Statistical Thermal Modeling and Optimization Considering Leakage Power Variations

Da-Cheng Juan[1], Yi-Lin Chuang[2], Diana Marculescu[1], and Yao-Wen Chang[3]
[1]Dept. of Electrical and Computer Engineering, Carnegie Mellon University, PA, U.S.A.
[2]Taiwan Semiconductor Manufacturing Company, HsinChu, Taiwan
[3]Dept. of Electrical Engineering & Grad. Inst. of Electronics Engineering, National Taiwan University, Taipei, Taiwan
[1]{dacheng, dianam}@cmu.edu, [2]nicky@eda.ee.ntu.edu.tw, [3]ywchang@cc.ee.ntu.edu.tw

Abstract – Unaddressed thermal issues can seriously hinder the development of reliable and low power systems. In this paper, we propose a statistical approach for analyzing thermal behavior under leakage power variations stemming from the manufacturing process. Based on the proposed models, we develop floorplanning techniques targeting thermal optimization. The experimental results show that peak temperature is reduced by up to 8.8˚C, while thermal-induced leakage power and maximum thermal variance are reduced by 13% and 17%, respectively, with no additional area overhead compared with best performance-driven optimized design.

I. INTRODUCTION

Excessively high operating temperature is the root of many reliability issues. The rate of several failure mechanisms, such as aging and electro-migration, increases exponentially when the operating temperature increases [1]. High operating temperature also impacts system's leakage power; the increase of leakage power contributes to the increase of total power consumption, which in turn increases the operating temperature. This thermal-leakage positive feedback loop may lead to the thermal runaway phenomenon, which may burn the chip in the worst case.

The presence of manufacturing process variations [2][3] further deteriorates thermal behavior by increasing leakage power [4]. Due to leakage variations, the power consumption of a module may vary from its nominal value, and therefore introduces unexpected thermal hotspots which may not be captured during design time [5]. This phenomenon requires a statistical, instead of a deterministic model, for evaluating and mitigating thermal hotspots early in the design cycle.

Modifying the floorplan of a system is one of the most effective and direct methods to mitigate thermal hotspots, since the operating temperature has strong spatial correlations [6][7]. However, an ideal thermal mitigation strategy should not affect the performance of a system. For example, in addition to mitigating thermal hotspots, an ideal thermal-aware floorplan should also optimize the physical length and routability of on-chip interconnects among modules [8]. The on-chip interconnects here refer to the connection among modules, IPs, or processing cores, not the wires among transistors or logic gates. Therefore, optimal planning of on-chip interconnects along with thermal mitigation under leakage variations requires a multi-constrained optimization strategy.

A. PRIOR ART

Thermal-aware designs have received a lot of attention in the research community; many thermal mitigation techniques have been proposed at each level of design hierarchy – from physical level up to system level. Huang et al. [7] proposed Hotspot, an accurate temperature model for planar ICs. Based on Hotspot, Sankaranarayanan et al. [6] proposed a thermal-aware strategy for floorplanning. Goplen et al. [9] and Cong et al. [10] proposed novel placement and floorplanning algorithms to reduce the temperatures in 3D systems. Ebi et al. [11] presented an agent-based power distribution approach to balance the power consumption in a pro-active manner. Chuang et al [12] proposed an effective placement strategy for mixed-size circuits considering global power spreading. Very recently, Zhuo et al. [13] presented a workload-aware framework that accounts for local variations in both the process and temperature.

The planning of on-chip interconnects has also been addressed by Cheng et al. [8] and Xiang et al. [14]. However, the goal of both papers was to facilitate the routability of on-chip interconnects, not mitigating the thermal emergencies for a design.

B. PAPER CONTRIBUTIONS

- Previous work has only addressed either the thermal mitigation or the interconnect routability of a system, but not both simultaneously, especially under leakage variation conditions. In this work, we perform a joint optimization on both criteria under process variations for the first time. The proposed framework helps designers create a thermal-aware floorplan which is highly robust to leakage variations while maintaining the routability and efficiency of on-chip interconnects.
- We develop a statistical thermal model based on the linearity of Gaussian distributions and linear time-invariant (LTI) system. A popular statistical method, Box-Cox transformation [15][16], is applied as an ancillary to provide normality when the dataset, i.e., the distribution of leakage power does not exactly follow a Gaussian distribution. The proposed model is able to estimate the steady-state temperature of a system under leakage power variations. Furthermore, this model can also be used in any thermal mitigation strategy for minimizing the peak temperature under leakage variations.
- To perform optimization under multiple constraints, we propose an adaptive simulated annealing (SA) algorithm incorporated with the proposed statistical thermal model, and implement it with a state-of-the-art B*-tree floorplanner [18].
- To evaluate the effectiveness and generality of the proposed thermal optimization framework, we perform experiments on multiple benchmark suites that cover a wide spectrum of designs and systems: (1) the *Alpha21264* processor [19], (2) Embedded System Synthesis Benchmarks Suite (*E3S*) [20], and (3) *MCNC* benchmarks modified by [14]. Experimental results confirm that the proposed framework is very effective for each type of benchmarks – up to 8.8˚C, 13% thermal-induced leakage power, and 17% thermal variance are reduced with no additional area cost or interconnect length overhead.

C. PAPER ORGANIZATION

The remainder of this paper is organized as follows. Section II introduces the related background knowledge required for our work. Section III provides the problem formulation. Section IV details the proposed statistical thermal optimization framework by using simulated annealing. Section V presents the implementation flow.

978-3-9810801-8-6/DATE12/©2012 EDAA

Section VI demonstrates the experimental results. Section VII concludes this paper.

II. Preliminaries

In this section, we introduce the background knowledge relevant to our proposed methodology, including thermal model, leakage power model, manufacturing process variations, and B*-tree floorplanning representation.

