VarSim: A Fast Process Variation-aware Thermal Modeling Methodology Using Green's Functions

Hameedah Sultan and Smruti R Sarangi

***Abstract*—Despite temperature rise being a first-order design constraint, traditional thermal estimation techniques have severe limitations in modeling critical aspects affecting the temperature in modern-day chips. Existing thermal modeling techniques often ignore the effects of parameter variation, which can lead to significant errors. Such methods also ignore the dependence of conductivity on temperature and its variation. Leakage power is also incorporated inadequately by state-of-the-art techniques. Thermal modeling is a process that has to be repeated at least thousands of times in the design cycle, and hence speed is of utmost importance.**

To overcome these limitations, we propose *VarSim*, an ultrafast thermal simulator based on Green's functions. Green's functions have been shown to be faster than the traditional finite difference and finite element-based approaches but have rarely been employed in thermal modeling. Hence we propose a new Green's function-based method to capture the effects of leakage power as well as process variation analytically. We provide a closed-form solution for the Green's function considering the effects of variation on the process, temperature, and thermal conductivity. In addition, we propose a novel way of dealing with the anisotropicity introduced by process variation by splitting the Green's functions into shift-variant and shift-invariant components. Since our solutions are analytical expressions, we were able to obtain speedups that were several orders of magnitude over and above state-of-the-art proposals with a mean absolute error limited to 4% for a wide range of test cases. Furthermore, our method accurately captures the steady-state as well as the transient variation in temperature.

I. Introduction

The [1] demand for high-performance computing as well as applications in fields such as machine learning, big data analytics, IoT, and edge computing has led to increased power densities in modern-day chips. The resultant temperature rise has several harmful effects that include an increase in leakage power and a disproportionate decrease in reliability. Hence, thermal simulation is now one of the most critical steps in the overall semiconductor design flow. It is typically a long and time-consuming process, that has to be repeated several times for a multitude of use cases in the design cycle.

To make matters worse, process variation has increasingly been leading to large deviations in electrical and thermal parameters of transistors, thereby leading to a high degree of unpredictability in key circuit parameters such as the timing delay and leakage power consumption. With ongoing device scaling, handling and mitigating process variation continues to become increasingly critical. Process variation affects all major architectural design decisions.

Sadly, thermal modeling in chips with process variation is extremely complex and slow; till date no fast and efficient solutions have been proposed; researchers still rely on traditional Finite Element Method (FEM) and Finite Difference Method (FDM) analysis.

There is however a **strong need** for fast thermal simulation methods in this space. Many architectural techniques have been proposed with the aim of mitigating the adverse effects of process variation such as functional unit level body biasing and retiming. However, to effectively incorporate such schemes, an accurate estimate of the impact of variation is needed. This requires extensive thermal simulations for a wide range of power plans. Similarly, while floorplanning or cell placement, a large number of optimization strategies need to be quickly evaluated for a range of process variation scenarios [2], [3]. Leakage power has a strong temperature dependence and is heavily influenced by process variation as well. The chip temperature itself is dependent on leakage power, resulting in a cyclic effect. Thus leakage has a significant impact on the temperature of modern-day chips, often contributing to half of the total temperature rise. **Thus, a proper design optimization requires performing a thorough thermal evaluation through simulations on chips having a significant range of process, conductivity and temperature variation, while correctly incorporating leakage power.** Hence, there is a need for a *fast* thermal simulator in this space. A disclaimer is due – at different stages of the design process the designer has different degrees of information. Nevertheless, there is still a need for ultra-fast thermal simulation because designers always seek the most productive design choices with the information that they have at that stage. For example, at the architectural level, in the product planning stage, just a broad idea of the power consumption is available. Hence, the designer uses high level core power and variation models here, which still need to factor in the effects of temperature. After RTL signoff, placement and routing, synthesis and layout, progressively more accurate power numbers are available at each stage, and any change introduced at any level requires further modeling updates to guide the optimal design choices. This exercise needs to be done for every DCVS corner. After tapeout in the post silicon stage, the exact dynamic power numbers and a concrete idea of the process variation is avail-

Hameedah Sultan is with the School of Information Technology, Indian Institute of Technology, Delhi, India, 110016.
E-mail: hameedah.sultan@gmail.com

Smruti R Sarangi is with the Department of Computer Science and Engineering, joint faculty in the Electrical Engineering department, Indian Institute of Technology, Delhi, 110016.
E-mail: srsarangi@cse.iitd.ac.in

Manuscript received Month dd, 2023; revised Month dd, 2023.

[1]This paper is an extended version of the paper: H. Sultan and S.R. Sarangi, "VarSim: a fast and accurate variability and leakage aware thermal simulator", DAC 2020

able. Thus at every stage of the design process, each simulation produces crisp, exact data, and thousands of such simulations are run with different power dissipation (variation) values for a range of use cases. This Monte Carlo simulation yields a distribution that helps determine the final chip design. This is the standard practice widely used across the semiconductor industry.

Unfortunately, existing **architectural thermal simulators do not consider the effects of process variation**. Ignoring process variation could lead to failure of the device after fabrication [4]. Additionally, most simulators fail to factor in the temperature dependence of conductivity, leading to significant errors in thermal estimation. Prior work has shown that ignoring the temperature dependence of conductivity can result in an error of up to 5°C [5]. Furthermore, traditional thermal simulators are based on the costly finite element and finite difference methods, making them slow and limiting the scope of design space exploration. On top of that, most existing thermal simulators consider the effects of leakage power by iterating through the leakage-temperature feedback loop, increasing their runtime several times. Since leakage power is indispensable in modern-day chips, it is essential for today's thermal simulators to naturally consider the effects of leakage power as part of the core modeling methodology and avoid iterative computations.

Consequently, fast thermal estimation that takes variability into account has hitherto remained an open problem. To solve this problem, we propose a thermal simulation methodology, *VarSim*, in this paper. ***VarSim* is a novel Green's function-based analytical thermal simulator, that inherently captures the effects of both process variation as well as leakage power, without running costly iterations.** Our main contributions can be summarized as follows:

1) *VarSim* considers the impact of process variation as well as the temperature dependence of conductivity. To the extent of our knowledge, no existing technique has done this.
2) Our method is based on Green's functions (impulse response of a power source), which are known to be very fast methods [6], [7]. However, Green's functions in thermal estimation rely on the shift-invariance of the impulse response, which ceases to be true when process variation is considered. We propose a novel way of overcoming this limitation by splitting the Green's function into a shift-invariant component and a shift-variant component.
3) We mathematically derive a novel, modified leakage-aware Green's function that incorporates in itself the impact of temperature-dependent conductivity and leakage power. This modified Green's function is directly used at runtime with the dynamic power profile to obtain an accurate estimate of the full-chip thermal profile considering all the desired effects. Since our method is analytical, it is also extremely fast.
4) Green's function-based methods have great potential, since their accuracy is not dependent on the grid size, making them faster. However, their applicability is limited by a lack of solutions for modern EDA problems. Thus our work contributes to an ecosystem, where more researchers can propose faster Green's function-based solutions for newer problems.

Using an analytical Green's function-based approach, we obtain a several orders of magnitude speedup over state-of-the-art approaches, while keeping the maximum error within 4%. **Thus by leveraging the underlying physics of heat transfer and process variation, we are able to increase the accuracy of thermal simulations by considering all the critical temperature-affecting phenomena, while simultaneously achieving a very high simulation speed.** The speed advantage becomes even more pronounced when thermal simulations need to be repeated thousands of times, as is done in a typical design cycle.

The work presented in this paper is an extension of the work presented in Reference [8]. However, the previous work involved solving the steady-state temperature profile only. In this work, we solve for the transient thermal profile as well, and include this to evaluate our method for both step and time-varying inputs. In addition, we have made major changes to the modeling methodology by removing simplifying assumptions and splitting the Green's functions into shift-varying and shift-invariant parts. We have also removed the use of k-means clustering for better generalizability. In addition, we have validated our method much more thoroughly in the current work.

The rest of the paper is organized as follows. We describe some background information for our work in Section II. The current state-of-the-art literature is described in Section III. We describe our modeling methodology in Section IV. Section V describes the evaluation of our proposed method and the corresponding results. We finally conclude in Section VI.

II. Theoretical Preliminaries

A. Fundamentals of Thermal Modeling

1) The Fourier Equation: The temperature distribution in a body is governed by the Fourier heat equation [9], described mathematically as:

$$\nabla.(\kappa \nabla T) + \dot{q} = \rho C_v \frac{\partial T}{\partial t}, \tag{1}$$

where ρ is the material density, T is the temperature, C_v is the volumetric specific heat, κ is the thermal conductivity, and $\dot{q}$ is the rate of heat energy generation inside the volume. However, this equation is too complex to be solved analytically in the general case. Hence most chip-level thermal simulators use numerical methods such as the finite difference and the finite element methods to arrive at an approximate solution quickly.

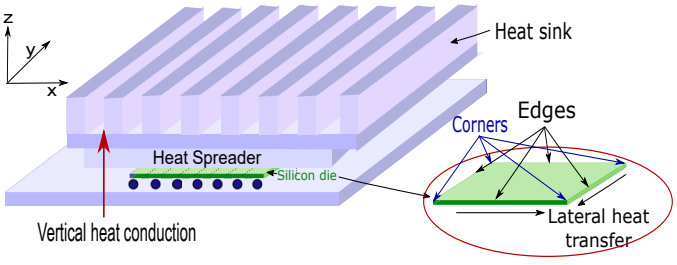


Fig. 1: Schematic of the chip with edges and corners labelled and the direction of heat flow shown

2) Heat Transfer in a Chip's Package: Figure 1 shows the schematic diagram of the chip's package. The chip's die area is typically limited to 400-500 mm^2 (maximum). This is much smaller as compared to other structures in the package. There is of course a lateral heat conduction path where heat flows along the silicon die (in a 2-dimensional fashion). A chip has additional metal layers; however, these layers generate very little heat and furthermore for heat transfer, the die (along with other structures) act as a low pass filter. This means that there is a high spatial correlation of the temperature. As a result, often the die temperature is modeled and also measured (using on-die temperature sensors). Lateral heat conduction is only considered along the transistor layer and lower-interconnect layers (lumped into one layer).

The majority of heat transfer however takes place vertically. Some of it leaks through the bottom (via the ball grid array through the PCB). However, this is not the primary heat transfer path. Most of the heat escapes through the heat spreader. The spreader is a much larger metal plate (typically 5 cm $\times$ 5 cm) made of a Cu-Ni alloy with high thermal conductivity. Its job is to homogenize the temperature profile on the die and also transfer the heat to the heat sink. The heat sink on top has multiple fins to increase its effective surface area. It transfers heat to the ambient where we can either have natural convection or we can force air over it using a fan.

We view the structure primarily as a cylinder. Heat transfer at most points (other than the rim of the die) is isotropic along the silicon surface (same in all directions). For every grid point, there is heat loss in a vertical direction as well. The mathematical abstraction is a 2D-plane with vertical heat loss at each grid point (towards the heat sink). The heat sink is modeled as a thermal resistance to the ambient (assumed to maintain a fixed temperature). The horizontal boundaries are adiabatic (no heat transfer). Note that the horizontal boundaries are the sides of Figure 1.

B. Green's Functions

An alternative approach to obtain the temperature profile is based on the impulse response of the chip (or the Green's function [10]) (refer to Appendix B). This impulse response is obtained by applying a unit power source to the center of the chip and getting the corresponding temperature rise. The Green's function is essentially the heat spread function, f_{sp}, of a point power source. This Green's function is then convolved with the power profile to obtain the full-chip temperature profile. This approach is analytical, and much faster than finite difference or finite element-based approaches since the entire heat transfer path is not modeled, rather only the power dissipating layers and the boundary conditions are considered [11], [6], [7]. This is because the accuracy of the finite element method is dependent on the grid size, and reducing the grid size comes at the expense of lost accuracy. However, because of the analytical nature of the Green's function, the grid size here merely determines the output resolution and the accuracy of the method is independent of the grid size.

Using the impulse response (Green's function), the complete full-chip temperature profile can be calculated as [12]:

$$T = f_{sp} \star P \tag{2}$$

where P is the power dissipation profile, f_{sp} is the Green's function, and $\star$ is the convolution operator.

The Green's function is radially symmetric and can be further decomposed into a rapidly decaying part f_{silic}, and a constant representing heat redistributed through the heat spreader [11].

Interested readers may read more about the Green's function in [11], [6], [7].

C. Process and Temperature Variation

The manufacturing of an IC involves a large number of steps or processes that are imperfect in nature. As a result, the properties of a manufactured chip often differ from its nominal values. The device dimensions have reached the scale of tens of atoms in modern-day chips. As a result, the impact of variation has become much more prominent[13].

The parameters affected by variation include the oxide thickness, threshold voltage, gate width, and channel length. The variation in these parameters is classified as: *wafer-to-wafer*, *die-to-die* and *within-die* variations. The first two effects (collectively known as *inter-die* variation) uniformly affect all regions of a given die. They used to have a larger significance in older technology generations; they can be mitigated easily by relatively simple methods such as *frequency binning*. These typically cause a constant shift in the mean value of a parameter across all the devices on a die.

