k_1 p_1(s,\sigma)\tanh(p_2(s,\sigma)b)+k_2 p_2(s,\sigma)} \tag{21}$$

The integral of Equation 16 and 21 can be evaluated using numerical integration techniques. The result is the Hankel transform of the Green's function. This can easily be converted to a 2D Fourier transform given the equivalence of the Fourier and Hankel transforms. This can subsequently be used to compute the thermal profile.

B. Solution to the Boltzmann Transport Equations

1) Steady state: The gray Boltzmann equation is given by:

$$\frac{\partial e}{\partial t}+\vec{\mathbf{v}}_g.\nabla e-\frac{Q}{4\pi} = -\frac{e-e_o(\vec{\boldsymbol{r}})}{\tau_{eff}} \tag{22}$$

TABLE I: Glossary

Symbol	Full Form	Meaning
b		Thickness of the heat spreader
δ		Thickness of the silicon die
k		Thermal Conductivity
σ		Hankel domain
s		Laplace domain
$\boldsymbol{x}$	Boldface	Laplace Transform
¯	Overline	Hankel Transform
~	Overtilde	Fourier Transform

The term in the RHS of Equation 22 comes from the relaxation time approximation [12]. e is frequency independent phonon energy density per unit solid angle, e_o is the phonon energy density, Q is the volumetric heat generation, and τ_{eff} is the effective relaxation time. The phonon energy density e_o follows Equation 23 [12].

$$e_o(\vec{\boldsymbol{r}}) = \frac{1}{4\pi}C\Delta T = \frac{1}{4\pi}C(T-T_{ref}), \tag{23}$$

where C is the specific heat at the reference temperature, T_{ref} is the reference temperature (computed by Fourier analysis), and T is the lattice temperature. Here e and ΔT are unknown variables and to relate them let us integrate Equation 22 over all solid angles (4π steradians). The LHS of Equation 22 after integration becomes:

$$\frac{\partial E}{\partial t}+\nabla.q-Q_{vol}, \tag{24}$$

where $E = \int e\,d\Omega$, Ω is the solid angle, E is the energy, $q = \int \vec{\mathbf{v}}_g.e\,d\Omega$ is the heat flux, and $Q_{vol} = \int Q/4\pi\,d\Omega$ is the volumetric heat generation. Equation 24 is the energy conservation equation, which must always be equal to zero. Hence, the RHS of Equation 22 (after integration) has to be zero. We thus have:

$$\int_\Omega\left[\frac{e}{\tau_{eff}}-\frac{1}{4\pi}\frac{C}{\tau_{eff}}\Delta T\right]d\Omega = 0 \tag{25}$$

Putting the value of e_o (Equation 23) into Equation 22, and expanding $\vec{\mathbf{v}}_g.\nabla e$, we have:

$$\frac{\partial e}{\partial t}+v_g\cos\theta\frac{\partial e}{\partial z}+v_g\sin\theta\cos\phi\frac{\partial e}{\partial x}+$$
$$v_g\sin\theta\sin\phi\frac{\partial e}{\partial y} = -\frac{e}{\tau_{eff}}+\frac{C}{4\pi\tau_{eff}}\Delta T+\frac{Q}{4\pi} \tag{26}$$

At steady state $\frac{\partial e}{\partial t}$ is equal to zero. Computing the Fourier transform of Equations 26 and 25 and re-arranging the terms, we get:

$$\widetilde{e} = \frac{C}{4\pi}\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C}{1+\Lambda\cos\theta i\xi_z+\Lambda\sin\theta\cos\phi i\xi_x+\Lambda\sin\theta\sin\phi i\xi_y} \tag{27}$$

Here, ~ represents the Fourier transform, ξ_x, ξ_y and ξ_z are in the spatial domain and i is equal to $\sqrt{-1}$. Putting Equation 27 into Equation 25, we thus have:

$$\Delta\widetilde{T} = \frac{1}{4\pi}\int_{4\pi}\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C}{1+\Lambda\cos\theta i\xi_z+\Lambda\sin\theta\cos\phi i\xi_x+\Lambda\sin\theta\sin\phi i\xi_y}d\Omega \tag{28}$$