A. Thermal model

This section presents the thermal model used throughout the paper. To this end, heat flow is approximated as a heat current flowing through thermal resistance, resulting in temperature differences. This phenomenon can be modeled as the electrical current in an RC network, and the temperature differences can be expressed as:

$$\boldsymbol{C}\frac{d\boldsymbol{T}(t)}{dt} = \boldsymbol{\Psi}^{-1}\boldsymbol{T}(t) - \boldsymbol{P}U(t) \qquad \text{Eq(1)}$$

where $\boldsymbol{C}$ is a diagonal thermal capacitance matrix, $\boldsymbol{\Psi}$ is a thermal resistance matrix, $\boldsymbol{T}(t) = (T_1 - T_A, ..., T_n - T_A)^T$ is the temperature vector and T_A is the ambient temperature, $\boldsymbol{P} = (P_1, ..., P_n)^T$ is the power vector[1], $U(t)$ is a step function, and n is the number of thermal grids used to represent thermal gradient. For simplicity and without losing much accuracy, n is set to 32×32 = 1,024 here [7].

For the purpose of steady-state thermal analysis the temperature is assumed constant, i.e., $d\boldsymbol{T}(t)/dt = 0$. Therefore, Eq(1) can be modified as follows:

$$\boldsymbol{P} = \boldsymbol{\Psi}^{-1}\boldsymbol{T} \implies \boldsymbol{T} = \boldsymbol{\Psi}\boldsymbol{P}, \boldsymbol{\Psi} = \begin{bmatrix} \psi_{11} & \cdots & \psi_{1n} \\ \vdots & \ddots & \vdots \\ \psi_{n1} & \cdots & \psi_{nn} \end{bmatrix} \qquad \text{Eq(2)}$$

From Eq(2), it is clear that the power vector $\boldsymbol{P}$ and the thermal resistance matrix $\boldsymbol{\Psi}$ can directly determine the steady-state temperatures of a system. In this paper, we focus on the optimization of the steady-state temperature, since the transient-state temperature can be controlled by many dynamic thermal management (DTM) methodologies, such as task reallocation [11] and dynamic frequency and voltage scaling (DVFS) [21]. Also, we assume the cooling system and packaging are fixed, which implies a fixed $\boldsymbol{\Psi}$.

B. Leakage current

The power consumption of a circuit consists of active power and leakage power. This section focuses on modeling the feedback loop between leakage and temperature, since the active power is not sensitive to either temperature or process variations [22]. Leakage power contains several components, among which sub-threshold leakage current and gate leakage current are main contributors [23]. Recently, due to the introduction of high-k dielectrics, the gate leakage component has become less important. We therefore concentrate only on sub-threshold leakage power dissipation. Eq(3) describes the leakage current model for a single transistor. For simplicity and without losing accuracy, terms not sensitive to temperature or effective channel length are merged together:

$$I_{leak} = K\frac{W}{L_{eff}}(T)^2 e^{-\frac{c\cdot V_{th}}{T}} \qquad \text{Eq(3)}$$

where K is a technology dependent constant, W is the transistor width, L_{eff} is the effective channel length, T is the temperature, V_{th} is the threshold voltage, and c is a positive constant. According to Eq(3), I_{leak} will increase when L_{eff} decreases and T increases. Combining Eq(1), Eq(2) and Eq(3) determines the interdependency of temperature and leakage power. The increase of either temperature or leakage power will trigger this positive feedback loop.

It is worth mentioning that V_{th} also exponentially depends upon L_{eff} due to drain induced barrier lowering (DIBL) [24]. Therefore, the decrease of L_{eff} determines both exponential and linear scaling factors for leakage current.

C. Process variations

In general, the variation of the effective channel length (L_{eff}) of a transistor can be described as:

$$L_{eff} = L_{nom} \pm \Delta_{total} \qquad \text{Eq(4)}$$

where L_{nom} is the nominal value of L_{eff}, and Δ_{total} is the total variation of L_{eff}. Δ_{total} can be further decomposed as:

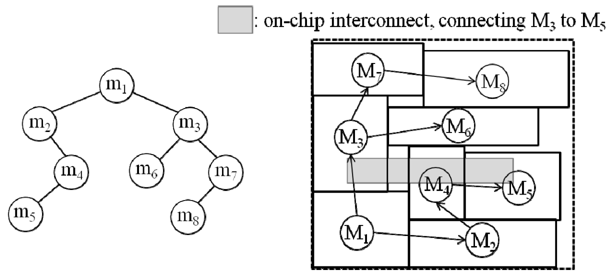
$$\Delta_{total} = w_1\Delta_{w2w} + w_2\Delta_{spat} + w_3\Delta_{rand} \qquad \text{Eq(5)}$$

where Δ_{w2w} is the wafer-to-wafer (W2W) variation, Δ_{spat} is the die-to-die (D2D) spatial variation, Δ_{rand} is the within-die (WID) random variation, and w_is are the corresponding weights. Δ_{w2w} and Δ_{rand} can be modeled as Gaussian random variables. In [2], Cheng et al. proposed an accurate, deterministic model of Δ_{spat} by exploiting the across-wafer variation from the manufacturing data in an industrial 45nm technology. In this paper, we use this model assuming Δ_{total} of 5% of L_{nom}, and L_{nom} is 45nm. Furthermore, based on [25], the relative ratio among Δ_{w2w}, Δ_{spat} and Δ_{rand} is set to 0.7:1:1.