For newer technology generations, within-die variation dominates and requires more complex management strategies [14]. This type of variation leads to deviations in the electrical and thermal properties of the chip on the same die. Within-die variation is further classified as:

1) **Systematic variations:** These are introduced because of lithographic aberrations and diffraction or chemical-mechanical polishing/planarization (CMP) effects. Systematic variation results in proximate regions on the die having similar values of parameters. It is modeled by a multivariate Gaussian distribution [15] having a spherical correlation 2.

2) **Random variations:** These are caused by random dopant fluctuations[16] (RDF) and line edge roughness; they are together modeled as a zero-mean Gaussian random variable. These variations do not exhibit any spatial correlation.

There are two variables in the heat equation that are strongly affected by parameter variation: leakage power and thermal conductivity. We discuss these next.

D. Leakage power

The variability in leakage power arises because of both systematic and random variations. However, it is well-known [17] that the effects of random variation tend to get averaged out at the architectural level when considering temperature. We have also observed the same in our experiments.

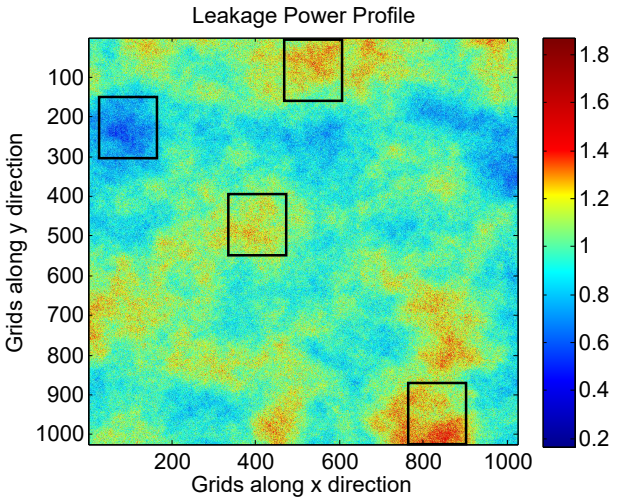


Fig. 2: Leakage power in the presence of variability. Note the spatial correlation in the leakage power values (high concentration of similar values in the bounding boxes).

The subthreshold leakage current, I_{leak} is given by Equation 3.

$$I_{leak} \propto v_T^2 * e^{\frac{V_{GS}-V_{th}-V_{off}}{\eta * v_T}}(1 - e^{\frac{-V_{DS}}{v_T}}) \tag{3}$$

where, v_T is the thermal voltage (kT/q), V_{th} is the threshold voltage, V_{off} is the offset voltage in the sub-threshold region and η is a constant. Because of variability, the oxide thickness and gate length change, which result in a change in the threshold voltage.

The temperature dependence of I_{leak} can be modeled with a reasonable accuracy using a linear equation [11], [6], [18]. Equation 3 then becomes:

$$I_{leak} \propto (1 + \beta\Delta T)e^{\beta_L \Delta L + \beta_{t_{ox}} \Delta t_{ox}} \tag{4}$$

where β represents the change in leakage power with temperature, $\beta_{t_{ox}}$ is a constant representing the variability in the oxide thickness t_{ox} and β_L represents the variability in the gate length, L. The corresponding leakage power is given by:

$$P_{leak} = (1 + \beta\Delta T)P_{leak_0}, \tag{5}$$

where P_{leak_0} is the leakage power at ambient temperature after considering the impact of variability. For improved accuracy, we can use a piece-wise linear leakage model, which provides an accuracy of over 99% [19].

E. Conductivity of Silicon

In addition to considering the impact of variation on leakage power, we also consider the impact of variability on the conductivity of silicon. This is because in accordance with the Fourier's law, the temperature profile is impacted by both the power consumed by the chip as well as the conductivity of the chip. Random dopant fluctuations (RDF) cause a variation in the doping profile, resulting in variations in the conductivity of the material as well. To model this variation in conductivity, we consider a Gaussian random variable, K. The range of variation in the doping profiles because of RDF is obtained from the literature [20]. The range of conductivity values of silicon for these dopant densities is obtained from the literature [21]. Using these, the variance in the conductivity of silicon is then obtained.

In addition to the random variation in conductivity, the conductivity of silicon also depends on temperature, as we will see in Section IV-B (Equations in 9).

The conductivity of silicon varies with temperature according to the following equation:

$$\kappa = k_0 \left(\frac{T}{300}\right)^{-\eta}, \tag{6}$$

where k_0 is the conductivity of silicon at 300K, T is the temperature in Kelvin, and η is a material-dependent constant.

As the chip gets heated, the conductivity of silicon decreases which further affects the temperature profile.

F. Transforms used in this Paper

Thermal problems are often easier to solve in the transform domain. We use two types of transforms in this work – the *Fourier transform* and the *Hankel transform*.

1) Fourier Transform: The Fourier transform decomposes a signal from the spatial domain and brings it into the frequency domain. The result is a complex function, the magnitude of which represents the amount of each frequency present in the signal. In the present work, we make use of the 2-dimensional Fourier transform, which is given by:

$$F(u,v) = \mathcal{F}(f(x,y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{7}$$

where u, v are the Fourier frequency domain variables, x, y are the spatial domain variables, and $f(x, y)$ is the spatial domain signal being transformed into the frequency domain.

2) Hankel Transform: The Hankel transform is equivalent to the 2-D Fourier transform of a radially symmetric function. It uses the Bessel function as its basis. The Hankel transform is defined as:

$$\mathcal{H}(f(r)) = H(s) = \int_0^{\infty} f(r)\mathcal{J}_0(sr)rdr, \tag{8}$$

where $\mathcal{J}_0$ is the Bessel function of the first kind of order 0, and $\mathcal{H}$ denotes the Hankel transform operator.

III. Related Work

Thermal modeling has been a focus area of the EDA industry in the last two decades, and hence researchers have extensively worked on various aspects of this problem, such as 2D and 3D ICs [22], [23], [24], [25], [26], smartphones and other mobile devices [27], thermal-aware DNN accelerators [28].

The fundamental approach of our paper – the Green's function-based method has been used before for thermal modeling [12], [29]. Sultan et. al [29] model temperature in 3D chips with micro-channels using a Green's function approach. They do not consider process variation, temperature-dependent conductivity, and random variation.

Process variation per se has been widely studied along with techniques for mitigating its pernicious effects. However, very rarely has the effect of process variation on temperature been looked at. The works that do consider the effects of process variation on temperature, often do so by neglecting the temperature dependence of leakage power [30]. We demonstrate that considering the effects of process variation on leakage power, but neglecting its temperature dependence may result in a 4 to 6°C error.

Prior works have also established the importance of modeling the temperature-dependence of the conductivity of silicon as well, and proposed methods to tackle the problem. However, such approaches do not consider process variation. Since both of these effects have never been considered simultaneously before, we look at each of these effects in related work separately.

A. Effects of Process Variation on Temperature

Varipower [31] models power variability at the architectural functional unit level by performing circuit-level Monte Carlo simulations incorporating parameter variation. However, it does not model the effects of variability on temperature.

Humenay et al. [17] recognized the challenges imposed by systematic variation in ensuring homogeneous performance across cores. They demonstrated a large variation in power, temperature and performance across cores because of core-to-core systematic variation.

Jaffari and Anis [30] statistically calculated the expected value of temperature considering the impact of variability. They first obtain the leakage-converged temperature iteratively without considering variation and then statistically compute the effect of parameter variation. They use their technique to iteratively update the computed power and temperature to estimate the full-chip power and the probability density function of the temperature distribution. However, a significant limitation of their technique is that it is iterative, making it extremely slow ($\approx 158s$ for a 50×50 grid), 121X slower than *VarSim*.

Lee and Huang [32] develop a statistical electrothermal simulator considering process variations. They solve the statistical steady-state heat transfer equation by partitioning the active layer into grids and model the physical parameters within a grid by KL (Karhunen-Loeve) expansion on the channel length and oxide thickness to simplify the within-die and die-to-die random variables within the grid. Then they sample the KL expanded random variables at a many sampling points. This transforms the statistical heat transfer equation to a deterministic heat transfer equation for each of the sampling points (similar to the classicial FEM approach). They solve this equation to obtain the temperature using a method similar to that used by Huang et al. [33] for each sampling point. They iterate a few times till the temperature values converge. Finally, they use Newton's interpolation polynomial formula to obtain the statistical temperature distribution on the chip.

While their method pretty accurately provides the thermal distribution considering process variation, it is quite expensive in terms of time. This is because it requires solving the heat equation iteratively for each grid point and all the sampling points; this is followed by an additional interpolation step. In contrast, we provide an analytical approach that adapts the parameter values based on their physical location by a convolution operation and Hadamard multiplication, and do not need to separately compute the temperature of each grid point or solve a large number of additional equations. Juan et al. [34] use a linear regression-based model to train and predict the maximum temperature in a 3D IC in the presence of variability. They use measured values of leakage power for training. They demonstrate that 3D ICs are much more susceptible to variation, as compared to 2D ICs. However, learning-based methods are very sensitive to input data and do not generalize well when test conditions change. Additionally, their method captures the maximum temperature only, and not the complete thermal profile.

Sultan et al. [35] use a machine-learning based method for modeling steady-state temperature in 3D chips with microchannels and process variation. However, a key limitation of machine learning methods is that they involve a long and extensive training phase, along with a large amount of computational resources and a comprehensive training dataset. This dataset must be large enough to encompass a broad spectrum of input variations that could potentially occur during runtime, as any deviations from anticipated inputs can yield very inaccurate outcomes. Moreover, machine learning models often do not generalize well to changes in the chip architecture. Also their method is not capable of modeling the transient temperature profile. In cases where a precise analytical solution is possible, it is always favored over the adoption of machine learning methodologies (given accuracy considerations). We provide a detailed comparison of our method against this work in Section V-E.

Jia et al. [36] use the proper orthogonal decomposition (POD) method to simulate the thermal profile specifically in interconnects. They extend the POD methodology such that the effect of material property variation in interconnects is included. However they do not consider process variation, rather material property variation because of a via appearing in the intervening dielectric, which causes a change in the material properties.

Shafique et al. [37] propose a variability-aware dark silicon management technique in which the cores to be throttled are determined on their workload patterns while accounting for the temperature map and variability. They propose a complex heuristic for predicting the temperature profile considering process variation. They superpose the impact of variability-affected leakage power on the estimated temperature map for a given thread-to-core mapping. This heuristic attempts to manually approximate the underlying logic that is captured well in our modified Green's function based approach. Our proposed method achieves this in a precise and efficient mathematical manner using the convolution operations. In comparison to our method, their technique is inexact, unnecessarily complex and slow.

Srinivasa et al. [38] demonstrate using measurements that because of process variation, smartphones of the same model may show a variation of upto 10-12% in energy and perfor-

mance.

B. Modeling the Temperature Dependence of Conductivity

Yang et al. [5] propose a temporally and spatially adaptive thermal analysis technique that accounts for the temperature dependence of conductivity. They assume variable thermal conductivity across all grid cells. This necessitates the maintenance of a large 2D matrix of thermal conductivities. This does not take the temperature dependence into account. After the full simulation is done, the authors propose a post-processing phase. Here, the varied thermal conductivities are calculated (based on the temperature), and a small heuristic-based algorithm is used to "patch up" the final temperature values. Any iterative process at this stage, is considered to be too expensive. This work does not model leakage power, perhaps for the same reason (iterations are expensive).

Li et al. [39] calculate the leakage power in the presence of variation, and use this as an input to the ISAC thermal modeling tool [5]. They additionally consider leakage power, which is iteratively recalculated at each time step until convergence. Finally, they use this augmented tool to study process variation in on-chip networks. This approach is iterative, and requires a large number of iterations to get to accurate variation-aware leakage-converged temperature values. For the sake of comparison, we implement a similar approach using the HotSpot thermal modeling tool [40], and demonstrate several orders of magnitude speedup using our method over such approaches.

Ziabari et al. [41] adopt a different approach. Their approach is a precursor to our approach. They consider the temperature-dependence of conductivity by using a lookup table to store Green's functions with different conductivities. At runtime, they iteratively update the Green's function until the temperature profile converges. They, however, do not model leakage or process variation.

In comparison, our approach encompasses the effects of leakage power, variability in leakage, and temperature-dependent conductivity analytically without requiring costly iterations. Previously proposed tools consider only one of these effects.

C. ML-based and Other Models

There is an extensive body of work in fast transient thermal modeling as well ([42], [43], [44]). These works focus on fast runtime thermal estimation, 2.5 and 3D stacked packages, and thermal models for TSVs but do not take process variation into account.

He et al. [45] propose a novel polynomial chaos for modeling uncertainty at the architectural level using mixed integer programming. Chittamuru et al. [46] demonstrate the sensitivity of photonic networks-on-Chip (NoCs) to thermal and process variation and propose a robust framework to overcome its impact on reliability.

However, process variation, along with the variation of conductivity with temperature, has never been considered before in a leakage-aware thermal simulation tool.