We convert the solid angle Ω into an azimuthal angle (ϕ) and a polar angle (θ). We assume: $\mu = \cos\theta$. We thus have:

$$\Delta\widetilde{T} = \frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1}\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C \;\; d\mu d\phi}{1+\Lambda\mu i\xi_z+\Lambda\sqrt{1-\mu^2}\cos\phi i\xi_x+\Lambda\sqrt{1-\mu^2}\sin\phi i\xi_y} \tag{29}$$

Equation 29 can be solved analytically very easily. The temperature distribution function for any arbitrary volumetric power density is given by Equation 30.

$$\Delta\widetilde{T} = \frac{\widetilde{Q}\tau_{eff}}{C}\frac{\frac{1}{\Lambda\xi}\tan^{-1}(\Lambda\xi)}{1-\frac{1}{\Lambda\xi}\tan^{-1}(\Lambda\xi)}, \tag{30}$$

where $\xi = \sqrt{\xi_x^2+\xi_y^2+\xi_z^2}$ and Λ is the mean free path of the phonon.

2) Transient analysis: The analogous transient BTE equation is given by Equation 26. This time we assume that $\frac{\partial e}{\partial t} \neq 0$. We compute the Fourier transform of both sides of Equation 26 and after re-arranging the terms, we have:

$$\widetilde{e} = \frac{C}{4\pi}\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C}{1+i\eta\tau_{eff}+\Lambda\cos\theta i\xi_z+\Lambda\sin\theta\cos\phi i\xi_x+\Lambda\sin\theta\sin\phi i\xi_y} \tag{31}$$

We compute the Fourier transform of Equation 25 and putting Equation 31 into it, we have:

$$\Delta\widetilde{T} = \frac{1}{4\pi}\int\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C \quad \times d\Omega}{1+i\eta\tau_{eff}+\Lambda\cos\theta i\xi_z+\Lambda\sin\theta\cos\phi i\xi_x+\Lambda\sin\theta\sin\phi i\xi_y} \tag{32}$$

Again we convert the solid angle Ω into an azimuthal angle (ϕ) and a polar angle (θ) and assume $\mu = \cos\theta$.

$$\Delta\widetilde{T} = \frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1}\frac{\Delta\widetilde{T}+\widetilde{Q}\tau_{eff}/C \quad \times d\mu d\phi}{1+i\eta\tau_{eff}+\Lambda\cos\theta i\xi_z+\Lambda\sin\theta\cos\phi i\xi_x+\Lambda\sin\theta\sin\phi i\xi_y} \tag{33}$$

Equation 33 can be solved analytically. The final temperature distribution function for any arbitrary volumetric power density is given by Equation 34.

$$\Delta\widetilde{T} = \frac{\widetilde{Q}\tau_{eff}}{C}\frac{\frac{1}{\Lambda\xi}\tan^{-1}\left(\frac{\Lambda\xi}{1+i\eta\tau_{eff}}\right)}{1-\frac{1}{\Lambda\xi}\tan^{-1}\left(\frac{\Lambda\xi}{1+i\eta\tau_{eff}}\right)}, \tag{34}$$

where η is in temporal domain.

For detailed solution of the Fourier and the BTE model, interested readers can refer to [15].

C. Combined Solution

We compute the base temperature profile of the chip by convolving the Fourier Green's function with the power profile of the chip, and then we use this thermal profile to compute the temperature of 1000×1000 nm^2 blocks using the BTE based Green's function (see Figure 2). This gives us the temperature profile of regions of interest: standard cells, and small functional units.

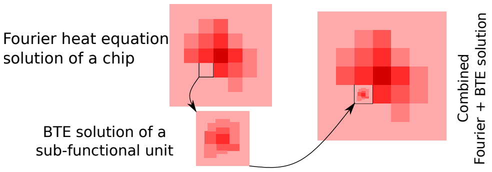


Fig. 2: Fourier-Boltzmann framework

1) Correction for Edges and Corners: The size of the chip is finite; however, for simplicity, we assume it to be infinite. This assumption results in an error in the calculation of the Green's function at the edges and the corners. To overcome this problem, we calculate the Green's function beyond the boundary of the chip (extended Green's function). This extended Green's function is then convolved with the power profile to obtain an extended thermal profile. The profile is then folded across the corners and edges to get the corrected thermal profile, since the boundaries are adiabatic.

