Dynamic Thermal Estimation Methodology for High-Performance 3-D MPSoC

Amir Zjajo, *Member, IEEE*, Nick van der Meijs, *Member, IEEE*, and Rene van Leuken, *Member, IEEE*

***Abstract*— In 3-D integrated circuits, accurate runtime sensing of on-chip temperature is required to establish dynamic thermal management instruction sets. Placement restrictions and excessive runtime thermal variations, however, compromise the performance and reliability of the sensor readings. Within this framework, a novel methodology for thermal estimation based on unscented Kalman filter, augmented only with a limited number of temperature sensors at a few selected locations, is proposed. In addition, we extend discontinuous Galerkin finite-element method to include coupling mechanism between neighboring grid cells for accurate thermal profile estimation and introduce a balanced stochastic truncation to find a low-dimensional but accurate approximation of the thermal network over the whole frequency domain. As the experimental results show, the runtime thermal estimation method reduces temperature estimation errors by an order of magnitude.**

***Index Terms*— 3-D integrated circuits (ICs), multiprocessor system-on-chip (SoC), simulation, thermal analysis, thermal management.**

I. Introduction

IN THE nanometer regime, the transistor scaling has been slowing down due to the challenges and hindrances of increasing variability, short-channel effects, power/thermal problems, and the complexity of interconnect. The 3-D integration has been proposed as one of the alternatives to overcome the interconnect restrictions [1]. Thermal management is, however, of critical importance for 3-D integrated circuit (IC) designs [2] due to the degradation of performance and reliability [3]. Heat and thermal problems are exacerbated for 3-D applications as the vertically stacked multiple layers of active devices cause a rapid increase of power density. Higher temperature increases the risk of damaging the devices and interconnects (as major back-end and front-end reliability issues including electromigration, time-dependent dielectric breakdown, and negative-bias temperature instability have strong dependence on temperature), even with advanced thermal management technologies [4]. The complexity of the interconnection structures, back end of line structures and through-silicon vias increase the complexity of the conductive heat transfer paths in a stacked die structure. Dummy vias and intertier connections can be used to increase the vertical heat transfer through the stack and reduce the temperature peaks in the die [5]. Successful application of 3-D integration requires analysis of thermal management problem, and the development of an analytical model for heat transport in 3-D ICs to establish thermal design rules governing the feasibility of integration options. A thermal analysis of heterogeneous 3-D ICs with various integration schemes has been presented in [6]. The analysis of temperature distribution on an inhomogeneous substrate layer is performed employing finite-difference time domain [7], based on the image method [8], neural networks [9], green function [10], fast Hankel transform of green function [11], or mesh based methods [12]. However, existing thermal-simulation methods, when applied to a full chip, reduce the computational complexity of the problem by homogenizing the materials within a layer, limiting the extent of an eigenfunction expansion, or ignoring sources' proximity to boundaries. These simplifications render their results less accurate at fine length scales on wires, vias, or individual transistors. Accurate computation of temperature at the length scales of devices and interconnects requires the development of a fundamental analytical model for heat transport in 3-D ICs and a detailed accounting of the heat flow from the power sources through the nanometerscale layout within the chip.

The thermal conductivity of the dielectric layers inserted between the device layers for insulation is very low compared with silicon and metal [13] leading to temperature gradient in the vertical direction of a 3-D chip. In hotspots, these thermal effects are even more pronounced. Therefore, continuous thermal monitoring is necessary to reduce thermal damage and increase reliability. Built-in temperature sensors predict excessive junction temperatures as well as the average temperature of a die within design specifications. Underlying chip power density is, however, highly random due to unpredictable workload, fabrication randomness, and nonlinear dependence between temperature and circuit parameters. Increasing the number of sensors could possibly resolve this issue; nevertheless, the cost of adding a large number of sensors is prohibitive. Moreover, even without considering the cost of added sensors, other limitations such as additional channels for routing and input/output may not allow placement of thermal sensors at the locations of interest. Several techniques have been proposed to solve the problem of tracking the entire thermal profile based on only a few limited sensor observations [14]–[20]. Among these techniques, the Kalman filter (KF)-based methods are especially resourceful as such

Manuscript received July 30, 2012; revised April 17, 2013; accepted August 29, 2013. This work was supported by the CATRENE program under the Computing Fabric for the High Performance Applications under Project CA104.