IV. Thermal Estimation Considering Variability

A. Overview

Our approach fundamentally uses the Green's functions to compute the full-chip thermal profile. These can either be theoretically calculated, or empirically obtained by applying a unit impulse power source to the center of the chip, and measuring the corresponding temperature rise. To consider the feedback effect of leakage power on temperature, Sarangi et al. [11] derived a leakage-aware Green's function that captures this temperature-dependence of leakage power. However, there are two effects that have not been considered in the use of these functions for full-chip temperature estimation:

1) The first effect is that the conductivity of silicon is also temperature-dependent, and has a non-negligible effect on temperature. To capture this effect, we derive a novel modified Green's function, that not only captures the temperature-dependence of leakage power, but also captures the temperature-dependence of conductivity.
2) The second unmodeled effect is process variation. Process variation poses multiple challenges in the use of Green's function-based methods. The first challenge is in deriving the Green's function itself considering the effects of process variation as well. In a previous version of this work [8], we had used the expected value of the leakage power map in place of the baseline leakage power map, to simplify the derivation of the modified temperature-dependent Green's function (note that the Green's function itself had captured the baseline variability, and this was not approximated; the approximation was in computing the temperature-dependent part of the leakage power). In the current work, we get rid of this approximation, and work out the complete expression.

The second challenge is in computing the full-chip thermal profile using the derived Green's function. Temperature estimation using the Green's function relies on its *shift-invariance*, which means that a power applied to any location of the chip will cause the same temperature rise, irrespective of the location of the power source on the chip (aside from the boundary effects, which need to be handled separately). This stops being true when process variation is considered. Hence, we split the Green's function into random and deterministic components. Then, we propose a novel way of combining the two components to generate the full-chip temperature profile. The deterministic component of the full-chip temperature profile is computed the regular way, using a convolution operation. The random component is location-dependent and hence is computed using the Hadamard product.

We provide an overview of our proposed approach in Figure **??**. Table I lists the terms used in our derivations.

Thus, our contributions can be summarized as follows:

❶ We derive an analytical expression for a novel leakage aware Green's function accounting for variability and the dependence of leakage power and conductivity on temperature. This Green's function has two components – a shift-invariant component that accounts for the temperature dependence of conductivity and leakage power, and a shift-variant component that accounts for process variation. We derive separate modified Green's functions for the steady-state (Section IV-C) and

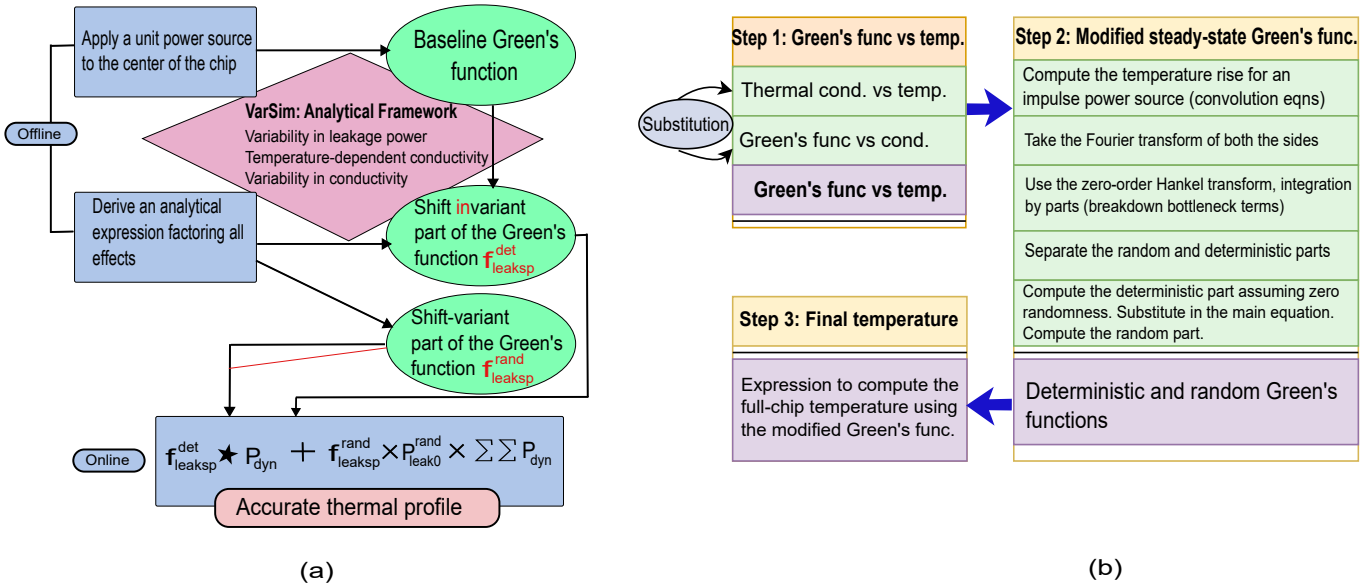


Fig. 3: A high-level overview

the transient case (Section IV-D). Furthermore, we describe the use of the transient Green's function to compute the temperature profile for a time-varying source in Section IV-E. ❷ We calculate the full-chip variability-aware thermal profile in the presence of leakage as well as dynamic power by convolving the shift-invariant modified Green's functions with the dynamic power map and multiplying the shift-varying Green's functions with the baseline leakage power map considering process variation.

TABLE I: Glossary

Symbol	Meaning
P_{leak_0}	Leakage power at ambient temperature considering variability
β	Temperature dependence of leakage power
α	Temperature dependence of conductivity
κ	Conductivity of silicon
T	Temperature
$\mathcal{T}$	Temperature rise above ambient temperature
f_{sp_0}	Green's function without considering leakage and temperature-dependent conductivity $= f_{silic_0} + \phi$
$\mathcal{F}$	Fourier transform operator
$\mathcal{H}$	Hankel transform operator
x, y	Spatial coordinates
u, v	Fourier frequency domain variables
h	Hankel variable
t	Time
C	Thermal capacitance
f^k_{leaksp}	Leakage aware Green's functions considering temperature dependence of conductivity
k'_0	Nominal conductivity of silicon at ambient temperature

B. Variation of the Thermal Conductivity with Temperature

The conductivity of silicon varies with temperature according to the following relation [47]:

$$\kappa = k_0 \left(\frac{T}{300} \right)^{-\eta}, \tag{9}$$

where k_0 is the conductivity of silicon at 300K, T is the temperature in Kelvin, and η is a material-dependent constant. In the operating range of ICs ($40 - 100$°C), we can linearize Equation 9:

$$\kappa(T) = k_0{}'(1 - c\Delta T), \tag{10}$$

where k'_0 is the nominal conductivity of silicon at the ambient temperature, and c is a constant.

Next, we vary the conductivity of silicon and observe the change in the Green's function f_{sp}. We then obtain an empirical relation between the Green's function and conductivity (using HotSpot [48]):

$$f_{sp}(\kappa) = f_{sp_0}(1 - c'(\kappa(T) - k'_0)), \tag{11}$$

where, f_{sp_0} is the baseline Green's function when the temperature dependence of conductivity is not considered (the variation in conductivity because of random dopant fluctuations is captured in f_{sp_0}), and c' is a constant. Combining Equations 10 and 11, we obtain a relation for the Green's function that captures the dependence of conductivity on temperature.

$$f_{sp}(T) = f_{sp_0}(1 + \alpha\Delta T), \tag{12}$$

where α is a constant that captures how much the Green's function varies because of a change in conductivity with temperature.

C. Modified Green's Functions for the Steady-State

1) Formulating the equation for the modified Green's function considering all effects: Next, we derive the Green's function considering temperature-dependent conductivity, as well as temperature-dependent leakage power. We start with an approach that is similar to that adopted by Sarangi et al. [11] while also incorporating the effects of process variation and the temperature dependence of conductivity.

The total power consumption (P) is the sum of the dynamic power (P_{dyn}) and the leakage power (P_{leak}). Using Equation 5, we get:

$$P = P_{dyn} + P_{leak_0}(1 + \beta\Delta T). \tag{13}$$

We assume that the dynamic power dissipation in the chip is initially zero. This implies that any temperature rise above the ambient temperature is because of leakage. From Equation 2, we have (subscript $_0$ refers to the initial state):

$$T_0 = f_{sp_0} \star P_{leak_0} \tag{14}$$

Now, let us apply a unit impulse (Dirac delta function) dynamic power source to the center of the chip. Using the results in Equations 2, 12, and 13 we get the updated temperature, T_f:

$$T_f = f_{sp_0}(1+\alpha\Delta T) \star (\delta(x,y) + P_{leak_0}(1+\beta\Delta T)) \tag{15}$$

where x, y are the spatial coordinates.

Next, we find out the increase in temperature because of the unit power source considering process variation and the temperature-dependent effects. We also use the property that the convolution of a function and a delta function is the function itself. We then arrive at:

$$\begin{aligned}\mathcal{T} &= T_f - T_0 \\ &= f_{sp_0}(1+\alpha\mathcal{T}) + f_{sp_0}(1+\alpha\mathcal{T}) \star P_{leak_0}(1+\beta\mathcal{T}) \\ &- f_{sp_0} \star P_{leak_0}\end{aligned} \tag{16}$$

We need to solve for the temperature rise, $\mathcal{T}$, here. To convert the convolution operation into multiplication, we compute the Fourier transform on both sides and apply the property that the Fourier transform of the convolution of two functions is equal to the product of their individual Fourier transforms. We compute the Fourier transform of both sides of Equation 16 to arrive at Equation 17.

In Equation 17, $\mathcal{F}$ is the Fourier transform operator, term I is the baseline Green's function, term II corresponds to the increase in temperature because of the temperature-dependence of conductivity, term III is the increase in temperature because of the temperature-dependence of leakage power, term IV corresponds to the compounded effect of the baseline leakage power and the temperature-dependence of the conductivity, and term V is the increase in temperature because of the compounded effects of temperature-dependent conductivity and leakage. The last term here is small because each of the temperature-dependent variables (conductivity/leakage power) by itself do not cause a large enough change in the other variable to result in a large temperature change. Hence, we neglect this term.

2) Reducing the bottleneck term (Term II): The most difficult term to compute in the above equation is $\mathcal{F}(f_{sp_0}\mathcal{T})$. Let $G(u,v) = \mathcal{F}(f_{sp_0}\mathcal{T})$. For this we make use of Lemma I described below:

Lemma I: : $G(u,v) = \mathcal{F}(f_{sp_0}\mathcal{T}) = \mathcal{F}(\mathcal{T})g_{sp_0}$, where $g_{sp_0} = (f_{sp_0} - \kappa + f_{sp_0}(0,0))$

Interested readers may refer to the proof of Lemma 1 in Appendix A.

Using Lemma I in Equation 17, we have:

$$\begin{aligned}\mathcal{F}(\mathcal{T}) = \mathcal{F}(f_{sp_0}) &+ \alpha g_{sp_0}\mathcal{F}(\mathcal{T}) + \beta\mathcal{F}(f_{sp_0})\mathcal{F}(P_{leak_0}\mathcal{T}) \\ &+ \alpha g_{sp_0}\mathcal{F}(P_{leak_0})\mathcal{F}(\mathcal{T})\end{aligned} \tag{18}$$

3) Separating the random and deterministic terms: The modified Green's function considering variability (solution of Equation 18) is not shift-invariant because of process variation (terms III and IV), which means that we will not be able to directly convolve the modified Green's function with a power profile without a loss of accuracy. To overcome this limitation, we split the modified Green's function into two components: a deterministic component f^{det}_{leaksp}, which is shift-invariant and is obtained by assuming the variability to be zero (replacing the baseline leakage power profile with its mean value), and a random component f^{rand}_{leaksp}, which is shift-variant and accounts for all the variation in leakage power.