D. B*-tree floorplan representation

A B*-tree [18] is an ordered binary tree used to represent non-slicing or slicing floorplans. Given a floorplan, a unique B*-tree can be constructed in linear time by a recursive procedure similar to the depth-first search. Furthermore, one can also pack each node in B*-tree to recover the original floorplan in linear time with a contour structure. Figure 1(a) shows an example of B*-tree. The root represents the block on the bottom-left corner, (0, 0). A block here can represent a module, IP, core, or any function block in a system. The node m_2 is the left child of m_1, which means block M_2 is the lowest block placed on the right-hand side and adjacent to M_1. On the other hand, m_3 is the right child of m_1, which means block M_3 is the first block placed above the block M_1 with the same x-coordinate. Therefore, given a B*-tree, the x-coordinate of all blocks can be determined by traversing this tree once in linear time. In addition, the y-coordinates can be computed by a contour structure in linear time [18].

On-chip interconnects are not represented as nodes in a B*-tree. Instead, they are modeled as constraints to restrain the planning of each module during the optimization stage. The orientation of interconnects among modules can be vertical or horizontal [14]. For the ease of routing, all vertical on-chip interconnects will be assigned to one metal layer and horizontal ones will be assigned to another one. Therefore, one legal deployment of on-chip



(a) B*-tree representation (b) Actual floorplan

Figure 1: B*tree and its corresponding floorplan.

[1] All the bold-font symbols represent vectors instead of scalars.

Inputs: $M_i, P_{Mi}^{dyn}, P_{Mi}^{leak}, W_i, H_i, B_j, \omega_j, \forall i, j$
Decision variable: Floorplan of $M_i, \forall i$
Initial Condition: $\boldsymbol{\Psi}$
Objective function: minimize $(T^{max}, L_{itr}, A_{chip})$**,** and
$(T^{max}, L_{itr}, A_{chip}) = f(M_i, P_{Mi}^{dyn}, P_{Mi}^{leak}, W_i, H_i, B_j, \omega_j, \Psi)$
Subject to:
The output floorplan of modules must be legal.

Figure 2: Problem formulation for multi-constrained thermal optimization.

interconnects needs to satisfy two conditions: (1) there is no overlap between any two horizontal (or vertical) interconnects, and (2) an on-chip interconnect must traverse through all modules that connect to it. Figure 1(b) shows a legal horizontal on-chip interconnect deployment.

III. Problem Formulation

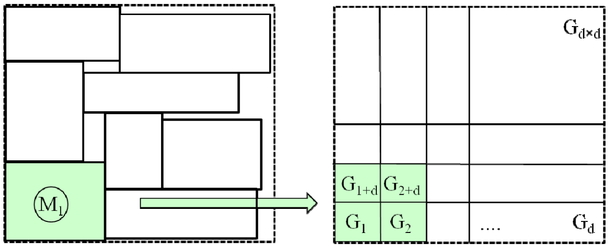
Figure 2 illustrates the formulation of the multi-constrained thermal optimization. First, several following inputs are given: (1) The dynamic and leakage power consumption of module *i*, M_i, denoted as P_{Mi}^{dyn} and P_{Mi}^{leak}. (2) The width W_i and the height H_i of M_i. (3) The information of on-chip interconnect j, B_j, including a set of modules connected by B_j, $B_j = < M_a, M_b, \ldots >$, and the width ω_j.

The decision variable here is the floorplan of all modules. The initial condition $\boldsymbol{\Psi}$ is given and fixed, which means the cooling system is given and fixed. The objective function is to minimize (1) the peak steady-state temperature T^{max} of a system, (2) the total on-chip interconnect length L_{itr}, and (3) the chip area A_{chip}. Finally, the final floorplan of each module needs to be legal, which means no overlap between any two modules and all on-chip interconnects are deployed correctly as described in Section II.D.

IV. Methodology

In this section, we explain the proposed multi-constrained thermal optimization framework in detail.

A. *Power Mapping: From Floorplan to Grids*



(a) Actual floorplan **(b) Thermal grids**

Figure 3: Mapping floorplan into thermal grids.

Before elaborating on the proposed methodology, we need to describe how to map the power of each block from the floorplan onto the thermal grids. Figure 3 shows an example of mapping a floorplan onto the thermal grids. For the simplicity of explanation, we only consider one module M_1 here. M_1 is mapped into G_1, G_2, G_{1+d} and G_{2+d} of the thermal grids based on the corresponding location. *d* is the granularity of the thermal grids, and as mentioned in Section II.A, *d* is set to 32 in this paper, so the total number of grids n = 32×32 = 1,024 [7]. Based on this mapping, the power consumption of a whole chip can be expressed as a vector $\boldsymbol{P} = (P_1, ..., P_n)^T$. In the rest of the paper, for the simplicity of explanations, $\boldsymbol{P}$ and P_i always refer to the power consumption of each grid, not each module unless we mention it explicitly. To capture the worst-case thermal scenario, we assume the power consumption of a chip is equal to its peak power and does not vary with time. In Section V, we will elaborate how to extract the peak power.