IV. Evaluation

A. Setup

We run the simulations on an Intel i7 3^{rd}generation processor based desktop with 8GB RAM running Ubuntu 18.04. For validating our Fourier analysis results, we used a commercial CFD simulator COMSOL (Version 5.3b), and we compare our BTE solution against ThermalScope (available as *ISAC2*). The Fourier solution was done in R (version 3.5.1), and the BTE solution was done on Matlab 17b.

Error Metric: We have reported the root mean square (RMS) value of the error for all the test cases. Where percentage errors are reported, these are relative to the maximum temperature rise (similar to [7]). In ThermalScope the authors report the average error (average error is always less than the RMS error).

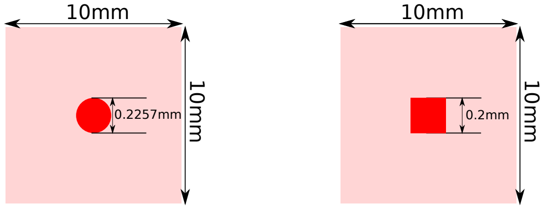
B. Fourier Analysis

We run the Fourier steady-state simulation for a chip with a heat spreader on top of it. The chip and heat spreader dimensions are $10\ mm \times 10\ mm \times 0.15\ mm$ and $10\ mm \times 10\ mm \times 3.52\ mm$ [9], [1] respectively. The conductivity of silicon and the heat spreader is $150\ W/mK$ and $256\ W/mK$ respectively (it is the effective conductivity of the heat spreader and the TIM).

*1) **Steady State**:* Green's function based full chip temperature calculation can be broken down into two parts [16], [6], [9]:

1) Green's function computation (offline)
2) Full chip thermal profile computation (online)

Green's function: We have calculated the Green's function assuming the source to be a circle of finite radius applied at the center of the chip (to exploit the symmetry of the thermal distribution).



(a) (b)

Fig. 3: (a) Circular source (b) Square source

However, the floorplan elements in a real chip are rectangular; they can only be discretized into small square grid points. Thus we need a way of mapping these square grid points to circular sources (for which we calculate the Green's function). For this, we take a circular source of equal area as the square source. We sample the continuous Green's function at the centers of the grid points determined by discretizing the chip into a grid. We have found that a grid size of 0.2 mm provides sufficient accuracy for a chip of area 100 mm^2 or more (Figure 3). Similar discretizations were found to be sufficient in [1]. The RMS error obtained by this approximation is $0.023°C$, which is small enough compared to the maximum temperature rise of $17°C$ (maximum error of 1.7%). Figure 4 shows the comparison of the calculated Green's function (using the circular source) against the Green's function obtained in COMSOL (using a square source of equal area).

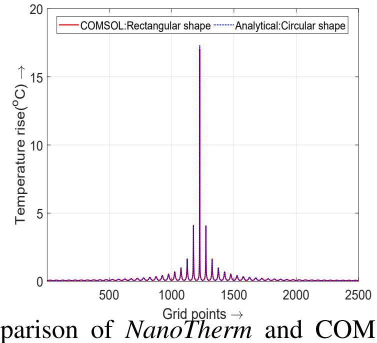


Fig. 4: Comparison of *NanoTherm* and COMSOL (Fourier, steady state Green's function)

Our implementation takes 0.082 s to compute the Green's function for a 10×10 mm^2 chip. We also calculated the Green's function using COMSOL for the same configuration, and observed a simulation time of 305 s. Other Green's function based simulators such as LightSim [6], [9] and PowerBlurring [1] depend on FEM based simulators for the calculation of the Green's function. If the geometry of the chip or the boundary conditions change, the Green's function will have to be recomputed. In our work, since we analytically obtain the Green's function, our approach gives designers the *flexibility to experiment with the package*, as there is no dependence on any external tool.