To arrive at the respective expressions for the modified Green's functions, we first split the variable leakage power, P_{leak_0}, into two components – a constant equal to its mean (μ) and a randomly varying part $P^{var}_{leak_0}$. Thus, $P_{leak_0} = \mu + P^{var}_{leak_0}$. Using this in Equation 18, we get:

$$\begin{aligned}\mathcal{F}(\mathcal{T}) &= \mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T}) + \beta\mathcal{F}(f_{sp_0})\mathcal{F}\big((\mu + P^{var}_{leak_0})\mathcal{T}\big) \\ &+ \alpha g_{sp_0}\mathcal{F}\left(\mu + P^{var}_{leak_0}\right)\mathcal{F}(\mathcal{T}) \\ &= \mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T}) + \mu\beta\mathcal{F}(f_{sp_0})\mathcal{F}(\mathcal{T}) \\ &+ \beta\mathcal{F}(f_{sp_0})\mathcal{F}(P^{var}_{leak_0}\mathcal{T}) + \alpha g_{sp_0}\mathcal{F}(\mu)\mathcal{F}(\mathcal{T}) \\ &+ \alpha g_{sp_0}\mathcal{F}(P^{var}_{leak_0})\mathcal{F}(\mathcal{T})\end{aligned} \tag{19}$$

Now, the temperature profile $\mathcal{T}$ is itself composed of a deterministic and a variable part, $\mathcal{T} = \mathcal{T}^{det} + \mathcal{T}^{var}$. The deterministic part can be obtained by assuming the variability to be zero, $P^{var}_{leak_0} = 0$. Applying this to Equation 19, we arrive at Equation 20:

$$\begin{aligned}\mathcal{F}(\mathcal{T}^{det}) = \mathcal{F}(f_{sp_0}) &+ \alpha g_{sp_0}\mathcal{F}(\mathcal{T}^{det}) + \mu\beta\mathcal{F}(f_{sp_0})\mathcal{F}(\mathcal{T}^{det}) + \\ &\alpha g_{sp_0}\mathcal{F}(\mu)\mathcal{F}(\mathcal{T}^{det})\end{aligned} \tag{20}$$

$$\mathcal{F}(\mathcal{T}^{det}) = \mathcal{F}(f^{det}_{leaksp}) = \frac{\mathcal{F}(f_{sp_0})}{1 - \alpha g_{sp_0}(1+\mathcal{F}(\mu)) - \mu\beta\mathcal{F}(f_{sp_0})} \tag{21}$$

Next, we use an equation similar to Equation 48 for $\mathcal{F}(P^{var}_{leak_0}\mathcal{T})$. We get:

$$\begin{aligned}\mathcal{F}(P^{var}_{leak_0}\mathcal{T}) &= \mathcal{F}(\mathcal{T})\underbrace{\big(P^{var}_{leak_0} - P^{var}_{leak_0}(\infty,\infty) + P^{var}_{leak_0}(0,0)\big)}_{=Q^{rand}_{leak_0}} \\ &= \mathcal{F}(\mathcal{T})Q^{rand}_{leak_0}\end{aligned} \tag{22}$$

Substituting Equation 22 and $\mathcal{T} = \mathcal{T}^{det} + \mathcal{T}^{var}$ in Equation 19 and simplifying, we finally get:

$$\begin{aligned}\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var}) = &\mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var}) \\ &+ \mu\beta\mathcal{F}(f_{sp_0})\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var}) + \\ &\beta\mathcal{F}(f_{sp_0})\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var})Q^{rand}_{leak_0} \\ &+ \alpha g_{sp_0}\mathcal{F}(\mu)\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var}) \\ &+ \alpha g_{sp_0}\mathcal{F}(P^{var}_{leak_0})\mathcal{F}(\mathcal{T}^{det} + \mathcal{T}^{var})\end{aligned} \tag{23}$$

$$\begin{aligned}\mathcal{F}(\mathcal{T}) &= (\mathcal{F}(f_{sp_0}) + \alpha\mathcal{F}(f_{sp_0}\mathcal{T})) + (\mathcal{F}(f_{sp_0}) + \alpha\mathcal{F}(f_{sp_0}\mathcal{T})) \times \\ &\quad (\mathcal{F}(P_{leak_0}) + \beta\mathcal{F}(P_{leak_0}\mathcal{T})) - \mathcal{F}(f_{sp_0})\mathcal{F}(P_{leak_0}) \\ &= \underbrace{\mathcal{F}(f_{sp_0})}_{I} + \underbrace{\alpha\mathcal{F}(f_{sp_0}\mathcal{T})}_{II} + \underbrace{\beta\mathcal{F}(f_{sp_0})\mathcal{F}(P_{leak_0}\mathcal{T})}_{III} \\ &\quad + \underbrace{\alpha\mathcal{F}(f_{sp_0}\mathcal{T})\mathcal{F}(P_{leak_0})}_{IV} + \underbrace{\alpha\beta\mathcal{F}(P_{leak_0}\mathcal{T})\mathcal{F}(f_{sp_0}\mathcal{T})}_{V}\end{aligned} \tag{17}$$

$$\begin{aligned}&\mathcal{F}(\mathcal{T}^{det})\left(-\alpha\left(1+\mathcal{F}(\mu)\right)g_{sp_0} - \mu\beta\mathcal{F}(f_{sp_0}) - \right. \\ &\qquad \left.\beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0} - \alpha\mathcal{F}(P^{var}_{leak_0})g_{sp_0}\right) + \\ &\mathcal{F}(\mathcal{T}^{var})\left(1-\alpha(1+\mathcal{F}(\mu))g_{sp_0} - \mu\beta\mathcal{F}(f_{sp_0})\right. \\ &\qquad \left. - \beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0} - \alpha\mathcal{F}(P^{var}_{leak_0})g_{sp_0}\right) \\ &\qquad = \mathcal{F}(f_{sp_0})\end{aligned} \tag{24}$$

Now, we substitute the expression for $\mathcal{F}(f_{sp_0})$ from Equation 20 in Equation 24. After cancelling the common terms (shaded/green ones) we get:

$$\begin{aligned}&\mathcal{F}(\mathcal{T}^{var})\left(1-\alpha(1+\mathcal{F}(\mu))g_{sp_0} - \mu\beta\mathcal{F}(f_{sp_0}) - \beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0}\right. \\ &\qquad \left. - \alpha\mathcal{F}(P^{var}_{leak_0})g_{sp_0}\right) \\ &\qquad = \mathcal{F}(\mathcal{T}^{det})\left(\beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0} + \alpha\mathcal{F}(P^{var}_{leak_0})g_{sp_0}\right)\end{aligned} \tag{25}$$

$$\begin{aligned}&\mathcal{F}(\mathcal{T}^{var}) = \mathcal{F}(f^{rand}_{leaksp}) \\ &= \mathcal{F}(\mathcal{T}^{det}) \times \frac{\left(\beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0} + \alpha\mathcal{F}(P^{var}_{leak_0})g_{sp_0}\right)}{\left(1 - \alpha\Big(1 + \mathcal{F}(\mu) + \mathcal{F}(P^{var}_{leak_0})\Big)g_{sp_0}\right.} \\ &\qquad \left. -\mu\beta\mathcal{F}(f_{sp_0}) - \beta\mathcal{F}(f_{sp_0})Q^{rand}_{leak_0}\right)\end{aligned} \tag{26}$$

The deterministic part of the Green's function remains the same for every variation profile. Hence, it needs to be computed once for a given chip only. The random part needs to be recomputed every time we get a new variation map.

4) Full-chip steady-state thermal profile: The total temperature profile is a sum of the random and deterministic components. The standard approach to compute the deterministic thermal profile is to convolve the deterministic Green's function (Equation 21) with the respective power profile. The random component of the thermal profile is not shift-invariant and cannot be computed using the convolution operation, since it is location-dependent. Hence, we compute the Hadamard product of the random component with the leakage power profile and scale it by the total dynamic power applied to the chip. This is an approximation that we justify empirically after conducting exhaustive experiments.

Thus the total thermal profile is given by:

$$\mathcal{T} = f^{det}_{leaksp} \star P_{dyn} + f^{rand}_{leaksp} * P^{var}_{leak_0} * \sum_{i=1}^{n}\sum_{j=1}^{n} P_{dyn(i,j)} \tag{27}$$

where $*$ represents the Hadamard product, $\star$ represents the convolution operation, and n represents the number of grid points in the chip in one direction.

D. Modified Green's function for the Transient Case

Next, we look at the temporal evolution of temperature. Because of the complexity involved in obtaining the transient solution, we do not split the transient Green's function into shift-variant and shift-invariant components.

The basic transient Green's function equation is given by [11]:

$$\mathcal{T} = f_{sp} \star P - C f_{sp} \star \frac{\partial \mathcal{T}}{\partial t}, \tag{28}$$

where C is the thermal capacitance.

Proceeding in the same manner as the steady-state solution, and looking at Equation 48 (see Appendix A), we arrive at the following equation for the transient case:

$$\begin{aligned}&\mathcal{F}(\mathcal{T}) = \mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T}) + \beta\mathcal{F}(f_{sp_0})\mathcal{F}(P_{leak_0}\mathcal{T}) + \\ &\alpha g_{sp_0}\mathcal{F}(P_{leak_0})\mathcal{F}(\mathcal{T}) - C\big(\mathcal{F}(f_{sp_0}) + \alpha\underbrace{\mathcal{F}(f_{sp_0}\mathcal{T})}_{G(u,v)}\big)\mathcal{F}\left(\frac{\partial\mathcal{T}}{\partial t}\right) \\ &= \mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T}) + \beta\mathcal{F}(f_{sp_0})\mathcal{F}(P_{leak_0}\mathcal{T}) + \\ &\alpha g_{sp_0}\mathcal{F}(P_{leak_0})\mathcal{F}(\mathcal{T}) - C\left(\mathcal{F}(f_{sp_0}) + \alpha g_{sp_0}\mathcal{F}(\mathcal{T})\right)\mathcal{F}\left(\frac{\partial\mathcal{T}}{\partial t}\right)\end{aligned} \tag{29}$$

The first three terms on the RHS correspond to the steady-state temperature profile, $\mathcal{T}_{ss}$. Thus we have:

$$\begin{aligned}\mathcal{F}(\mathcal{T}) &= \mathcal{F}(\mathcal{T}_{ss}) - C\big(\mathcal{F}(f_{sp_0}) + \alpha\mathcal{F}(\mathcal{T})g_{sp_0}\big)\mathcal{F}\left(\frac{\partial\mathcal{T}}{\partial t}\right) \\ &= \mathcal{F}(\mathcal{T}_{ss}) - C\mathcal{F}(f_{sp_0})\left(\frac{\partial\mathcal{F}(\mathcal{T})}{\partial t}\right) \\ &\quad - \alpha C\mathcal{F}(\mathcal{T})g_{sp_0}\left(\frac{\partial\mathcal{F}(\mathcal{T})}{\partial t}\right)\end{aligned} \tag{30}$$

The shaded term is of the form $\mathcal{F}(\mathcal{T})\frac{\partial\mathcal{F}(\mathcal{T})}{\partial t}$, making the solution complex.

We separate the partial derivative term next:

$$\frac{\partial\mathcal{F}(\mathcal{T})}{\partial t} = -\frac{\mathcal{F}(\mathcal{T}) - \mathcal{F}(\mathcal{T}_{ss})}{C\mathcal{F}(f_{sp_0}) + \alpha C\mathcal{F}(\mathcal{T})g_{sp_0}} \tag{31}$$

Separating the variables and replacing partial derivatives with total derivatives (since there is only one variable):

$$\frac{C\mathcal{F}(f_{sp_0}) + \alpha C\mathcal{F}(\mathcal{T})g_{sp_0}}{\mathcal{F}(\mathcal{T}) - \mathcal{F}(\mathcal{T}_{ss})}d\mathcal{F}(\mathcal{T}) = -dt \tag{32}$$

Integrating on both sides:

$$C\mathcal{F}(f_{sp_0})ln(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))+\alpha C g_{sp_0}(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))+ \alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})ln(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))=-(t+C_1)$$

$$ln(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))\big(C\mathcal{F}(f_{sp_0})+\alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})\big) + \alpha C g_{sp_0}(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))=-(t+C_1) \tag{33}$$

We need to solve for $\mathcal{F}(\mathcal{T})$ here. We observe that the last term on the LHS is small since both α and C are small numbers. Hence, we ignore this term.

Thus we have:

$$ln(\mathcal{F}(\mathcal{T})-\mathcal{F}(\mathcal{T}_{ss}))\big(C\mathcal{F}(f_{sp_0})+\alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})\big)=-(t+C_1) \tag{34}$$

Taking the exponential on both sides and simplifying:

$$\begin{aligned}\mathcal{F}(\mathcal{T}) &= \mathcal{F}(\mathcal{T}_{ss})+e^{-\frac{t+C_1}{C\mathcal{F}(f_{sp_0})+\alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})}} \\ &= \mathcal{F}(\mathcal{T}_{ss})+k_1 e^{-\frac{t}{C\mathcal{F}(f_{sp_0})+\alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})}}\end{aligned} \tag{35}$$

where k_1 is a constant.

To compute k_1, we see that at $t=0$, the temperature rise $\mathcal{T}$ is zero. Substituting these in Equation 35, we get $k_1 = -\mathcal{F}(\mathcal{T}_{ss})$.

Thus the final transient leakage and variation-aware Green's function is given by:

$$\boxed{\mathcal{F}(\mathcal{T})=f^{tran}_{leaksp}=\mathcal{F}(\mathcal{T}_{ss})\big(1-e^{-\frac{t}{C\mathcal{F}(f_{sp_0})+\alpha C g_{sp_0}\mathcal{F}(\mathcal{T}_{ss})}}\big)} \tag{36}$$

E. Full-chip Transient Thermal Profile for Time-varying Sources

Next, we use the transient Green's function to compute the full-chip thermal profile corresponding to a time-varying power profile. Note that the transient Green's function is a 3D function of space as well as time.

We observe that the thermal response decays to under 1% of its peak value within 5 ms after a power source is removed. Thus, we conclude that incorporating the thermal response corresponding to the power sources in the last 5 ms only should be sufficient to attain a reasonable accuracy. In the general case, we need to consider the power sources from the last k time instants.

Let $P(t_i)$ denote the instantaneous power profile at time t_i, and let $\mathcal{T}(t_i)$ be the temperature profile. The Green's function is obtained by applying a power source of unit magnitude and 1 ms width at $t=0$ at the center of the chip. The resulting thermal response is measured for the entire chip at intervals of 1 ms, resulting in a 3D tensor.