The power consumption of a system can be decomposed into dynamic and leakage power, i.e., $\boldsymbol{P} = \boldsymbol{P}^{dyn} + \boldsymbol{P}^{leak}$. From Eq(4), we know L_{eff} is a random variable depending on Δ_{total}, and since L_{eff} affects I_{leak}, I_{leak} is also a random variable. Without losing generality, we assume transistors in the same grid have the same I_{leak} distribution, since one can tune the granularity according to different accuracy requirements. Also, we assume $\boldsymbol{I_{leak}} = [I_1, \cdots, I_n]$ follows a multivariate Gaussian distribution $N(\boldsymbol{\mu_I}, \boldsymbol{\Sigma_I})$ with a mean vector $\boldsymbol{\mu_I}$ and a covariance matrix $\boldsymbol{\Sigma_I}$. The bold-font symbols here are vectors or matrices, not scalars. The dimensionalities of $\boldsymbol{\mu_I}$, $\boldsymbol{\Sigma_I}$ are (*n* × 1) and (*n* × *n*), respectively. This assumption will be relaxed later in Section IV.C. With the supply voltage V_{dd} set to 1V, we have:

$$\boldsymbol{P} = \boldsymbol{P^{dyn}} + \boldsymbol{P^{leak}} \sim N(\boldsymbol{P^{dyn}} + \boldsymbol{\mathfrak{L}\mu_I}, \boldsymbol{\mathfrak{L}\Sigma_I\mathfrak{L}^T}) \quad \text{Eq(6)}$$

where $\boldsymbol{\mathfrak{L}}$ is a (*n* × *n*) diagonal matrix, and each entity $\mathfrak{L}_{ii}$ represents the transistor count in the *i*th grid [26] used for estimating leakage power. Note that we did not explicitly express V_{dd} in Eq(6) because its value equals one. Also, since $\boldsymbol{P^{dyn}}$ is not sensitive to process variations [22], it will not contribute to the non-deterministic part and therefore can be merged into the mean vector of the multivariate Gaussian distribution.

From the power consumption point of view, a floorplan actually represents a certain permutation of $\boldsymbol{P} = (P_1, P_2, \ldots, P_n)^T$, which can be expressed by $\boldsymbol{\varpi P}$ where $\boldsymbol{\varpi}$ is a permutation matrix [27]. A permutation matrix is a square binary matrix that corresponds to a given floorplan. By using the linearity of a Gaussian distribution [16], $\boldsymbol{P}$ from Eq(6) can further be expressed as:

$$\boldsymbol{P_{\varpi}} \sim N\Big((\boldsymbol{\varpi P^{dyn}}) + (\boldsymbol{\varpi\mathfrak{L}})\boldsymbol{\mu_I}, (\boldsymbol{\varpi\mathfrak{L}})\boldsymbol{\Sigma_I}(\boldsymbol{\varpi\mathfrak{L}})^T\Big) \quad \text{Eq(7)}$$

Parentheses are intentionally placed around $\boldsymbol{\varpi}$, $\boldsymbol{P^{dyn}}$, $\boldsymbol{\mathfrak{L}}$ to indicate that modifying a floorplan via a permutation $\boldsymbol{\varpi}$ will change $\boldsymbol{P^{dyn}}$ and $\boldsymbol{\mathfrak{L}}$, because of remapping modules to different grids. For example, for module M_i, its dynamic and leakage power, P_{Mi}^{dyn} and P_{Mi}^{leak}, will reflect the floorplan described by the permutation. However, the underlying leakage variations $\boldsymbol{\mu_I}$, $\boldsymbol{\Sigma_I}$ will not change with $\boldsymbol{\varpi}$, because $\boldsymbol{\varpi}$ does not affect to manufacturing processes. To simplify the notation, we rewrite $\boldsymbol{P_{\varpi}}$ from Eq(7) as $\boldsymbol{P} \sim N(\boldsymbol{\mu_P}, \boldsymbol{\Sigma_P})$. Therefore, the original thermal mitigation problem can be converted into finding the best permutation matrix $\boldsymbol{\varpi}$ under multiple constraints described in Section III.

B. *Statistical Thermal Model*

We are now ready to investigate $\boldsymbol{\Psi}$ from Eq(2) more carefully. Mathematically, each P_i in $\boldsymbol{P}$ will impact thermal behavior (and therefore the whole $\boldsymbol{T}$) due to the multiplication by $\boldsymbol{\Psi}$. The intuition behind this is that thermal diffusion process is continuous and thus, each thermal grid will contribute to all other grids. This effect is also known as *spatial thermal correlation.* Therefore, if the power value of a certain grid varies from its nominal value due to leakage variations, this variability will "propagate" to each grid due to spatial correlations, which in turn will contribute to the thermal behavior of the whole system. As a result, the original thermal behavior will be altered, making thermal hotspots difficult to be captured [5].

As Eq(2) shows, $\boldsymbol{T}$ is a linear combination of $\boldsymbol{P}$ due to the multiplication of $\boldsymbol{\Psi}$. Therefore, steady-state temperature can be modeled as a linear time-invariant (LTI) system [17]. To describe a statistical thermal model, Eq(2) can be further derived into:

$$\boldsymbol{T} = \boldsymbol{\Psi P} \sim N(\boldsymbol{\Psi \mu_P}, \boldsymbol{\Psi \Sigma_P \Psi}^T) = N(\boldsymbol{\mu_T}, \boldsymbol{\Sigma_T}) \qquad \text{Eq(8)}$$

where $\boldsymbol{\mu_T}$ represents the mean temperature of each grid, and $\boldsymbol{\Sigma_T}$ is the covariance matrix. We point out that only the diagonal entities of $\boldsymbol{\Sigma_T}$ are informative, and hence can be converted into a (n × 1) vector denoted as $\boldsymbol{\Sigma_T^d}$. From Eq(8), the temperature $\boldsymbol{T}$ = $[T_1, \cdots, T_n]$ of a chip follows a multivariate Gaussian distribution, which means the temperature of each grid T_i is a Gaussian distribution with mean μ_{Ti} and variance Σ_{Ti}^d. To put it all together, once the proposed multi-objective optimization framework determines a floorplan, $\boldsymbol{\varpi}$ and the corresponding $\boldsymbol{\mu_T}$ and $\boldsymbol{\Sigma_T^d}$ can be calculated by using Eq(8).