Full Chip Thermal Profile: We computed the thermal profile for four test cases (Figure 5 and 6). In the first two test cases we discretized a 10×10 mm^2 chip into a 50×50 grid. Test cases 3 and 4 have been implemented to evaluate our algorithm on two real floorplans.

Test Case 1: In this case, we have applied power sources at the center and all corners of the chip. This represents one of the worst case power profiles possible, since the corners and edges contribute to a large part of the error in Green's function based approaches [1]. We obtained an RMS error of $0.046°C$ (maximum error 1.57%) compared to COMSOL (Figure 5b). This verifies our corners and edge correction approach.

Test Case 2: This test case is similar to Test case 1, except that the power densities are much higher here (2500 W/cm^2). We have used a very high power density figure to evaluate our algorithm for extreme cases anticipated in next generation processors. An RMS error of $0.169°C$ (maximum error 2.88%) was observed in comparison to COMSOL.

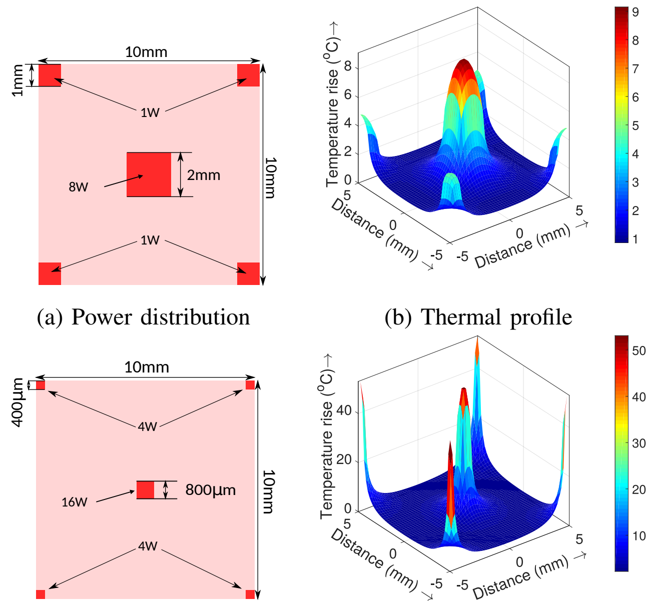
Test Case 3: We have further evaluated our algorithm using the floorplan of a processor containing a single core of Alpha21264 and an L2 cache. The dimensions of the processor are 16 $mm \times 16$ $mm \times 0.15$ mm. The core has 15 functional units. The power density of each functional unit is shown in Figure 6a (obtained from HotSpot). The calculated thermal profile of the chip is shown in Figure 6b. An RMS error of $0.047°C$ (maximum error 3.2%) was observed against the COMSOL model. This error is greater than that of test cases 1 and 2, since all the major sources have been placed along an edge of the processor.

Test Case 4: We have also implemented a dual-core processor in 45 nm technology based on the Intel Gainestown architecture. Each core is divided into six sub-units, and all cores share an L3 cache. The power density of each block is shown in Figure 6c. The size of the processor is 11.2 mm $\times$ 11.2 mm $\times$ 0.15 mm. The calculated thermal profile is shown in Figure 6d. An RMS error of $0.099°C$ (maximum error 1.9%) was observed against the COMSOL model.

Runtime: The total runtime of the algorithm was 83.5 ms for all the test cases (including the Green's function computation time). We need only 1.5 ms in the online stage to compute the full chip temperature profile, by taking the FFT of the Green's function and the power profile and computing the inverse transform of the product. To compute the same steady state thermal profile, COMSOL requires 305 s. Thus *NanoTherm* provides a speedup of $3652X$ over COMSOL.

*2) **Transient:**:* For transient analysis, we use the same setup. The density of silicon and the heat spreader are 2330 kg/m^3 and 8960 kg/m^3 respectively, and the specific heat values are 700 $J/kg.K$ and 390 $J/kg.K$ respectively.