We start with an initial temperature T_0, and at every time instant, we either increase or decrease the temperature depending on the change in power values in each time interval. To determine the amount of change in temperature, we compute a difference of the power profiles at time instant t_i and t_{i-1} and convolve this with the corresponding leakage-aware Green's function sampled at t_i. We do this for each of the last k time steps, and sum the effects up. Beyond k time steps, the power values in the past would have achieved steady-state and their effects would already have been considered. Let us first describe our full-chip transient estimation approach mathematically for a 2D chip without leakage. Without any loss of generality, let us assume the initial temperature T_0 to be zero.

Let the Green's function at time instant t_0 be denoted by $f_{sp}(x,y,t_0)$. Let the power dissipation profile be $P(x,y,t)$.

Let us apply a power source at $t=0$, and continue to do so at every subsequent time instant: $t=1$ (t_1), $t=2$ (t_2), $t=3$ (t_3), ..., $t=n$ (t_n).

$$\begin{aligned}\mathcal{T}(t_1) &= fsp(t_1)\star(P(t_1)-P(t_0)) \\ &= fsp(t_1)\star(P(t_1)-0) \\ \mathcal{T}(t_2) &= fsp(t_1)\star(P(t_2)-P(t_1))+fsp(t_2)\star(P(t_1)-0) \\ \mathcal{T}(t_3) &= fsp(t_1)\star(P(t_3)-P(t_2))+fsp(t_2)\star(P(t_2)-P(t_1)) \\ &\quad + fsp(t_3)\star(P(t_1)-0) \\ &\ldots \\ \mathcal{T}(t_n) &= fsp(t_1)\star(P(t_n)-P(t_{n-1}))+fsp(t_2)\star(P(t_{n-1})- \\ &\quad P(t_{n-2}))+\ldots+fsp(t_5)\star(P(t_{n-4})-P(t_{n-5}))+ \\ &\quad fsp(t_6)\star(P(t_{n-5})-P(t_{n-6}))+\ldots+fsp(t_n)\star \\ &\quad (P(t_1)-P(t_0))\end{aligned} \tag{37}$$

Now, beyond 5 ms, we assume that the step response saturates.

Thus $fsp(t_5)=fsp(t_6)=fsp(t_7)....=fsp(t_\infty)$.

Equation 37 reduces to:

$$\begin{aligned}\mathcal{T}(t_n) \approx\ & fsp(t_1)\star(P(t_n)-P(t_{n-1}))+fsp(t_2)\star \\ & (P(t_{n-1})-P(t_{n-2}))+\ldots+fsp(t_\infty)\star \\ & \underbrace{(P(t_{n-4})-P(t_{n-5})+P(t_{n-5})-P(t_{n-6})+\ldots+P(t_1)-P(t_0))}_{=P(t_{n-4})}\end{aligned} \tag{38}$$

We can see that all terms cancel each other in the third convolution term, and only $P(t_{n-4})$ remains.

Thus we can calculate the transient thermal profile by considering the last 5 time instants only. To incorporate leakage, we simply need to substitute the *modified leakage aware* Green's functions in place of the basic Green's function.

F. Thermal Estimation at the Edges and Corners

In the Green's function approach, the edges and corners have to be handled separately. The standard procedure to do so is to use an analogy with the method of images from electromagnetics (also used in [12]). In this approach, the power matrix is extended to twice its size and padded with mirror image sources on the other side of the boundary at an equal distance from the edge. To compute the full-chip thermal profile, we convolve the modified deterministic Green's functions with the dynamic power profile and multiply the random Green's function with the baseline leakage power profile (Equation 27).

V. Evaluation

A. Setup

We use an augmented version of the thermal modeling tool Hotspot [48] to carry out simulations with variable leakage power and conductivity. The scripts to invoke HotSpot has been written in R. We run all our HotSpot simulations on an Intel i7-7700 4-core CPU running Ubuntu 16.04 with 16 GB of RAM. We implemented and tested our proposed algorithm in Matlab on a Windows 8 desktop with an Intel i7-2600S processor and 8 GB of RAM. We discretized the chip into a 64×64 grid. The parameters of the modeled chip are given in Table II.

TABLE II: Parameters of the chip [48]

Parameter	Value
No. of grid points per layer, n	64×64
β	0.0275
Die size	100 mm^2
Die thickness	0.15 mm
Nominal Silicon conductivity	130 W/m-K
TIM thickness	0.02 mm
TIM conductivity	4 W/m-K
Spreader thickness	3.5 mm
Spreader conductivity	400 W/m-K
Ambient temperature	318.15K
α	0.0021
k_0	135 W/m-K

Error Metric: We use the mean absolute error and the percentage error relative to the maximum temperature **rise** (calculated temperature minus the ambient temperature) as the error metric. Other thermal modeling tools often report errors relative to the absolute maximum temperature in the die, which under-represents the error [12], [5].

B. Calibration of the Setup

To calibrate our HotSpot setup, we use the commercial CFD software Ansys Icepak. It is an industry-standard tool widely used for high-accuracy thermal simulations. We model an identical layout in HotSpot and Icepak and compare the temperature values obtained from the two tools. We find that the normalized temperature values obtained using both of these tools conform well (within 1.5%).

C. Steady-State Results

1) Modified Green's functions for the steady-state: The first step in our proposed method is to compute the modified Green's functions that we have derived in our work. To do so, we first need the baseline leakage power map considering variation at ambient temperature. To obtain the variation-aware baseline leakage power maps, we use the popular variation modeling tool, *Varius* [15]. Using a similar approach, we model the randomness in conductivity values because of random dopant fluctuations.

Next, we obtain the corresponding baseline Green's functions using a modified version of HotSpot. We apply a unit impulse power source to the center of the chip and obtain the baseline Green's function considering the variability in conductivity. We then use Equation 21 to obtain the deterministic part of the modified Green's function accounting for the effects of temperature-dependent conductivity and leakage power. Our approach takes $0.55\ ms$ to compute the deterministic modified Green's function. Next, we compute the random part of the Green's function using Equation 26. This takes a further $1.86\ ms$. This part needs to be computed once for a chip.

Simulator	Considering $\kappa(T)$	Without considering $\kappa(T)$
Hotspot[2]	$18\ minutes$	$4\ s$
3D-ICE	–	$1.36\ s$
Icepak	$15\ minutes$	$15\ minutes$
Jaffari et al.[30]	–	$158\ s$
Sultan et al.[3][35]	$0.65\ s$	$0.65\ s$
VarSim	$2.9\ ms$	$2.9\ ms$
VarSim det. Green's func. (Offline)	$0.55\ ms$	$0.55\ ms$
VarSim rand. Green's func. (Offline)	$1.6\ ms$	$1.6\ ms$
VarSim full-chip (Online)	$0.74\ ms$	$0.74\ ms$

1. To model temperature-dependent conductivity, detailed thermal modeling is done in HotSpot, since the properties of each block are different.
2. HotSpot, 3D-ICE and Icepak do not consider variability
3. Although the inference time is small in Sultan et al. [35], the training time is extremely large and resource-intensive.

TABLE III: Speed of the studied simulators

2) Full-chip steady-state thermal simulations: At runtime, the dynamic power profile is mirrored and the full-chip thermal profile is computed using the calculated Green's functions using Equation 27. This step takes an additional $0.74\ ms$. Thus the total time taken by our algorithm is $2.89\ ms$ (online time = $0.74\ ms$ + deterministic Green's function computation = $0.55\ ms$ + random Green's function computation = $1.6\ ms$). The mean absolute error is limited to 2% (as demonstrated by multiple test cases, Table V).

To validate our proposal, we adopt the following approach: the leakage power obtained from Varius is added to the dynamic power profile, and HotSpot is invoked iteratively. After each iteration, we update the leakage power and conductivity values based on the current temperature. We keep iterating until the temperature values converge. HotSpot supports modeling of variable conductivity only when detailed 3D modeling is enabled, since different conductivity values for different blocks result in a change of the parameters of the differential equation from block to block. As a result, HotSpot requires $18\ min$ to compute the final temperature. If we do not model variable conductivity, the simulation completes within 4s. Thus, our method provides a $370000\times$ speedup over HotSpot in steady-state thermal simulation.

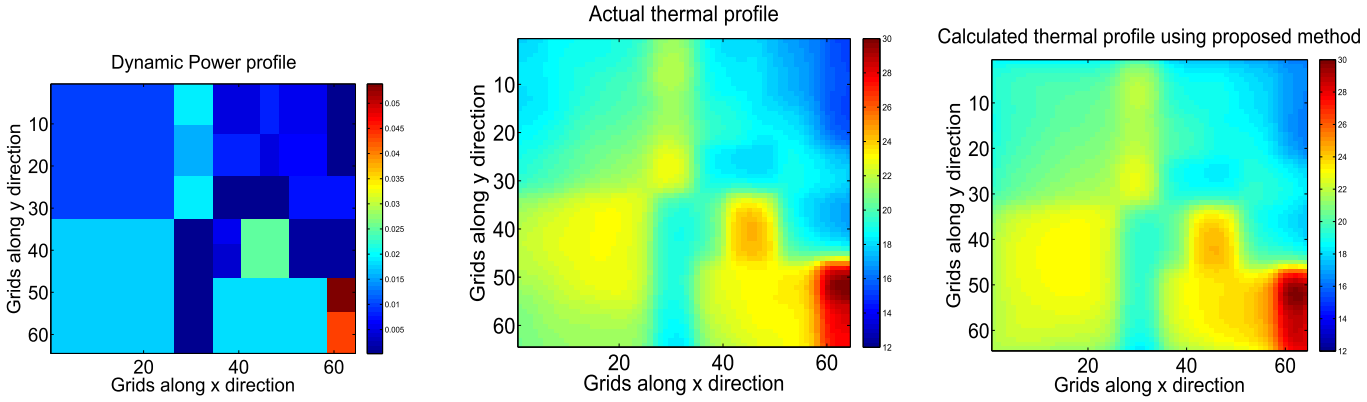
Test Case 1 [Real floorplan]: We validate our approach using the floorplan of the Alpha21264 processor [1]. The power values are taken from the *ev6* test case of HotSpot. The leakage and dynamic power profile and the corresponding temperature profiles are shown in Figure 4. We can see in Figure 4b that the calculated thermal profile matches the actual thermal profile very well (mean absolute error = $0.36°$C , i.e., within 2%).

Test Case 2 [Stress testing]: In this case, multiple dynamic power sources are applied to different locations on the chip. The total dynamic power is $8\ W$. Although the total power

TABLE IV: Errors in various scenarios

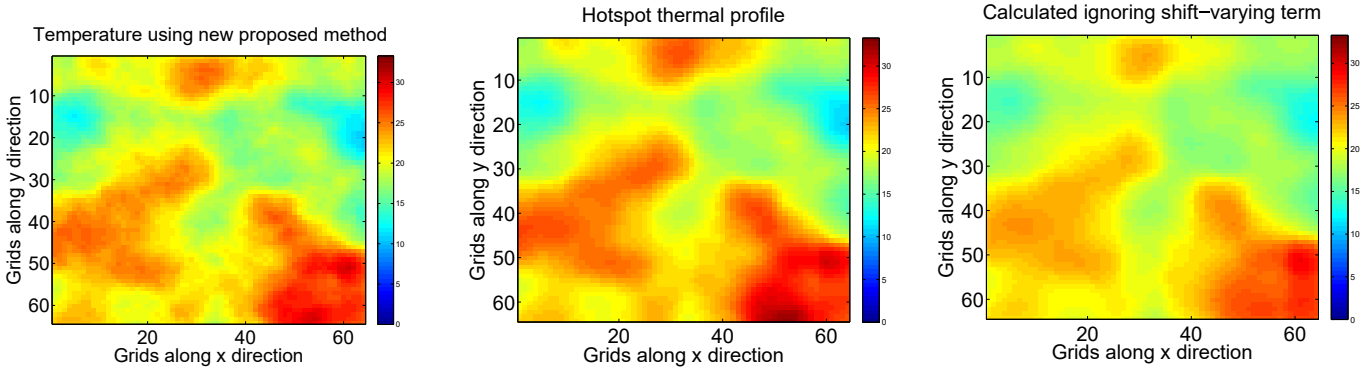
Effects considered	Test Case 1 (Alpha21264)			Test Case 2		
	Max. Temp. (K)	Max. Deviation (K)	Percent Deviation	Max. Temp. (K)	Max. Deviation (K)	Percent Deviation
No effects	341.36	6.77	22.6	372.90	9.01	14.1
Rand.-cond., Cond.(T)	341.86	6.27	20.9	377.59	4.32	6.8
Leakage-var	344.04	4.09	13.6	375.29	6.62	11.6
Leakage(T)	344.30	3.83	12.8	374.56	7.35	12.2
Rand.-cond., Cond.(T), Leakage(T)	344.91	3.22	10.7	378.39	3.52	5.8
Leakage-var, Leakage(T)	347.43	0.70	2.33	377.16	4.75	7.5
Cond.(T), Leakage-var, Leakage(T)	348.12	0.01	0.03	381.36	0.55	0.86
Rand.-cond., Cond.(T), Leakage-var, Leakage(T)	348.13	–	–	381.91	–	–
VarSim	**347.58**	**0.55**	**1.8**	**379.31**	**2.60**	**4.1**

Leakage(T) = temperature-dep. leakage, Leakage-var = variability in leakage, Rand.-cond. = random conductivity, cond.(T) = temperature-dep. conductivity



(a) Dynamic Power (b) Actual thermal profile (c) Calculated thermal profile

Fig. 4: Evaluation for test case 1: Alpha21264



(a) Calculated thermal profile (b) HotSpot thermal profile (c) Calculated thermal profile without f_{leaksp}^{rand}

Fig. 5: Evaluation for test case 3: high variance

applied is lower than test case 1, the power density of the sources is much higher, resulting in a higher maximum temperature. In this case, too, the temperature obtained using our algorithm matches the actual value very well, with a mean absolute error limited to 0.7°C (1.1%).