Let us examine the criterion of thermal optimization again. One of the major goals in this work is to minimize the peak temperature T^{max} of a system by finding the best $\boldsymbol{\varpi}$. T^{max} can be calculated by the infinity norm of $\boldsymbol{T}$ from Eq(8), denoted as $\|\boldsymbol{T}\|_\infty$ = $max\{T_1 \ldots T_n\}$ [27]. As Eq(8) shows, $\boldsymbol{T}$ is a random variable, which means T^{max} will also be a random variable. Due to adding the thermal variance Σ_{Ti}^d, the location and magnitude of the original thermal hotspot may vary. To truly mitigate T^{max}, we need to minimize both $\|\boldsymbol{\mu_T}\|_\infty$ (or μ_T^{max}) and $\|\boldsymbol{\Sigma_T^d}\|_\infty$ (or Δ_T^{max}).

C. *Box-Cox transformation*

In this work, we rely on the linearity of a Gaussian distribution to develop the statistical thermal model. In practice, the leakage variations are unlikely to exactly follow a Gaussian distribution. In such cases, it is necessary to transform the data so it fits a Gaussian distribution. Data transformation is usually applied so that the data appear to more closely meet the assumptions of a statistical procedure that is to be applied [28]. Also, data transformation is nearly always invertible. One of the widely-used methods in the statistics community for data transformation is the Box-Cox transformation [15]. For the ease of explanation, we focus on univariate random variable, and rewrite I_{leak} of Eq(3) into y, because multivariate random variables can be transformed in a similar fashion. The Box-Cox transformation of variable y is given by:

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0 \\ ln(y) & \text{for } \lambda = 0 \end{cases} \qquad \text{Eq(9)}$$

where ln is the natural log function and λ is the fitting parameter. Given a set of data $y_1 \ldots y_k$, the best λ will be chosen to maximize the following likelihood function:

$$L(\lambda) = -\frac{k}{2} ln\Big(\frac{1}{k}\sum_{i=1}^{k}(y^{(\lambda)} - \overline{y^{(\lambda)}})^2\Big) + (\lambda - 1)\sum_{i=1}^{k} ln(y_i) \qquad \text{Eq(10)}$$

where $\overline{y^{(\lambda)}}$ is the average of transformed data. The best λ can be calculated by setting $dL/d\lambda = 0$, which is also known as the maximum likelihood estimate (MLE) [16].

To evaluate the effectiveness of Box-Cox transformation, we compare the normality of the original data and the transformed data. The dataset used here is the leakage measurement from 1,000 dies of an industrial design in 45nm technology. We perform the popular Jarque–Bera normality test [29] on both original and transformed datasets under the null hypothesis that the data has a Gaussian distribution. The null hypothesis will be rejected at the 5% significance level. The original dataset was rejected, whereas the transformed dataset was accepted as Gaussian distributed data. This result confirms the effectiveness of Box-Cox transformation which we are using in our framework. In [3], the authors also demonstrated the effectiveness of Box-Cox transformation.

D. *Multi-constrained optimization*

Finding the best $\boldsymbol{\varpi}$ to minimize μ_T^{max}, Δ_T^{max}, L_{itr} and A_{chip} simultaneously is not a trivial task. Here, we rely on simulated annealing (SA), for this multi-constrained optimization.

1. Simulated annealing

Simulated annealing (SA) is well-known for its nonzero probability of accepting inferior solutions. The probability of accepting inferior solutions is typically defined by:

$$Prob = min\{1, e^{-\Delta\eta/T_{SA}}\} \qquad \text{Eq(11)}$$

where $\Delta\eta$ is the cost difference between the neighboring state and the current state, and T_{SA} is the current annealing temperature. In classical SA process, T_{SA} is reduced by a fixed ratio ξ, usually around 0.85–0.95 [8], during each annealing iteration. During each iteration, B*-tree creates a new floorplan by randomly rotating its two nodes, calculates η and $Prob$ of Eq(11), and then determines whether to keep this floorplan as the best floorplan or to discard it. This annealing process will repeat until T_{SA} reaches 0, and finally outputs the best floorplan $\boldsymbol{\varpi}^*$ [8].

2. Adaptive floorplanning: η selection

One straightforward method to select η is to lump all criteria into the cost function:

$$\eta = \alpha \times A_{chip} + \beta \times L_{itr} + \gamma \times \mu_T^{max} + \delta \times \Delta_T^{max} \qquad \text{Eq(12)}$$

where A_{chip} is the total chip area, L_{itr} is the total on-chip interconnect length, μ_T^{max} is the maximum of thermal mean, and Δ_T^{max} is the maximum of thermal variance as defined in Section IV.B. $\alpha, \beta, \gamma, \delta$ are the weights of A_{chip}, L_{itr}, μ_T^{max} and Δ_T^{max}, respectively. Once $\alpha, \beta, \gamma, \delta$ are determined, η can be calculated based on the floorplan generated by B*-tree at each iteration; then η will be plugged into Eq(11) to calculate $Prob$. Note that μ_T^{max} and Δ_T^{max} can be calculated by using Eq(8). However, according to our experimental results, we found that optimizing these four criteria simultaneously by using conventional SA was not able to provide a floorplan with lower μ_T^{max}.

Inspired by [30], we developed an adaptive two-stage optimization framework, aiming at finding the desired thermal-aware solution space. In the first stage, we modified the algorithm proposed by [8] to optimize A_{chip} and L_{itr} only. In the second stage, we use the result from the first stage as the initial solution of SA, and only consider the chip area and the thermal costs. The intuition behinds this method is that B*-tree rotates only two nodes to generate a new floorplan at each SA iteration, so the new and old floorplans should share a large proportion of similarities. Since we use the solution from the first stage, which has small A_{chip} and L_{itr}, as an initial solution of the second stage, the resulting floorplans in the second stage are likely to have small A_{chip} and L_{itr}, too. Also, at each iteration of SA process, if a floorplan generated by B*-tree is illegal, this floorplan will be discarded and SA continues on the next iteration.