The authors are with Circuits and Systems, Delft University of Technology, Delft 2628, The Netherlands (e-mail: amir.zjajo@ieee.org; n.p.vandermeijs@tudelft.nl; t.g.r.m.vanleuken@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2013.2280667

methods are capable of exploiting the statistical properties of power consumption along with sensor observations to estimate temperatures at all the chip locations during runtime, while simultaneously retaining the possibility to incorporate associated sensor noise caused by fabrication variability, supply voltage fluctuation, cross coupling and so on. Existing KF-based approaches, however, imply a linear model ignoring the nonlinear temperature-circuit parameters dependency or employ a linear approximation of the system around the operating point at each time instant. These approximations, however, can introduce large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to suboptimal performance and sometimes divergence of the filter.

In this paper, we propose statistical linear regression technique based on unscented KF (UKF) to explicitly account for this nonlinear temperature-circuit parameters dependency of heat sources, whenever they exist. Because we are considering the spread of random variable, the technique tends to be more accurate than Taylor series linearization employed in existing KF-based approaches. As the experimental results show, the runtime thermal estimation method reduces temperature estimation errors by an order of magnitude. In addition, we extend the study for accurate thermal profile estimation based on discontinuous Galerkin finite-element method [21] to include coupling mechanism between neighboring grid cells. The extended method provides both steady-state and transient 3-D temperature distribution and can be used to simulate geometrically complicated physical structures with limited complexity overhead. To reduce computational complexity, we adopt a more stable semi-implicit treatment of the numerical dissipation terms in Runge–Kutta solver and introduce a balanced stochastic truncation to find a low-dimensional but accurate approximation of the thermal network over the whole frequency domain.

This paper is organized as follows. Section II focuses on the thermal conduction in ICs and associated thermal model. Section III introduces the UKF for temperature estimation. In Section IV, two algorithms are described, namely modified Runge–Kutta method for fast numerical convergence, and a balanced stochastic truncation for accurate model order reduction (MOR) of thermal network. Section V elaborates the experimental results. Finally, Section VI provides a summary and the main conclusion.

II. Thermal Model

A 3-D IC contains multiple vertically stacked silicon layers, each containing processing elements (PEs) and memory modules (Fig. 1) [22], [23]. An offline temperature profile estimation methodology [21] has the capability to include layout geometry of individual circuit blocks in a chip (Fig. 2). The model is composed by three types of layers: 1) bulk silicon; 2) active silicon; and 3) the heat-spreading copper layer. The chip is partitioned into a mesh according to the information provided by the layout geometry and power distribution map.

Nominal power distribution (including switching and leakage power dissipation) for each functional unit according to its activity factor is assigned an initial value. Each functional unit in the floorplan is represented by one or more thermal cells of the silicon layer (Fig. 3). Physical parameters such as thermal conductivity and heat transfer coefficient depend on specific packaging material properties and applied cooling techniques. Boundary conditions are determined by the operating environment. The simulator uses layout geometry, power distribution, boundary conditions, and physical thermal parameters as initial values to formulate the system of partial differential equations, which are approximated into a system of ordinary differential equations (ODEs) with discontinuous Galerkin method. The first step in discontinuous Galerkin finite-element discretizations is to form weak formulation/algebraic system: the variables are expanded in the domain or in each element in a series in terms of a finite number of basis functions. Each basis function has compact support within each element.