Test Case 3 [Variance testing]: In this case, we apply the same dynamic power as test case 1, but the baseline leakage power has a higher variance. The calculated thermal profile is shown in Figure 5a, while the corresponding thermal profile obtained from HotSpot is shown in Figure 5b. This test case is a limit study. We see that here, too, the calculated and the actual thermal profiles match closely. The mean absolute error is 0.61°C for a maximum temperature rise of 32.3°C (1.9%), while the maximum temperature is 77.3°C . The maximum error at the hotspot location is 2.8°C . In Figure 5c, we show the temperature profile if the random component of the solution is ignored (Equation 27). We see that the errors are larger in this case, and the main hotspot location is missed. The mean absolute error upon ignoring the random effects is 1.2°C (3.7%), while the maximum error is 5.9°C (18.3%). Thus, we are able to lower the error in modeling process variation-aware thermal profile by up to 52% using our proposed approach.

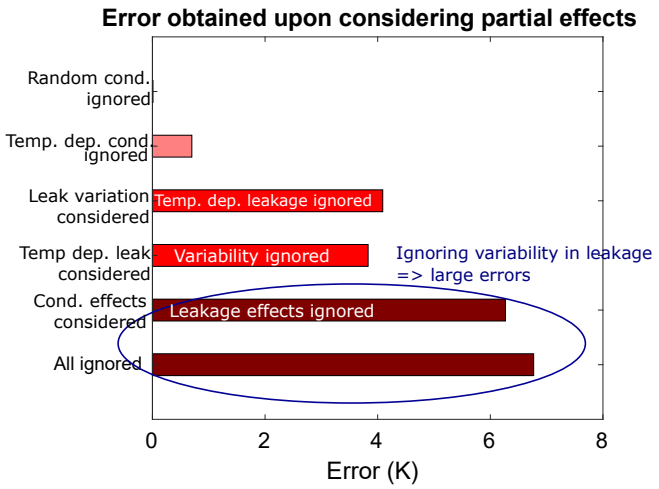


Fig. 6: Error in various scenarios: considering variability in leakage power is as important as considering the temperature dependence of leakage

Moreover, our proposed method helps in capturing the location of the hotspot much more accurately, which would otherwise get missed.

Test Case 4 [Variance + Stress testing]: In this case, we have a uniform power applied to the entire chip. The total power dissipated is $204.8\ W$. In this case, the mean absolute error using our proposed approach is $1.5°$C for a maximum temperature rise of $77°$C (1.9%). The mean absolute error upon ignoring the shift-varying Green's function goes up to $3°$C (3.9%).

Test Case 5 [Variance + Stress testing]: In this case, the alternate grid points have power sources applied to them. The total power dissipated is $51.2\ W$. Our proposed method results in a mean absolute error of $0.6°$C for a maximum temperature rise of $29.2°$C (2%).

The steady state test cases are summarized in Table V. Note that across variation maps, the error remains roughly steady. The reported values are mean values. The maximum value is never more than the mean value by 20%.

Test Case	Total power (W)	Max. temp. ($°$C)	Mean abs. error ($°$C)
Test case 1	48.9	74.9	0.36°C (1.2%)
Test case 2	8.0	108.8	0.70°C (1.1%)
Test case 3	48.9	77.3	0.61°C (1.9%)
Test case 4	204.8	122.1	1.45°C (1.9%)
Test case 5	51.2	74.2	0.59°C (2%)

TABLE V: Steady-state test cases summary

D. Transient Results

1) Modified transient Green's functions: We obtain the modified transient Green's function using the leakage-aware steady-state Green's function as the starting point. Figure 7 shows the temporal evolution of the calculated transient Green's function using Equation 36 at the center of the chip. The estimation error is less than 3% at all times. We compute the modified Green's function at 100 time instants between 0 and 10 ms. Our algorithm takes $0.12\ s$ to compute the temperature profiles for the 100 time steps (Table VI).

Simulator	Time
Hotspot	$18 - 20\ minutes$
VarSim	$0.29\ s$ **(150 time steps)**
VarSim modified Green's func.	$120\ ms$ (100 time steps)
VarSim full chip step response	$70\ ms$ (100 time steps)
VarSim full chip time varying temp.	$290\ ms$ (150 time steps)

1. To model temperature-dependent conductivity, detailed thermal modeling is done in HotSpot, since the properties of each block are different.
2. HotSpot, 3D-ICE and Icepak do not consider variability

TABLE VI: Speed of existing simulators

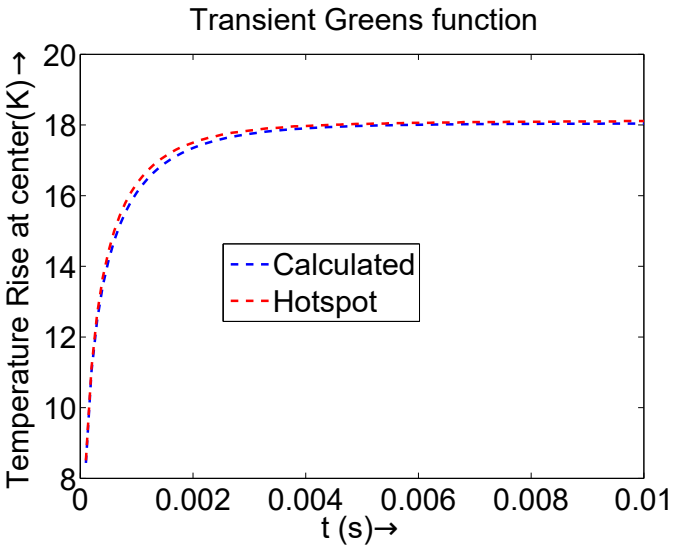
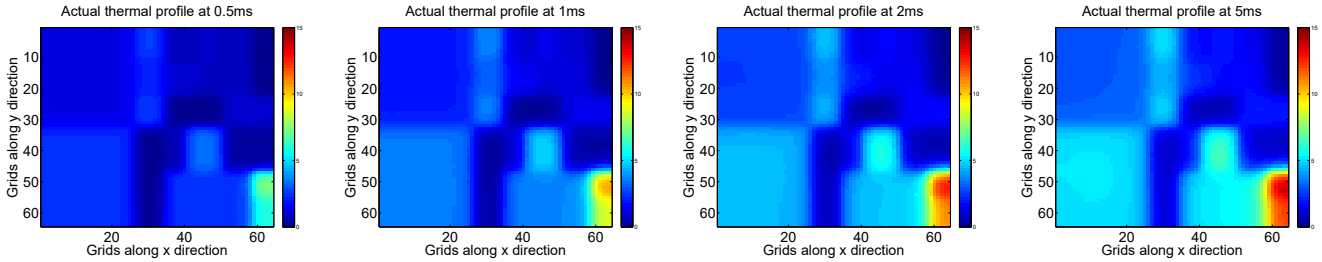


Fig. 7: Transient Green's function considering all effects

2) Full chip transient thermal profile: Next, we use the derived transient Green's function to obtain the transient thermal profile for the floorplan of Alpha21264, corresponding to the power profile in Figure 4a. The error at all times was observed to be less than 5% with a simulation time of $70\ ms$ for 100 time steps. The corresponding computed transient thermal profiles at $0.5\ ms$, $1\ ms$, $2\ ms$, and $5\ ms$ are shown in Figure 9. The corresponding actual thermal profiles obtained from a modified version of Hotspot are shown in Figure 8.

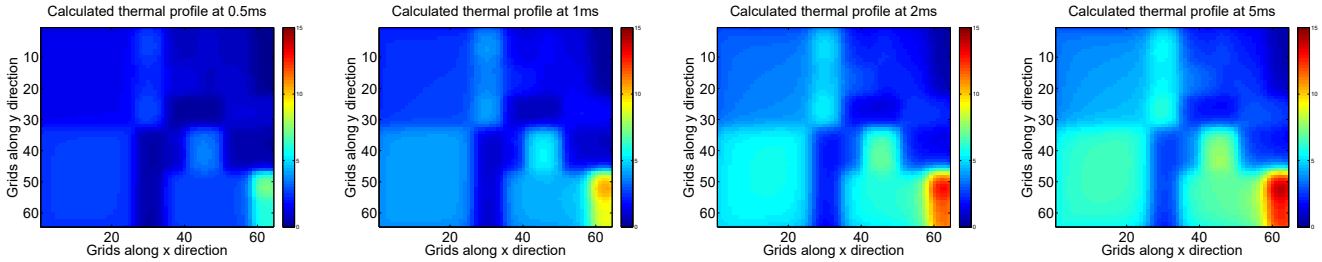
3) Full chip time-varying transient thermal profile: **Test Case 1 [Real floorplan]:** We randomly vary the power profile in Figure 4a every $1\ ms$ and obtain the thermal profile for this time-varying power input using HotSpot as well as our proposed method (Equation 37). We simulate the temperature until $15\ ms$ at intervals of $0.1\ ms$. Using our algorithm, we needed $290\ ms$ to compute the thermal profile for 150 time steps ($\approx 2\ ms$ per time step), while HotSpot took nearly 20 minutes ($4000\times$ speedup). The average error in our case was 3.8%. The evolution of the dynamic power profile at the hottest location is shown in Figure 10a, while the corresponding thermal profile is shown in Figure 10b.

Test Case 2 [Stress testing]: Next, we randomly vary the power profile in test case 2 of the steady-state every $1\ ms$ and observe the evolution of temperature over 30 time steps. The power profile at one of the locations and the corresponding



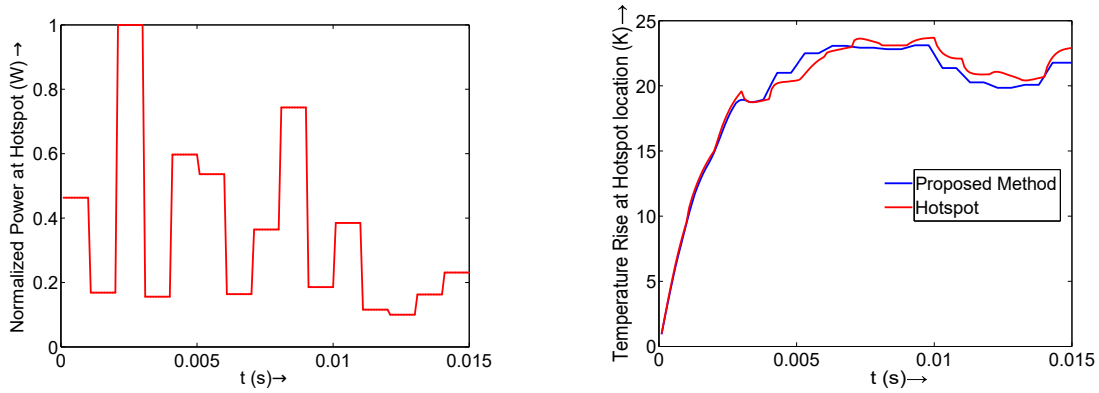
(a) Temperature rise at $0.5\ ms$ (b) Temperature rise at $1\ ms$ (c) Temperature rise at $2\ ms$ (d) Temperature rise at $5\ ms$

Fig. 8: Power and temperature map, transient, actual



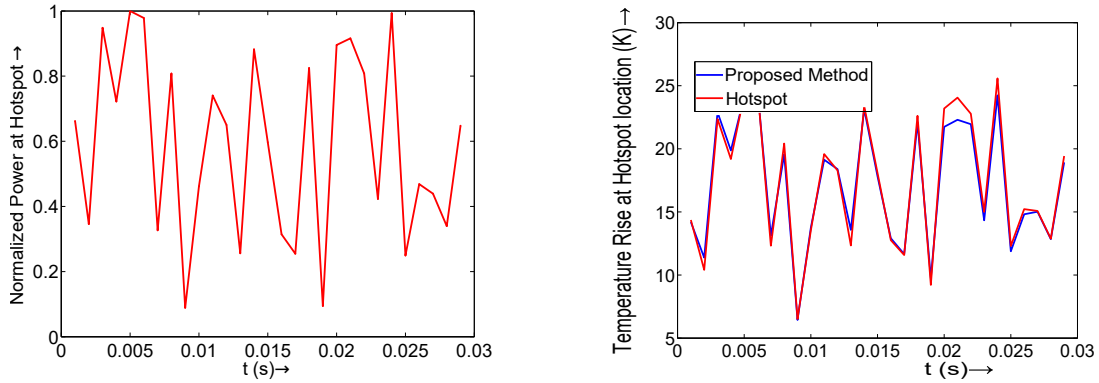
(a) Temperature rise at $0.5\ ms$ (b) Temperature rise at $1\ ms$ (c) Temperature rise at $2\ ms$ (d) Temperature rise at $5\ ms$

Fig. 9: Power and temperature map, transient, calculated



(a) Transient power variation (b) Transient evolution of temperature at the hottest location

Fig. 10: Transient evolution of temperature: test case 1



(a) Transient power variation (b) Transient evolution of temperature at one location

Fig. 11: Transient evolution of temperature: test case 2

thermal profiles are given in Figure 11. Our algorithm takes $59\ ms$ to compute the thermal profile for 30 time steps at intervals of $1\ ms$ each ($2\ ms$ per time step). In contrast, HotSpot takes over 15 minutes to compute the same thermal profile ($15000\times$ speedup). These results are summarized in Table VI.