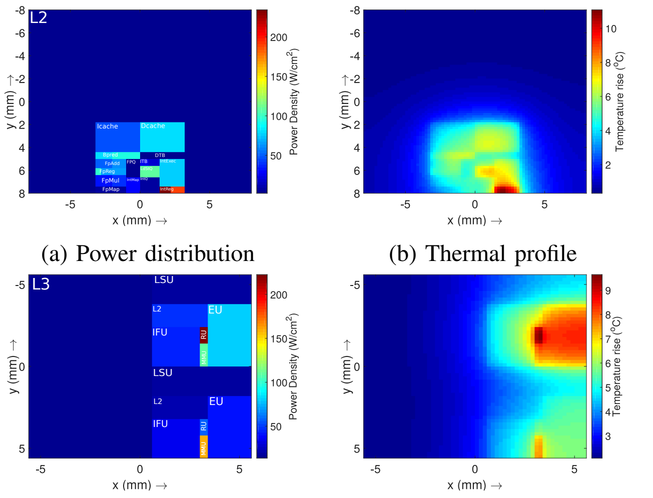
Step response: We start with applying a 1 W step source at the center of the chip. We calculated the step response of the chip for 100 radial points and 40-time steps. The runtime of the algorithm was 4.15 s (Note: 25% of the time is going in the slow inverse Laplace transform routine of R). Also, *NanoTherm* has been implemented in R whereas HotSpot and 3DICE



(a) Power distribution (b) Thermal profile

(c) Power distribution (d) Thermal profile

Fig. 5: Power and thermal profiles for test cases 1 and 2



(a) Power distribution (b) Thermal profile

(c) Power distribution (d) Thermal profile

Fig. 6: Evaluation for Alpha21264 and Gainestown architectures

have been implemented in C++. R is several times slower than C++, and hence, the implementation of *NanoTherm* in C++ or any other similar language would be faster. In comparison, for calculating the same step response, COMSOL took 3005 s (*NanoTherm* is faster by roughly $724X$). Figure 8 compares the accuracy of our transient simulation against COMSOL, where the error is limited to 1%.

Full Chip Thermal Profile: We compute the transient thermal profile of a $10 \times 10\ mm^2$ chip for the power profile given in Figure 11. The calculated thermal profiles at time instants t = $0.01\ ms$, $2ms$, and $4ms$ are shown in Figure 12, 13, and 14 respectively. An RMS error of $0.057°C$ was observed for t = $4\ ms$. A total simulation time of $4.2\ s$ was observed. To compute the same transient thermal profile COMSOL took 3005 s (speedup of $715X$).

C. BTE Analysis:

We compare the BTE steady state solution against the Fourier steady state solution for a $60\ nm \times 45\ nm \times 20nm$ channel FET. Solving the Fourier equation only results in a maximum error of $1.59°C$ or 53% (Figure 10), similar to [17].

*1) **Steady State**:* We run the simulation for steady state BTE with a $60\ nm \times 45\ nm \times 20nm$ channel FET [10], [12]. A simulation time of 3.3 s was observed for $400 \times 400 \times 200$ grid points. In comparison the steady state simulation in ThermalScope takes $36.76\ min$ (speedup of $641X$). An RMS error of $0.06°C$ was observed against the ThermalScope. Figure 7 compares the results of *NanoTherm* and ThermalScope.

*2) **Transient**:* We used the same steady state configuration for the transient simulation as well. A simulation time of 39.5 s was observed for 200 time steps. We run the same simulation with ThermalScope with the same meshing (as we had for steady state) and the simulation time observed was $48.1\ min$. An RMS error of $0.087°C$ was observed. Figure 9 compares the result of *NanoTherm* and ThermalScope.

D. Simulation Speed

Table II summarizes the time needed to compute the temperature profile for all cases (chip level and nanometer level, steady state and transient) by popular commercial and open source tools. The time taken by COMSOL to obtain the steady state thermal profile was 305 s. Therefore our algorithm is $3652X$ faster than COMSOL. We are 7X faster than ThermalScope in calculating the transient Fourier solution. For BTE solution, we are $668X$ faster than ThermalScope while calculating the steady state profile and 73X faster in computing the transient thermal profile.