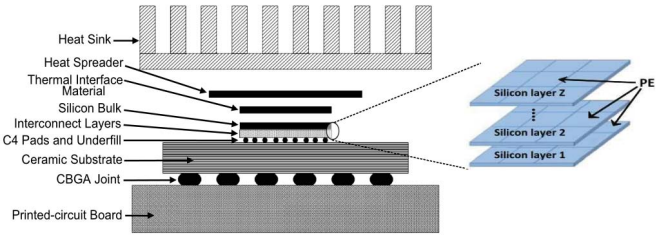

Fig. 1. 3-D chip package with PEs on vertically stacked silicon layers [22], [23].

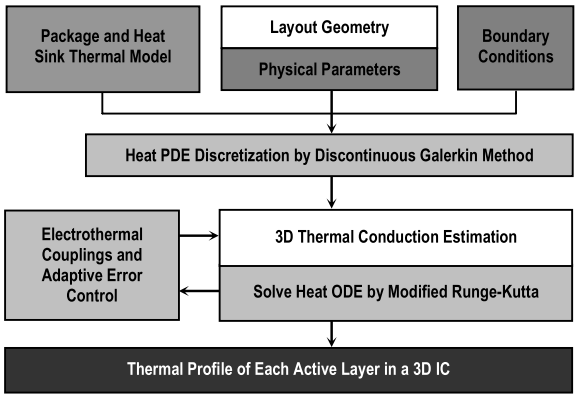

Fig. 2. Offline setup of the methodology for thermal profile estimation [21].

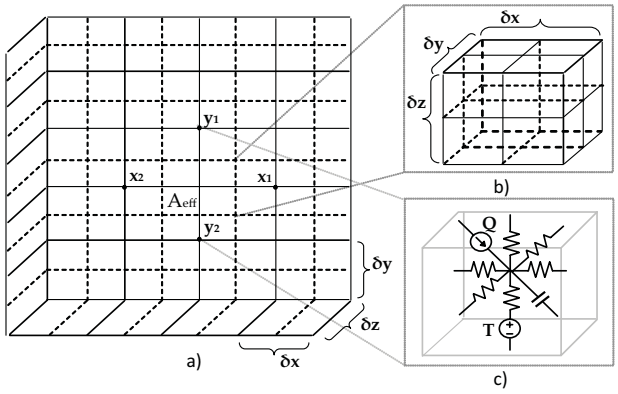

Fig. 3. (a) Chip top view. (b) 3-D view of the grid point a. (c) Equivalent electrical circuit for each cell.

This expansion is then substituted into the weak formulation, and a test function is chosen alternately to coincide with a basis function, to obtain the discretized weak formulation. Next, the integrals are evaluated in local coordinate system and global matrices and vectors are assembled in the assembly routine. The resulting ODEs are then numerically integrated in a self-consistent manner using modified Runge–Kutta method. To control the error due to the surface approximation, we evaluate the magnitude of the difference between the analytical distribution of temperature T, and an interpolation of this function on a finite-element edge length. The errors of interpolation increase when the heat is changing faster (the higher the curvature of the function of the exact temperature T).

To control this error, we employ l-adaptive control [21] by designing graded meshes, with small elements located in regions of expected high error, and proportionally large elements elsewhere. To accurately estimate power dissipation and resulting temperature profile, the electrothermal couplings are also embedded in the core of the simulator that simultaneously estimates temperature-dependent quantities for each simulation step. The scheme based on [24] and extended in [25] uses instantaneous temperature monitoring coupled with information on the physical structure of the die stack to determine operating voltage–frequency levels for PEs.