E. Analysis of the Results

We have established through a wide range of test cases that our proposed method provides fast as well as accurate solutions for the effects considered. Next, we analyze the importance of modeling each of the individual effects – temperature-dependent leakage, variation in leakage, temperature-dependent conductivity, and variation in conductivity profiles.

To do so, we sequentially consider a subset of these effects while ignoring the rest of the effects. This helps us determine the individual thermal contribution of each of these effects in the cases we have modeled.

Table IV summarizes the results obtained in various scenarios. Figure 6 graphically represents this error. We see that not accounting for any kind of variability leads to a temperature estimation error of up to 22%. If we ignore the temperature dependence of conductivity but consider variability and temperature dependence of leakage, the error varies from 2 to 7.5%. Ignoring variations in the conductivity profile leads to $< 1\%$ error in thermal estimation. Thus in thermal modeling, the effects of random variations in the conductivity profile can be safely ignored. The need to model temperature-dependent conductivity would depend on the accuracy requirement of the design. However, considering variability in leakage power along with the temperature dependence of leakage power is *absolutely essential* to achieve a meaningful simulation accuracy.

F. Comparison with State-of-the-art Approaches

We compare our results against the modified version of HotSpot as well as with a CNN-based approach, similar to [35]. Our algorithm provides a $4000 - 370000\times$ speedup over HotSpot, while maintaining the error within 4% in all cases.

We implemented the same CNN architecture that is there in in Tensorflow for thermal modeling as was done in [35] for a similar setup as Test Case 1 in the present work. We used an Intel i7-7700 4-core desktop PC with 16 GB RAM running Ubuntu 16. To collect the training data, we used the open-source tool Varius [15] to create a large set of variation maps. We then obtained several dynamic power profiles by modifying the power values in the *ev6* test case of HotSpot. We then added those two to get the baseline power map and provided them as an input to HotSpot. We iteratively ran HotSpot multiple times to close the leakage-temperature feedback loop, and finally obtain the final leakage-converged thermal maps.

It takes 18-30 minutes to generate a single map using HotSpot. We found that 100 training maps is acceptable. In the next step, we provided this dataset to the 3-CNN-layer architecture proposed in [35]. However, again the training time was very large and did not converge even after 48 hours. So we used the simpler 1-CNN layer architecture for validation purposes, along with 5-fold cross validation. This is a known problem with CNNs that handle big data and is one of the prime reasons for not preferring ML-based approaches. Figure 12 shows some qualitative results.

We were able to reduce the training time to 8 hours with such reductions in the model capacity and input samples. The training mean absolute error was $0.8°C$. In the inference stage, we used 20 maps. The test accuracy was $0.93°C$ with an inferencing time of $0.65\ s$.
In comparison, the mean absolute error of our analytical method is $0.36°C$, with a total time requirement of $2.9\ ms$. In addition, our analytical method does not require any special hardware resources for training such as large GPUs.

This exercise underscores two things: 1) analytical methods almost always outperform machine learning based methods, since machine learning methods have long training times and require specialized hardware resources. ML algorithms should only be used when an analytical solution is not possible or the requirement for accuracy is not there.

2) The existing methods, such as HotSpot, CNN-based or FEM-based solvers/inference engines have severe limitations when it comes to modeling process variation. Our current work hopes to alleviate these limitations of the state-of-the-art methods.

Table III summarizes the simulation speed of various tools for the steady-state case.

Memory and Energy Analysis: Since we use compact thermal models in our work, our method consumes a lot less memory compared to the existing simulators. For the steady-state, our method uses 4MB memory only, compared to 360MB for HotSpot. Similarly, in the transient case, we use approximately 85MB, compared to the 900 MB used by HotSpot. The high memory consumption in HotSpot is primarily because finite difference methods involve matrices with a very large number of nodes, whereas Green's function methods need just as many nodes as the required temperature resolution.

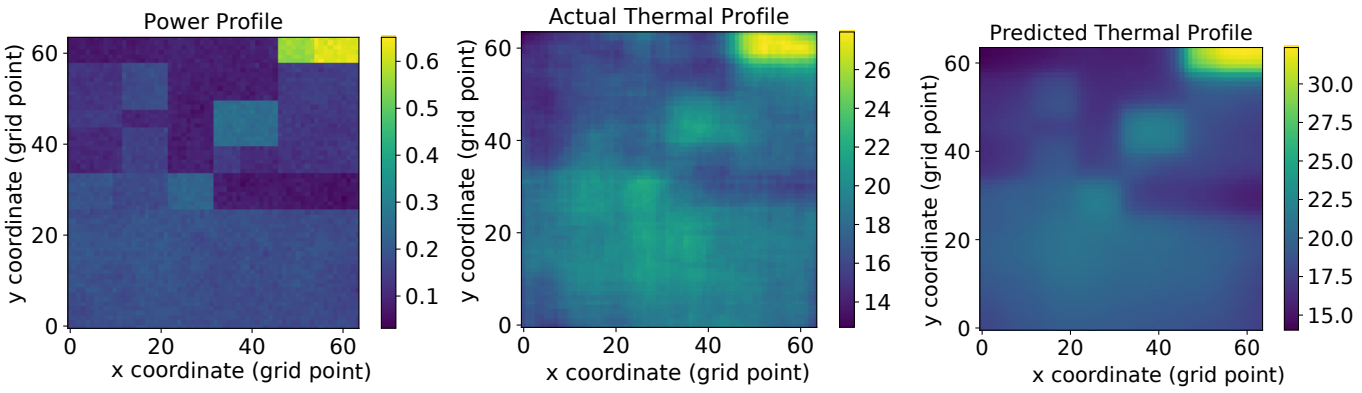
Owing to its analytical nature and ultra-high speed, we outperform state-of-the-art methods in terms of energy consumption as well.

VI. Conclusion

In this paper, we propose a fast leakage and variability-aware thermal simulation method that also captures the temperature dependence of conductivity. We derive a closed-form of the Green's function considering all these effects using novel insights and algebraic techniques. Our approach provides fast and accurate solutions for both the steady-state and the transient thermal profile and has been validated with a wide variety of test cases. As device dimensions continue to shrink, process variation has become a serious problem. The methods proposed in this work can equip designers to tackle this problem and may spawn further research in this area.

References

[1] R. Kessler, "The alpha 21264 microprocessor," *IEEE micro*, vol. 19, no. 2, pp. 24–36, 1999.



(a) Power map (b) Predicted temp. map (c) Actual temp.

Fig. 12: Power and thermal profiles using an ML-based method

[2] W. Guan, X. Tang, H. Lu, Y. Zhang, and Y. Zhang, "A novel thermal-aware floorplanning and tsv assignment with game theory for fixed-outline 3-D ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[3] Y. Chen, D. Zhao, F. Liu, J. Gao, and H. Zhu, "Thermal layout optimization for 3D stacked multichip modules," *Microelectronics Journal*, vol. 139, p. 105882, 2023.

[4] Y. Zhan, B. Goplen, and S. Sapatnekar, "Electrothermal analysis and optimization techniques for nanoscale integrated circuits," in *ASPDAC'06*, 2006, pp. 219–222.

[5] Y. Yang, Z. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated space-and-time-adaptive chip-package thermal analysis," *IEEE TCAD*, vol. 26, no. 1, pp. 86–99, 2006.

[6] H. Sultan and S. Sarangi, "A fast leakage aware thermal simulator for 3D chips," in *DATE'17*, 2017, pp. 1733–1738.

[7] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 9, pp. 1661–1675, 2007.

[8] H. Sultan and S. R. Sarangi, "Varsim: a fast and accurate variability and leakage aware thermal simulator," in *DAC 2020*. IEEE, 2020, pp. 1–6.

[9] D. V. Widder, *The heat equation*. Academic Press, 1976, vol. 67.

[10] J. V. Beck, K. D. Cole, A. Haji-Sheikh, and B. Litkouhl, *Heat conduction using Green's function*. Taylor & Francis, 1992.

[11] S. Sarangi, G. Ananthanarayanan, and M. Balakrishnan, "Lightsim: A leakage aware ultrafast temperature simulator," in *ASPDAC'14*, 2014, pp. 855–860.

[12] A. Ziabari, J.-H. 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 TVLSI*, vol. 22, no. 11, pp. 2366–2379, 2014.

[13] "Timing analysis algorithm using a neural network under pvt variations," *Microelectronics Journal*, vol. 125, p. 105480, 2022.

[14] S. Mittal, "A survey of architectural techniques for managing process variation," *CSUR*, vol. 48, no. 4, p. 54, 2016.

[15] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Varius: A model of process variation and resulting timing errors for microarchitects," *IEEE TSM*, vol. 21, no. 1, pp. 3–13, 2008.

[16] "Study of variability induced by random dopant fluctuation in fe ds-sbtfet," *Microelectronics Journal*, vol. 125, p. 105467, 2022.

[17] E. Humenay, D. Tarjan, and K. Skadron, "Impact of process variations on multicore performance symmetry," in *DATE 2007*. IEEE, 2007, pp. 1–6.

[18] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," in *DATE 2007*, 2007, pp. 1–6.

[19] H. Sultan, A. Chauhan, and S. R. Sarangi, "A survey of chip-level thermal simulators," *CSUR*, vol. 52, no. 2, pp. 1–35, 2019.

[20] G. Leung and C. O. Chui, "Variability impact of random dopant fluctuation on nanoscale junctionless FinFETs," *IEEE Electron Device Letters*, vol. 33, no. 6, pp. 767–769, 2012.

[21] M. G. Burzo, P. L. Komarov, and P. Raad, "Non-contact thermal conductivity measurements of p-doped and n-doped gold covered natural and isotopically-pure silicon and their oxides," in *EuroSimE'04*, 2004, pp. 269–276.

[22] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage VLSI design," *VLSI Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 501–513, 2006.

[23] L. Jiang, A. Dowling, Y. Liu, and M.-C. Cheng, "Chip-level thermal simulation for a multicore processor using a multi-block model enabled by proper orthogonal decomposition," in *iTherm, 2022.*, 2022.

[24] Y. Chen, S. Vinco, E. Macii, and M. Poncino, "Systemc-ams thermal modeling for the co-simulation of functional and extra-functional properties," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 24, no. 1, pp. 1–26, 2018.

[25] Z. Liu, Y. Li, J. Hu, X. Yu, S. Shiau, X. Ai, Z. Zeng, and Z. Zhang, "DeepOHeat: Operator learning-based ultra-fast thermal simulation in 3D-IC design," in *DAC*, 2023.

[26] S. Niknam, Y. Shen, A. Pathania, and A. D. Pimentel, "3D-TTP: Efficient transient temperature-aware power budgeting for 3d-stacked processor-memory systems," in *ISVLSI*, 2023.

[27] S. Krishnaswamy, P. Jain, M. Saeidi, A. Kulkarni, A. Adhiya, and J. Harvest, "Fast and accurate thermal analysis of smartphone with dynamic power management using reduced order modeling," in *ITherm, 2017.*, 2017.

[28] P. Shukla, V. F. Pavlidis, E. Salman, and A. K. Coskun, "TREAD-M3D: Temperature-aware dnn accelerators for monolithic 3D mobile systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[29] H. Sultan and S. R. Sarangi, "A fast leakage-aware green's-function-based thermal simulator for 3-D chips," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2342–2355, 2020.

[30] J. Jaffari and M. Anis, "Statistical thermal profile considering process variations: Analysis and applications," *IEEE TCAD*, vol. 27, no. 6, pp. 1027–1040, 2008.

[31] K. Meng, F. Huebbers, R. Joseph, and Y. Ismail, "Modeling and characterizing power variability in multicore architectures," in *ISPASS'07*, 2007, pp. 146–157.

[32] Y.-M. Lee and P.-Y. Huang, "An efficient method for analyzing on-chip thermal reliability considering process variations," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 18, no. 3, 2013.

[33] P.-Y. Huang and Y.-M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 17, no. 5, pp. 613–626, 2009.

[34] D.-C. Juan, S. Garg, and D. Marculescu, "Statistical thermal evaluation and mitigation techniques for 3D chip-multiprocessors in the presence of process variations," in *DATE'11*, 2011, pp. 1–6.

[35] H. Sultan and S. R. Sarangi, "Variability-aware thermal simulation using CNNs," in *2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID)*. IEEE, 2021, pp. 65–70.