TABLE II: Speed of popular simulators

Simulator	Fourier heat eq		BTE	
	Steady	Transient	Steady	Transient
Hotspot[1]	1 s	36 s	-	-
3DICE	1.36 s	1.77 s	-	-
COMSOL	305 s	3005 s	-	-
ThermalScope	11.1 s	32.55 s	2206 s	2888 s
NanoTherm	0.083s	4.2s	3.3s	39.5s

1. For an acceptable accuracy in Hotspot and 3DICE, a grid size of $\sim 128 \times 128$ is used.

2. HotSpot, 3D-ICE and COMSOL do not solve the BTE equations.

V. Conclusion

In this paper, we proposed a fast and analytical thermal estimation technique that solves both the Fourier and BTE equations using novel transform based approaches. As compared to other tools that solve the Fourier equations, we are at least 12X faster for steady state analysis, and within 3X of the best tool for transient analysis. For the BTE based analysis, our tool is 7-668X faster than the state of the art tool, ThermalScope, with an error limited to 3%.

Acknowledgement

This work has been sponsored in part by the Semiconductor Research Corporation (SRC).

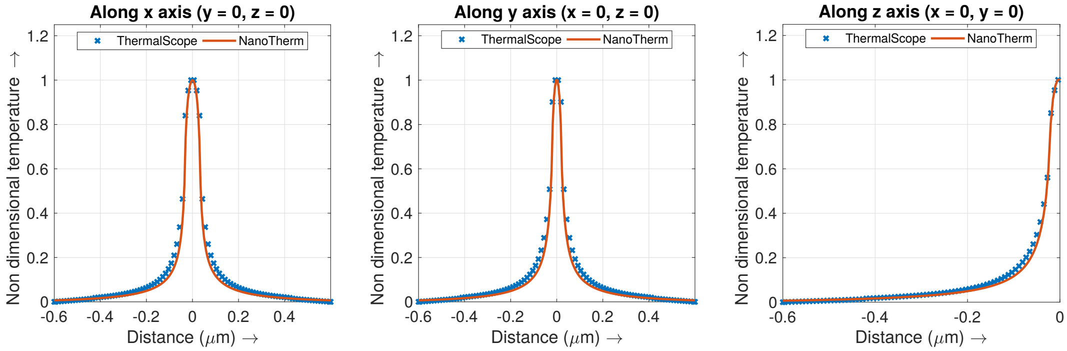


Fig. 7: Comparison of *NanoTherm* and ThermalScope (BTE, steady state)

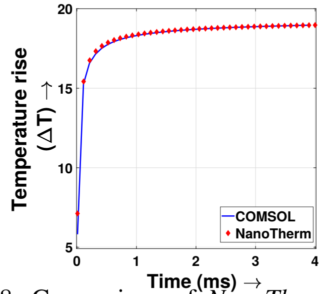


Fig. 8: Comparison of *NanoTherm* and COMSOL (Fourier, transient)

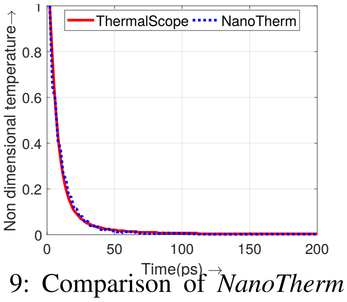


Fig. 9: Comparison of *NanoTherm* and ThermalScope (BTE, transient)

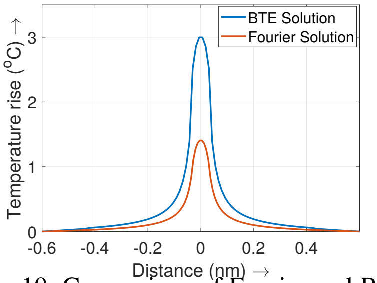


Fig. 10: Comparison of Fourier and BTE (steady state)