A. Thermal Conduction in ICs

Fundamentally, IC thermal modeling is the simulation of heat transfer from heat producers (transistors and interconnect), through silicon die and cooling package, to the ambient environment. A schematic representation of the chip layer and its thermal mesh model is shown in Fig. 3. The chip is divided into meshes according the layout geometry and power distribution map in the x-, y-, and z-directions, here, δx, δy, and δz are each mesh's side sizes. The Fourier equation governing heat diffusion via thermal conduction in an IC is as follows:

$$c_V \frac{\partial T}{\partial t} = \nabla \cdot g(\nabla T)^T + Q \tag{1}$$

where Q is the heat source, T is the temperature at time t, c_V is a capacitance of the volume V, $\nabla T = [\partial T/\partial x, \partial T/\partial y, \partial T/\partial z]$, and the matrix g is the conductivity matrix of material with three orthogonal directions of different thermal conductivities $g = \text{diag}(g_a)$, $a = x, y, z, g_x, g_y$, and g_z are the thermal conductivities coefficients. The source of heat generation Q depends on the nature of the circuit operation. At the device simulation level, it is the local Joule heat as a function of current density and electric field, and at the block level, it can be assumed that the power consumption for the functional block under the typical signal pattern is the source for the entire block. To approximate the solutions of these equations using numerical methods, we use finite discretization, i.e., an IC model is decomposed into numerous 3-D elements, where adjacent elements interact via heat diffusion.

Each element is sufficiently small to permit its temperature to be expressed as a difference equation, as a function of time, its material characteristics, its power dissipation, and the temperatures of its neighboring elements. The temperature in the control volumes along the boundaries of the computational domain is determined using constraints representing boundary conditions. Each cell is assigned the specific heat capacity of the associated material and also a temperature. If a dual grid is formed by joining the centers of adjacent cells, each edge of the dual grid will intersect exactly one face of the primary grid. The thermal conductivity can be thought to be assigned for the edge of the dual grid. If the two cells on either side of the face belong to the same material, the assigned thermal conductivity is that of the material. If the two cells belong to different materials, the thermal conductivity is chosen on the basis of the thermal conductivity values of both the materials. We also allow for the existence of interfacial thermal resistance (due to scattering of thermal carriers at the interface).

B. Thermal Conduction Model

We take up the Galerkin finite-element discretization for the thermal conduction initial boundary value problems. Balancing the order of differentiation by shifting one derivative from the temperature to the test function η is beneficial: we use basis functions that are less smooth because we do not require the second derivatives, and also we are able to satisfy the natural boundary conditions without having to include them as a separate residual. The integration by parts in the case of a multidimensional integral is generalized in the divergence theorem. The surface heat transfer coefficient h is defined as $h = 1/(A_{\text{eff}} R)$, where A_{eff} is the effective area normal to the direction of heat flow and R is the equivalent thermal resistance. We assume a Dirichlet boundary condition of the form $T = 0$ (absolute temperature equal to ambient temperature) at the radial and the $z = \max(z)$ boundaries. This condition is applied by setting the temperature at the center of the boundary cells along the radial and the $z = \max(z)$ boundaries to zero. Note that the boundary conditions are specific to the package design. Although different packages with varying heat sink properties would change the boundary conditions, the general nature of the solution will not change. The boundary condition at $z = \min(z)$ is assumed to be of the mixed type $g_z \partial T/\partial z - hT = 0$, where g_z is the thermal conductivity in the z-direction. Physically, this corresponds to heat loss being proportional to the difference between the absolute temperature and the ambient temperature. To simplify the problem, we reduce the originally 3-D model to two active coordinates, while still describing the heat conduction through a 3-D domain; the function describing the temperature distribution depends only on two spatial coordinate variables though.

The surface of the 3-D solid consists of the two cross sections, and of the cylindrical surfaces, the inner and the outer. The two cylindrical surfaces may be associated with boundary condition of any type. We simplify calculation by preintegrating in the thickness direction, $\mathrm{d}V = \Delta z \mathrm{d}\mathcal{S}$ and $\mathrm{d}\mathcal{S} = \Delta z \mathrm{d}\mathcal{C}$. The volume integrals are then evaluated over the cross-sectional area $\mathcal{S}_c$, provided h is independent of z; the surface integrals are computed as integrals over the contour of the cross section $\mathcal{C}_c$. Adding the surface (Newton) boundary