In the first stage of the proposed SA process, we introduce an adaptive mechanism to gradually increase the interconnect-related weight. In the very beginning, α is set to 1/10 of the interconnect -related weight β. The main goal is to optimize the floorplan on-chip interconnect first. During SA iterations, we record the total number of legal solutions, and increase α gradually by the ratio of legal solutions and total solutions as:

$$\alpha' = \alpha \times (2 - \frac{N_{legal}}{\Gamma}) \qquad \text{Eq(13)}$$

where N_{legal} is the number of legal floorplans, and Γ represents the duration to the update α. Γ is empirically set to 1,000 iterations. By using this adaptive update, we can gradually guide the floorplan solutions from SA to satisfy both A_{chip} and L_{itr} constraints. In the second stage, we use the best floorplan from stage 1 as an initial solution, and we do not include interconnect-related costs to the cost function. Therefore, the cost function in stage 2 is defined as:

$$\eta = \alpha \times A_{chip} + \gamma \times \mu_T^{max} + \tfrac{\gamma}{\tau} \times \Delta_T^{max} \qquad \text{Eq(14)}$$

where τ is an adjusting parameter and empirically set to 2. Therefore, when SA finds the floorplan candidate with the lowest total cost, usually it also has the lowest temperature.

V. Implementation

Figure 4 illustrates the implementation flow of this work. First, the variation parameters described in Section II.C to our in-house simulator based on [2] to generate $\boldsymbol{L_{eff}}$ within each die. Based on the $\boldsymbol{L_{eff}}$ values, the leakage currents $\boldsymbol{I_{leak}}$ are characterized by using HSPICE simulation with the 45nm high performance Predictive Technology Model (PTM) [32], to calculate the corresponding $\boldsymbol{\mu_I}, \boldsymbol{\Sigma_I}$. Next, the attributes of each module, such as P_{Mi}^{dyn} and W_i described in Section III, along with $\boldsymbol{\mu_I}, \boldsymbol{\Sigma_I}$ are fed as inputs into our Adaptive Floorplanner described in Section IV.D.2. If $\boldsymbol{\mu_I}$ and $\boldsymbol{\Sigma_I}$ do not exactly follow Gaussian distributions, Box-Cox transformation described in Section IV.C will be applied to transform them into Gaussian distributions so as to provide the linearity needed for building the statistical thermal model described in Eq(8). We modify the steady-state temperature calculation of Hotspot5.0 [7] to include *thermal-leakage feedback loop*, and then embed it into Adaptive Floorplanner for calculating $\boldsymbol{\Psi}$ to estimate μ_T^{max} and Δ_T^{max} in each SA iteration. After that, the best floorplan $\boldsymbol{\varpi}^*$ is generated and fed into the Power Estimator. Power estimator calculates the whole-chip power consumption under leakage variations $\boldsymbol{P_{\varpi}}$ by using Eq(7). Finally, with $\boldsymbol{P}$ and $\boldsymbol{\varpi}^*$, Hotspot5.0 thermal simulator calculates $\boldsymbol{T}$ and T^{max} distributions for statistical thermal evaluations. Table 1 lists the detail settings of Hotspot parameters. The parameters not mentioned here are assumed to be the default values. For the Adaptive Floorplanner, we use the same configuration as [8]: the aspect ratio of a floorplan is set to one and the outline is fixed.

Next, we introduce the multi-faceted benchmark suite used in this paper.

- *Alpha21264*: we extract the size of each module from [7], and build the on-chip interconnects based on [19]. Then we use Wattch [31] as the power simulator, and modified the corresponding leakage power model based on [26] and Section II.B. Finally we execute SPECcpu2000 benchmarks on *Alpha21264* to obtain the power profiles. Among all SPECcpu2000 benchmarks, the power profile of *mesa*, an application with extreme thermal demands, will lead to the worst-case thermal scenario. We select the peak power from *mesa*'s power profile as the power benchmark, i.e, P_{Mi}^{dyn} and P_{Mi}^{leak}, for each module of *Alpha21264*.
- *E3S*: *E3S* contains the area and power information of 17 processors and each processor is characterized based on the execution of 47 tasks and its datasheet. To create a synthetic benchmark close to a general SoC design, we select ten processors, and build the on-chip interconnection based on commonly-used task graphs, such as image processing tasks including JPEG compression and RGB-to-YIQ conversion. Power values here are scaled according to the simulation with 45nm high performance Predictive Technology Model [32].
- *MCNC*: these benchmarks are provided by [14]. The number of interconnect constraints ranges from 2 to 18. Each interconnect needs to go through 2–7 modules according to the constraints. The power of each module is assigned according to the power consumption reported by [33], with the total thermal design power (TDP) of 150W treated as peak power.