[36] W. Jia and M.-C. Cheng, "A methodology for thermal simulation

of interconnects enabled by model reduction with material property variation," *Journal of Computational Science*, vol. 61, p. 101665, 2022.
[37] M. Shafique, D. Gnad, S. Garg, and J. Henkel, "Variability-aware dark silicon management in on-chip many-core systems," in *DATE 2015*, 2015, pp. 387–392.
[38] G. Prasad Srinivasa, S. Haseley, G. Challen, and M. Hempstead, "Quantifying process variations and its impacts on smartphones," in *ISPASS, 2019.*, 2019, pp. 117–126.
[39] B. Li, L.-S. Peh, and P. Patra, "Impact of process and temperature variations on network-on-chip design exploration," in *NOCS*, 2008, pp. 117–126.
[40] W. Huang, K. Skadron, S. Gurumurthi, R. J. Ribando, and M. R. Stan, "Differentiating the roles of IR measurement and simulation for power and temperature-aware design," in *ISPASS, 2009.*, 2009, pp. 1–10.
[41] A. Ziabari, Z. Bian, and A. Shakouri, "Adaptive power blurring techniques to calculate IC temperature profile under large temperature variations," *IMAPS'10*, 2010.
[42] A. Kumar, N. Chang, D. Geb, H. He, S. Pan, J. Wen, S. Asgari, M. Abarham, and C. Ortiz, "ML-based fast on-chip transient thermal simulation for heterogeneous 2.5D/3D ic designs," in *VLSI-DAT, 2022.*, 2022, pp. 1–8.
[43] L. Siddhu, R. Kedia, S. Pandey, M. Rapp, A. Pathania, J. Henkel, and P. R. Panda, "CoMeT: An integrated interval thermal simulation toolchain for 2D, 2.5D, and 3D processor-memory systems," *ACM TACO.*, vol. 19, no. 3, aug 2022.
[44] "Frequency-scaled thermal-aware test scheduling for 3d ics using machine learning based temperature estimation," *Microelectronics Journal*, vol. 128, p. 105535, 2022.
[45] Z. He, W. Cui, C. Cui, T. Sherwood, and Z. Zhang, "Efficient uncertainty modeling for system design via mixed integer programming," in *ICCAD, 2019.*, 2019, pp. 1–8.
[46] S. V. R. chittamuru, I. G. Thakkar, and S. Pasricha, "LIBRA: Thermal and process variation aware reliability management in photonic networks-on-chip," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 758–772, 2018.
[47] Z. Yu, D. Yergeau, and R. W. Dutton, "Full chip thermal simulation," in *ISQED'00*, 2000, pp. 145–149.
[48] R. Zhang, M. R. Stan, and K. Skadron, "Hotspot 6.0: Validation, acceleration and extension," University of Virginia, Tech. Rep., 2015.

Hameedah Sultan Hameedah Sultan has obtained her Ph.D. degree from the School of Information Technology, Indian Institute of Technology Delhi in 2021. She now works as a part of the Graphics Systems team in Qualcomm, Singapore. She has done her Masters in VLSI Design Tools and Technology, IIT Delhi. Her research interests include architectural-level power, thermal and noise modeling and its impact on performance.

Smruti R. Sarangi Prof. Smruti Ranjan Sarangi is a Professor in the Computer Science and Engineering Department at IIT Delhi with a joint appointment in the Department of Electrical Engineering. He primarily works on computer architecture, EDA and operating systems. His research areas specifically cover EDA, multicore processors, cyber-security, emerging technologies, networks on chip, operating systems for parallel computers and parallel algorithms. Dr. Sarangi obtained his Ph.D in computer architecture from the University of Illinois at Urbana Champaign(UIUC), USA in 2006, and a B.Tech in computer science from IIT Kharagpur in 2002. He has filed five US patents, seven Indian patents and has published 125 papers in reputed international conferences and journals. He is the author of two popular textbooks in computer architecture for UG and PG students, respectively. He is a member of the IEEE and ACM.

APPENDIX A
HANKEL TRANSFORM AND ITS PROPERTIES

The Hankel transform is a mathematical transform that decomposes any function $f(r)$ into an infinite number of Bessel functions of the first kind. It is defined as:

$$\mathcal{H}(f(r)) = H(s) = \int_0^\infty f(r)\mathcal{J}_0(sr)rdr, \tag{39}$$

where $\mathcal{J}_0$ is the Bessel function of the first kind of order 0, and $\mathcal{H}$ denotes the Hankel transform operator.

The inverse Hankel transform of $H(s)$ is defined as:

$$\mathcal{H}^{-1}(H(s)) = f(r) = \int_0^\infty H(s)\mathcal{J}_0(sr)sds, \tag{40}$$

where $\mathcal{H}^{-1}$ denotes the inverse Hankel transform operator.

Theorem I: The Hankel transform of a radially symmetric function in polar coordinates is equivalent to its 2D Fourier transform.

Proof:

The 2D Fourier transform is given by:

$$\mathcal{F}(f(r)) = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j(ux+vy)}dxdy, \tag{41}$$

Let $x = rcos\theta$ and $y = rsin\theta$. Let $u = \rho cos\phi$ and $v = \rho sin\phi$

Substituting these in the above equation:

$$F(\rho,\phi) = \int_0^\infty \int_{-\pi}^{\pi} f(r,\theta)e^{-ir\rho cos(\phi-\theta)}rdrd\theta \tag{42}$$

If $f(r)$ is radially symmetric, it is independent of the angle. Thus we can rewrite the above equation as:

$$F(\rho,\phi) = \int_0^\infty rf(r)dr \int_{-\pi}^{\pi} e^{-ir\rho cos(\phi-\theta)}d\theta \tag{43}$$

Using the definition of the zeroth-order Bessel function:

$$J_0(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-ixcos(\phi-\theta)} \tag{44}$$

Using this in the equation above, we get:

$$F(\rho) = \mathcal{F}(f(r)) = 2\pi \int_0^\infty f(r)J_0(\rho r)rdr \tag{45}$$

which is the same is 2π times the Hankel transform of order 0. Thus:

$$F(\rho) = \mathcal{F}(f(r)) = 2\pi\mathcal{H}(f(r)) \tag{46}$$

Lemma I: : $G(u,v) = \mathcal{F}(f_{sp_0}\mathcal{T}) = \mathcal{F}(\mathcal{T})g_{sp_0}$, where $g_{sp_0} = (f_{sp_0} - \kappa + f_{sp_0}(0,0))$

Proof:

We use the zero order Hankel transform on $G(u,v)$ to reduce the 2D Fourier transform to a 1D Hankel transform. We denote the variables in polar coordinates by the ˜ operator. Next, we apply integration by parts:

$$\begin{aligned} H(h) = \mathcal{H}(\tilde{f}_{sp_0}\tilde{\mathcal{T}}) &= \int_0^\infty (\tilde{f}_{sp_0}\tilde{\mathcal{T}})J_0(hr)rdr \\ &= \tilde{f}_{sp_0}\int_0^\infty \tilde{\mathcal{T}}J_0(hr)rdr - \int_0^\infty \tilde{f}'_{sp_0}dr\int_0^\infty \tilde{\mathcal{T}}J_0(hr)rdr \\ &= \tilde{f}_{sp_0}\mathcal{H}(\tilde{\mathcal{T}}) - \int_0^\infty \tilde{f}'_{sp_0}dr \times \mathcal{H}(\tilde{\mathcal{T}}) \\ &= \mathcal{H}(\tilde{\mathcal{T}})\left(\tilde{f}_{sp_0} - \int_0^\infty \frac{\partial \tilde{f}_{sp_0}}{\partial r}dr\right) \\ &= \mathcal{H}(\tilde{\mathcal{T}})\left(\tilde{f}_{sp_0} - \tilde{f}_{sp_0}(\infty) + \tilde{f}_{sp_0}(0)\right) \end{aligned} \tag{47}$$

where $'$ denotes the derivative with respect to r.

Let $\tilde{f}_{sp_0}(\infty) = \kappa$. The equivalent expression in the Cartesian coordinates becomes:

$$\begin{aligned} G(u,v) = \mathcal{F}(f_{sp_0}\mathcal{T}) &= \mathcal{F}(\mathcal{T})\underbrace{(f_{sp_0} - \kappa + f_{sp_0}(0,0))}_{=g_{sp_0}} \\ &= \mathcal{F}(\mathcal{T})g_{sp_0} \end{aligned} \tag{48}$$

where $\mathcal{F}$ is the Fourier transform operator.

Appendix B
Mathematics of Green's Functions

Let $\mathcal{L}$ be a linear partial differential operator. Let us consider the inhomogeneous equation:

$$\mathcal{L}(f(x)) = q(x) \tag{49}$$

Let us define a function $G(x)$, which is the solution to the equation when $q(x) = \delta(x)$. Here, $\delta(x)$ is the Dirac delta function. It is defined as having a value of 0, $\forall\, x \neq 0$. Furthermore,

$$\int_{-\infty}^{\infty} \delta(x).dx = 1$$

We thus have:

$$\mathcal{L}(G(x)) = \delta(x) \tag{50}$$

Let $G(x)$ be referred to as the Green's function or the *fundamental solution*. Let us show how we can construct solutions for Equation 49 using the fundamental solution, $G(x)$. Using the properties of the convolution operation $\star$, we can argue as follows:

$$\begin{aligned} &\mathcal{L}(G(x)\star q(x)) = \mathcal{L}(G(x))\star q(x) \\ &= \delta(x)\star q(x) = \int_{-\infty}^{\infty}\delta(x-x')q(x')dx' \\ &= q(x) \end{aligned} \tag{51}$$

Hence, $G(x)\star q(x)$ is a solution to the equation $\mathcal{L}(f(x)) = q(x)$ (Equation 49). For the multi-variable case, the explanation is exactly similar.

Now, the Fourier's heat equation (steady state) is of the form:

$$\nabla^2(y) = q(x)$$

Here, ∇^2 is the Laplacian operator ($\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$). y represents the temperature and $q(x)$ stands for the dynamic power (or is proportional to it). We thus observe that the temperature can be computed by calculating a convolution between the Green's function and the dynamic power profile. This approach can be extended to the transient case as well where there is an additional temperature derivative term. The basic idea is to compute a Green's function that is a function of time.

Green's functions can be readily computed for the 2D Laplacian operator ∇^2 (given our 2D Laplacian geometry). The fundamental solution or Green's function is of the form $A.ln(r)+B$, assuming circular symmetry in polar coordinates (as is our case without variation). ln stands for the natural logarithm. Here A is a constant (- 1/2π). We can modify A and B subsequently to satisfy all the boundary conditions. The result equation will be of the form $C - Dln(r)$, where C and D are positive constants. If we plot this equation, we will have a rapidly decaying component for small r (shape of -ln(r)), which will asymptotically appear to converge to a constant line. This graph is very similar to that produced by the expression that we have used for the Green's function, $f_{sp} = fsilic + \kappa$. This is the Green's function for a simple geometry without leakage and variation. Here, f_{silic} is the rapidly decaying part. κ is a constant, which represents the roughly constant asymptote. This is the intuition behind the form of the Green's function that we have chosen (also observed experimentally). Figure 13 shows an example of a Green's function solution for a 2D Laplacian equation that has the same form. We can clearly see the rapidly decaying part and the constant part.

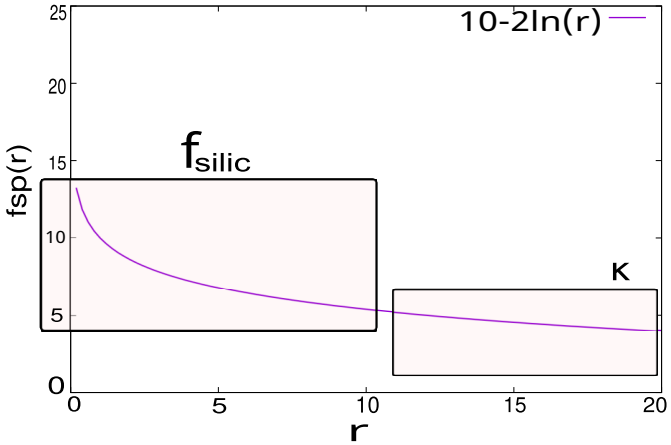


Fig. 13: Examples of a Green's function solution for a 2D Laplacian equation

The Green's functions in practice are often not computed directly. The resulting expressions are too complex once factors like leakage and process variation are taken into account.

We instead require the Fourier or Hankel transform (Fourier transform in polar coordinates) of the Green's function in practice. This can be directly multiplied with the Fourier transform of the power profile to obtain the Fourier transform of the temperature profile. Multiplication is clearly a much faster operation than convolution.

The final temperature field is obtained after computing an inverse Fourier transform of the Fourier trasform of the temperature (computed using a convolution operation). This is computed only once, at the very end.

The final expressions for the Fourier transform of the Green's functions for the steady and transient versions of the problem are shown in Equations 26 and 36, respectively (in the manuscript). This is the closest that we get in terms of closed-form analytical solutions. Note that these equations do not produce an analytical solution for the final Green's function. They show the relationship between the baseline Green's function (under ideal conditions like no leakage) and the real Green's function (assumes all kinds of effects inclusive of leakage and process variation) using simply analytic expressions. Hence, in practice, the expressions for the final Green's function are computed numerically given the baseline Green's function. However, there is almost never any need to explicitly compute an inverse Fourier transform of the Green's functions themselves.