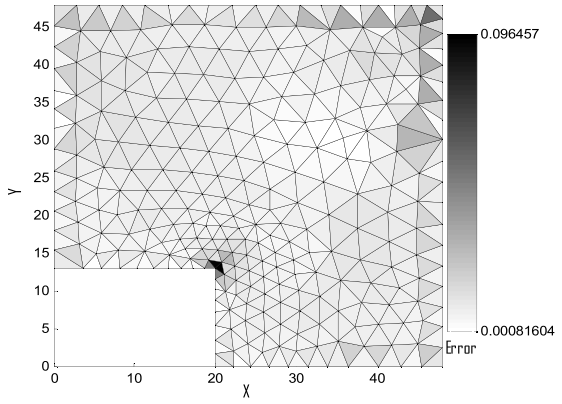

Fig. 4. Surface approximation of L-form wire with collection of triangles.

condition residual, (1) is expressed as

$$\int_{\mathcal{S}_c} \eta c_V \partial T/\partial t \Delta z d\mathcal{S} = \int_{\mathcal{S}_c} \nabla \eta g (\nabla T)^T \Delta z d\mathcal{S} + \int_{\mathcal{S}_c} \eta Q \Delta z d\mathcal{S} + \int_{C_c} \eta h (T - T_a) \Delta z dC \quad (2)$$

where T_a is the known temperature of the surrounding medium. The domain of the surface is approximated as a collection of triangles. As an illustrative example, we show in Fig. 4 L-form wire. As the triangles are the finite elements with straight edges, we are only approximating any boundaries that are curved. This error is controlled by length-adaptive error control [21]. Because the basis on the standard triangle satisfies the Kronecker delta property, the values of the degrees of freedom $T_i(t)$, $i = 1,\ldots, \mathcal{N}_f$, at the i nodes are simply the values of the interpolated temperature at the nodes, $T_i(t) = T(x_i, y_i, t)$. We express the system of ODEs, which results from the introduction of the Galerkin finite-element test function η (the so-called discretization in space) on (2) as

$$\sum_{i=1}^{\mathcal{N}_f} C_{ji} \partial T_i \big/ \partial t = \sum_{i=1}^{\mathcal{N}_f} G_{ji} T_i + P_j, \quad j = 1, \ldots, \mathcal{N}_f \quad (3)$$

where

$$\begin{aligned} C_{ji} &= \int_{\mathcal{S}_c} \mathcal{N}_j c_V \mathcal{N}_i \Delta z d\mathcal{S}, \quad i,j = 1, \ldots, \mathcal{N}_f \\ G_{ji} &= \int_{\mathcal{S}_c} (\nabla \mathcal{N}_j) h_{ji} g (\nabla \mathcal{N}_i)^T \Delta z d\mathcal{S}, \quad i,j = 1, \ldots, \mathcal{N}_f \\ P_j &= P_{Q_j} + P_{C_j} + P_{G_j}, \quad j = 1, \ldots, \mathcal{N}_f \\ P_{Q_j} &= \int_{\mathcal{S}_c} \mathcal{N}_j Q \Delta z d\mathcal{S}, \quad j = 1, \ldots, \mathcal{N}_f. \end{aligned} \quad (4)$$

C_{ji} and G_{ji}, are the capacity and conductivity matrices, respectively, P_{Qj} designates internal heat generation, and $\mathcal{N}$ is piecewise linear Galerkin basis function.