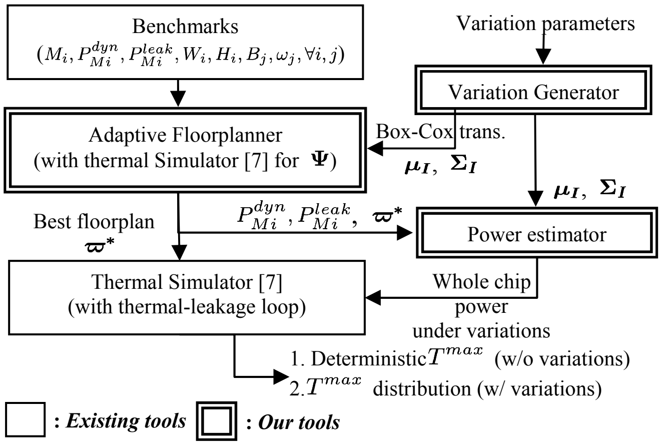

Benchmarks
$(M_i, P_{Mi}^{dyn}, P_{Mi}^{leak}, W_i, H_i, B_j, \omega_j, \forall i, j)$
Variation parameters
Variation Generator
Adaptive Floorplanner
(with thermal Simulator [7] for $\boldsymbol{\Psi}$)
Box-Cox trans.
$\boldsymbol{\mu_I},\ \boldsymbol{\Sigma_I}$
$\boldsymbol{\mu_I},\ \boldsymbol{\Sigma_I}$
Best floorplan
$\boldsymbol{\varpi}^*$
$P_{Mi}^{dyn}, P_{Mi}^{leak},\ \boldsymbol{\varpi}^*$
Power estimator
Thermal Simulator [7]
(with thermal-leakage loop)
Whole chip
power
under variations
1. Deterministic T^{max} (w/o variations)
2. T^{max} distribution (w/ variations)
: Existing tools
: Our tools

Figure 4: Flow chart of the proposed framework.

Parameters	Values
Chip size	0.039m×0.039m.
Heat sink thermal res.	0.48, based on [7].
Heat spreader size	Same as the Chip size.
Heat sink size	0.06m×0.06m.

Table 1. Hotspot Setup.

VI. Experimental Results

This section presents the experiment results, including overall comparisons and statistical thermal evaluations.

A. Overall Comparisons

To evaluate the proposed framework for multi-constrained thermal optimization, we re-implement the method proposed by [6] and [8]. [6] is the state-of-the-art of thermal-aware floorplanner, and [8] is a the state-of-the-art of performance-driven floorplanner. For clear expression, we denote [6] as T[6] (thermal), and [8] as P[8] (performance). Table 2 lists detailed comparisons among T[6], P[8] and our proposed method. Since SA is not a deterministic methodology and all three methods are based on SA, we execute T[6], P[8] and our proposed method for ten times and calculate the average of each target criterion. For our method and P[8], each instance of SA can be finished within 60 seconds, whereas T[6] requires on average 974 seconds to finish one instance of SA. We report these average values as the representative results. To show the extra design overhead required to trade with thermal optimization, the results of chip area, total on-chip interconnect lengths are normalized to P[8].

Let us take a look on the chip area and on-chip interconnect lengths first. As the results show, on average, our proposed method ties with P[8] on the chip area, while T[6] requires slightly bigger chip space. As to on-chip interconnect lengths, our method outperforms T[6] and P[8] by 4% and 2%, respectively. It is worth mentioning, when the number of modules increases (e.g., in *ami49-1* and *ami49-2*), T[6] requires more than 10% longer interconnect lengths than ours.

Bench.	Block #	Inter-connect #	Avg A_{chip}			Avg L_{itr}			Avg T^{max}(˚C)			Avg Thermal-induced leakage			T^{max} distribution					
															T^{max} Mean (˚C)			T^{max} Variance		
			P[8]	T[6]	Ours	P[8]	T[6]	Ours	P[8]	T[6]	Ours	P[8]	T[6]	Ours	P[8]	T[6]	Ours	P[8]	T[6]	Ours
Alpha	27	7	1	0.99	0.98	1	1.01	0.95	96.25	94.01	93.70	1	0.97	0.96	103.06	99.16	97.13	1	0.90	0.87
E3S	10	2	1	1.01	0.99	1	1.02	1.01	100.9	95.13	96.21	1	0.93	0.95	108.79	100.23	100.26	1	0.91	0.85
ami33-1	33	8	1	1.03	1.01	1	1.02	0.97	96.29	91.98	92.75	1	0.94	0.95	102.53	97.77	97.11	1	0.97	0.93
ami33-2	33	18	1	1.02	1.00	1	1.03	1.02	95.88	91.55	92.54	1	0.92	0.94	101.48	96.15	97.83	1	1.02	0.99
ami49-1	49	9	1	1.00	0.99	1	1.02	0.92	98.13	93.25	91.26	1	0.95	0.92	103.71	98.60	94.93	1	0.92	0.88
ami49-2	49	12	1	1.00	1.01	1	1.05	0.91	96.53	94.78	91.20	1	0.93	0.87	104.02	100.20	95.29	1	0.89	0.84
apte	9	5	1	1.01	1.00	1	0.98	0.93	111.96	109.47	108.08	1	0.99	0.97	118.78	114.35	112.04	1	0.87	0.83
xerox	10	6	1	0.99	0.99	1	0.99	1.10	82.15	77.54	79.90	1	0.87	0.88	88.95	82.65	82.93	1	0.85	0.89
avg	27.50	8.38	1	1.01	1.00	1	1.02	0.98	97.26	93.46	93.21	1	0.94	0.93	103.92	98.64	97.19	1	0.92	0.89

Table 2. Overall comparison between existing work [6], [8] and our proposed framework.

As to the peak temperature of a system, T^{max}, on average we achieve 4˚C reduction on each benchmark in all scenarios compared to P[8]. The maximum reduction happens at *ami49-1*, 6.9˚C is reduced. More interestingly, floorplans generated by our method constantly have lower T^{max} than P[8]. Our method achieves almost the same average T^{max} as T[6] does. In some cases, such as *ami33-1* and *ami33-2*, T[6] outperforms ours by 1 ˚C because the resulting areas are larger.

For completeness, we also report the normalized results of thermal-induced leakage. As the results show, our method consistently achieves lower leakage power than P[8]. The maximum reduction happens at *ami49-2,* 13% of thermal-induced leakage reduction is achieved. On average, we reduce 7% of the thermal-induced leakage power. T[6] achieves almost the same leakage reduction as our method does.