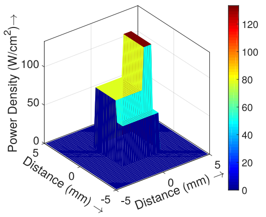


Fig. 11: Power distribution

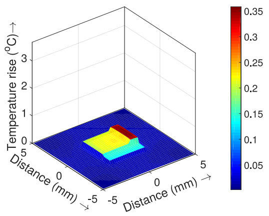


Fig. 12: ΔT at t = 0.01 ms

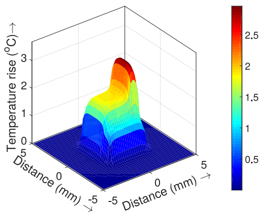


Fig. 13: ΔT at t = 2 ms

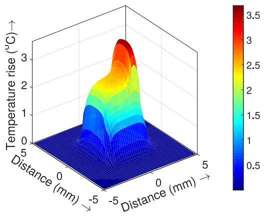


Fig. 14: ΔT at t = 4 ms

References

[1] A. Ziabari, J. Park, E. K. Ardestani, J. Renau, S.-M. Kang, and A. Shakouri, "Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, 2014.

[2] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "Interconnect and thermal-aware floorplanning for 3d microprocessors," in *ISQED*, 2006.

[3] J.-M. Lin, T.-T. Chen, Y.-F. Chang, W.-Y. Chang, Y.-T. Shyu, Y.-J. Chang, and J.-M. Lu, "A fast thermal-aware fixed-outline floorplanning methodology based on analytical models," in *ICCAD 2018*, 2018.

[4] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, " Temperature-Aware Microarchitecture," in *ISCA*, 2003.

[5] J. Park, A. Shakouri, and S. Kang, "Fast evaluation method for transient hot spots in vlsi ics in packages," in *ISQED*, 2008.

[6] S. R. Sarangi, G. Ananthanarayanan, and M. Balakrishnan, "Lightsim: A leakage aware ultrafast temperature simulator," in *ASP-DAC*, 2014.

[7] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling," in *ICCAD*, 2010.

[8] Y. Zhan and S. Sapatnekar, "A high efficiency full-chip thermal simulation algorithm," in *ICCAD*, 2005.

[9] H. Sultan and S. R. Sarangi, "A fast leakage aware thermal simulator for 3d chips," in *DATE*, 2017.

[10] Z. Hassan, N. Allec, F. Yang, L. Shang, R. P. Dick, and X. Zeng, "Full-spectrum spatial–temporal dynamic thermal analysis for nanometer-scale integrated circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 12, pp. 2276–2289, 2011.

[11] C. Hua and A. J. Minnich, "Analytical green's function of the multidimensional frequency-dependent phonon boltzmann equation," *Phys. Rev. B*, vol. 90, p. 214306, Dec 2014.

[12] S. V. Narumanchi, J. Y. Murthy, and C. H. Amon, "Boltzmann transport equation-based thermal modeling approaches for hotspots in microelectronics," *Heat and mass transfer*, vol. 42, no. 6, pp. 478–491, 2006.

[13] S. Mazumder and A. Majumdar, "Monte carlo study of phonon transport in solid thin films including dispersion and polarization," *Journal of Heat Transfer*, vol. 123, no. 4, pp. 749–759, 2001.

[14] A. Nabovati, D. P. Sellan, and C. H. Amon, "On the lattice boltzmann method for phonon transport," *Journal of Computational Physics*, vol. 230, no. 15, 2011.

[15] S. Varshney, H. Sultan, P. Jain, and S. R. Sarangi, "Fourier-bte solution and validation," http://www.cse.iitd.ac.in/~srsarangi/files/thesis/shashank.pdf, 2019.

[16] Y. Zhan and S. S. Sapatnekar, "Fast computation of the temperature distribution in vlsi chips using the discrete cosine transform and table look-up," in *ASPDAC*, 2005.

[17] N. Allec, Z. Hassan, L. Shang, R. P. Dick, and R. Yang, "Thermalscope: Multi-scale thermal analysis for nanometer-scale integrated circuits," in *ICCAD*, 2008.