Boundary condition in a weighted residual sense is given as

$$\begin{aligned} P_{C_j} &= \sum_{i=\mathcal{N}_f+1}^{\mathcal{N}} \left[\int_{\mathcal{S}_c} \mathcal{N}_j c_V \mathcal{N}_i \Delta z d\mathcal{S} \right] \partial T_i/\partial t, \quad j = 1, \ldots, \mathcal{N}_f \\ P_{G_j} &= \sum_{i=\mathcal{N}_f+1}^{\mathcal{N}} \left[\int_{\mathcal{S}_c} (\nabla \mathcal{N}_j) h_j g (\mathcal{N}_i)^T \Delta z d\mathcal{S} \right] T_i, \quad j = 1, \ldots, \mathcal{N}_f. \end{aligned} \quad (5)$$

The analogy between heat flow and electrical conduction is invoked here, because they are described by exactly the same differential equations for a potential difference. The temperature is represented as voltage, heat flow represented as electric current, the term on the left-hand side in (3) represented as a capacitor and the rest of the terms on the right-hand side represented as conductances, giving rise to an RC circuit [26]. The resulting thermal network in (3) is represented in state-space form with the grid cell temperatures as states and the power consumption as inputs to this system

$$C_{ji} \left(\frac{dT_i}{dt} \right) = G_{ji} T_i(t) + B_j P_j(t) \quad (6)$$

where C_{ji}, $G_{ji} \in \mathcal{R}^{m_{ji} \times m_{ji}}$ are the matrixes describing the reactive and dissipative parts in the model, respectively, $T_i(t) \in \mathcal{R}^{m_i}$ are the time-varying temperature vectors, $B_j \in \mathcal{R}^{m_j \times p_j}$ is the input selection matrix, and $P_j(t) \in \mathcal{R}^{p_j}$ is the vector of power inputs (heat sources as function of time, wherever they exists). The number of state variables m is called the order of (6) and p is the number of inputs. The outputs of this state-space model are the temperatures at the sensor locations, which are observed by sensor readings $S_j(t) \in \mathcal{R}^{q_j}$

$$S_j(t) = E_j^T T_i(t) \quad (7)$$

where $E_j \in \mathcal{R}^{q_j \times m_j}$ is the output matrix, which identifies the sensor grid cells at which temperatures are observable. For simplicity, and because this holds true for electrical circuits, we restrict ourselves to (7) with $q = p$. We are assuming that distinct measurements are coming from distinct sensors: E_j has only one nonzero element per row. We connect the nodes of the thermal network of the grid cells (Fig. 2) to the nodes of their neighboring cells through the coupling relations

$$\begin{aligned} P_j(t) &= K_{j1} S_1(t) + \cdots + K_{jk} S_k(t) + D_j P(t), \quad j = 1, \ldots, k \\ S(t) &= L_1 S_1(t) + \cdots + L_k S_k(t) \end{aligned} \quad (8)$$

where $K_{jk} \in \mathcal{R}^{p_j \times q}$, $D_j \in \mathcal{R}^{p_j \times p}$, and $L_j \in \mathcal{R}^{q \times q_j}$ are the coupling matrixes. If $I\text{-}H(s)K$ is the invertible, the input–output relation of the coupled system (6)–(8) can be written as $S(s) = \Gamma(s)P(s)$, where $S(s)$ and $P(s)$ are the Laplace transforms of $S(t)$ and $P(t)$, respectively, and the closed-loop transfer function $\Gamma(s)$ has the following form:

$$\begin{aligned} \Gamma(s) &= L(I - H(s)K)^{-1} H(s) D \\ H(s) &= \mathrm{diag}(H_1(s), \ldots, H_k(s)) H_j(s) = E_j^T (sC_j - G_j)^{-1} B_j. \end{aligned} \quad (9)$$

We express a generalized state-space realization of $\Gamma(s)$ by

$$\begin{aligned} \mathcal{C}(dT/dt) &= \mathcal{G}T(t) + \mathcal{B}P(t) \\ \mathcal{C} &= C \in \mathcal{R}^{m,m} \\ \mathcal{G} &= G + BKE^T \in \mathcal{R}^{m,m} \\ \mathcal{S}(t) &= \mathcal{E}^T T(t) \\ \mathcal{B} &= BD \in \mathcal{R}^{m,p} \\ \mathcal{E}^T &= PE^T \in \mathcal{R}^{q,m}. \end{aligned} \tag{10}$$