B. STATISTICAL THERMAL EVALUATIONS

Here, we perform statistical thermal evaluations for all three methods. For each benchmark, we first select a floorplan generated by P[8] with the smallest chip area, and the corresponding ones generated by our proposed method and T[6] with approximately the same chip area. Next, we generate 1,000 samples to emulate the leakage variations, $\boldsymbol{I_{leak}}$, on 1,000 dies by using the settings described in Section V. Given $\boldsymbol{I_{leak}}$ and the floorplan provided by T[6], P[8] or our method, the whole chip power under variations $\boldsymbol{P_{\varpi}}$ of each benchmark can be calculated and fed into Hotspot5.0 for thermal simulations.

Let us focus on the mean of the T^{max} distribution first. Compared to P[8] and T[6], our method can reduce the mean by 6.7˚C and 1.5˚C on average. The trend is similar to the T^{max} reduction without considering leakage variations. The maximum reduction happens at *ami49-1* where the temperature reduction is 8.78˚C. As to the variance of the T^{max} distribution, our method achieves on average 11% and 3% reduction compared to P[8] and T[6], respectively. It is worth mentioning that our method constantly outperforms P[8] on temperature variance across all benchmarks, but T[6] does not (*ami33-2*). All these results demonstrate the strength of our proposed framework to statistically mitigate the thermal hotspots of a system, without introducing any design overheads.

VII. CONCLUSION

In this paper, we provide a statistical thermal model and the corresponding optimization framework, which aims at reducing peak temperature of a system under leakage variations without compromising any performance or introducing any design overhead. The experimental results show that up to 8.8˚C, 13% thermal-induced leakage power, and 17% maximum thermal variance are reduced with no additional design cost or overhead.

ACKNOWLEDGEMENT

The authors would like to thank Huapeng Zhou and the anonymous reviewers for the constructive suggestions.

REFERENCES

[1] D. Brooks et al., “Power, thermal, and reliability modeling in nanometer-scale microprocessors,” *MICRO*, 2007.

[2] L. Cheng et al., “Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability,” *DAC*, 2009.

[3] S. Reda et al., “Analyzing the impact of process variations on parametric measurements: novel models and applications,” *DATE*, 2009.

[4] S. Borkar et al., “Parameter variations and impact on circuits and microarchitecture,” *DAC*, 2003.

[5] D.-C. Juan et al., “Statistical thermal evaluation and mitigation techniques for 3D chip-multiprocessors in the presence of process variations,” *DATE*, 2011.

[6] K. Sankaranarayanan et al., "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," *The Journal of Instruction-Level Parallelism*, 2005.

[7] W. Huang et al., “HotSpot: thermal modeling for CMOS VLSI systems,” *TCPMT*, 2005.

[8] T.-C. Chen et al., “Modern floorplanning based on B*-tree and fast simulated annealing,” *TCAD*, 2006.

[9] B. Goplen et al., “Thermal via placement in 3D IC,“ *ISPD* ,2005.

[10] J. Cong et al., “Thermal via planning for 3-D ICs,” *ICCAD*, 2005.

[11] T. Ebi et al., “TAPE: thermal-aware agent-based power economy for multi/many-core architectures,” *ICCAD*, 2009.

[12] Y.-L. Chuang et al., "Voltage-drop aware analytical placement by global power spreading for mixed-size circuit designs, “ *ICCAD*, 2009.

[13] C. Zhuo et al., “Process variation and temperature-aware reliability management,” *DATE*, 2010.

[14] H. Xiang et al., “Bus-driven floorplanning,” *ICCAD*, 2003.

[15] G. Box et al., “An analaysis of transformations,” *J. Royal Statistical Soc.*, 1964.

[16] L. Wasserman, “All of statistics: a concise course in statistical Inference,” *Springer Texts In Statistics*, 2004.

[17] L. Ljung, “System identification: theory for the user,” *Prentice-Hall*, 1987.

[18] Y.-C. Chang et al., “B*-Trees: a new representation for non-slicing floorplans,” *DAC*, 2000.

[19] R. E. Kessler, “The alpha 21264 microprocessor,” *MICRO*, 1999.

[20] R. Dick, “E3S: The embedded system synthesis benchmarks suite,” *E3S link at http://robertdick.org/tools.html*.

[21] S. Herbert et al., “Analysis of dynamic voltage/frequency scaling in chip-multiprocessors,” *ISLPED*, 2007.

[22] C. Kim et al., “Dynamic VTH scaling scheme for active leakage power reduction,” *DATE*, 2002.

[23] “International technology roadmap for semiconductors,” 2009.

[24] “BSIM4, http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html,” 2011.

[25] J. Sartori et al., “Variation-aware speed binning of multi-core processors," *ISQED*, 2010.

[26] J. A. Butt et al., “Static Power Model for Architects,” *MICRO*, 2000.

[27] S. Leon, “Linear algebra with applications,” *Prentice Hall*, 1998.

[28] R. Johnson et al., “Applied multivariate statistical analysis,” *6th ed. Englewood Cliffs*, 2007.

[29] C. Jarque et al., “Efficient tests for normality, homoscedasticity and serial independence of regression residuals,” *Economics Letters*, 1980.

[30] L. Xiao et al., “Fixed-outline Thermal-aware 3D Floor planning,” *ASPDAC*, 2010.

[31] D. Brooks et al., “Wattch: a framework for architectural-level power analysis and optimizations”, *ISCA*, 2000.

[32] W. Zhao et al., "New generation of Predictive Technology Model for sub-45nm early design exploration," *IEEE Transactions on Electron Devices*, 2006.

[33] S. Li et al., “McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures,” *MICRO*, 2009.