C. Electrothermal Couplings

Thermal issues arising from the high density of integration in 3-D architectures necessitates the use of aggressive thermal management techniques, and the inclusion of thermal effects in the architecture space exploration stage of the design flow. Given the gravity of thermal issues encountered deep within die stacks, a runtime power management strategy is essential toward ensuring a reliable design. A comprehensive thermal management policy for 3-D multiprocessors incorporating temperature aware workload migration and runtime global power-thermal budgeting is presented in [27]. Within the policy, PEs with available temperature budgets executing high instructions per cycle workloads are scaled to higher voltage and frequency levels to improve their performance after weighing the potential performance benefits of such scaling against the consequent thermal implications for neighboring PEs.

We incorporate a runtime power manager with a thermal simulation engine to yield a methodology for temperature power simulation of 3-D architectures [24]. In multiprocessor system-on-chip, the activity rate is replaced by a cycle accurate trace of each PE execution, suggesting the cycles during which computational operations were performed, and those during which it remained idle. The voltage and frequency levels of PEs are controlled by a custom power management scheme that enables the investigation of the thermal implications of various power management techniques on 3-D stacks. The scheme based on [24] and extended in [25] uses instantaneous temperature monitoring coupled with information on the physical structure of the die stack to determine operating voltage–frequency levels for PEs. Additionally, a weighted policy is adapted while implementing scaling decisions, thereby preventing PEs on deeper tiers from reaching critical temperatures and thus being turned off. The methodology outperforms conventional 2-D dynamic voltage and frequency scaling technique, both in its ability to maintain the temperatures of all PEs stable, as well as in its improvement of performance by increasing the aggregate system frequency [24], [25].

III. Dynamic Thermal Tracking

Complex ICs with large die area require multiple thermal sensors to capture temperatures at a wide range of locations as the unpredictability of a workload leads to continuous migration of hotspots, and within-die manufacturing variations lead to parameter variability that further conceal the locations of the thermal hotspots. However, the thermal sensors, together with their support circuitry and wiring, complicate the design process and increase the total die area and manufacturing costs.

Given the limitations on the number of thermal sensors, it is necessary to optimally place them near potential hot-spot locations. In [28], a clustering algorithm is described that computes the thermal sensor positions, which best serve clusters of potential hot-spot locations. In [29], an optimal sensor problem is computed as the unite-covering problem. In [30], the unknown temperature at a particular location is computed as a weighted combination of the known measurements at other locations. Nevertheless, these techniques may be ineffective for dynamic thermal tracking or if the accuracy or availability of sensors measurements is in question.

A. Preliminary

Several online techniques have been proposed to solve the above problem [14]–[20]. Among these techniques KF-based methods generate thermal estimates for all the chip locations while countering sensor noise and can be applied to real-time thermal tracking problems. The KF propagates the mean and covariance of the probability density function of the model state in an optimal (minimum mean square error) way in linear dynamic systems. As very large scale integration fabrication technology, however, continues to scale down, leakage power can take up to 50% of the total chip power consumption [31]. Note that leakage has the nonlinear nature that increase exponentially with the chip temperature. Therefore, the standard KF tends to underestimate the actual chip temperature due to the assumed linear model. Consider (10) in corresponding discrete-time state space

$$\begin{aligned} T_n &= AT_{n-1} + J(P_{D(n-1)} + P_{L(n-1)}) + r_{n-1} \\ &= AT_{n-1} + JP_{D(n-1)} + JK_1T_{n-1}^2 e^{K_2/T_{n-1}} + r_{n-1} \\ &= f(T_{n-1}) + r_{n-1} \\ S_n &= h(T_n) + u_n \end{aligned} \tag{11}$$

where T_n is the state vector representing temperatures at different grid cells at time n, A, and J are the coefficient matrices determined by the circuit parameters (C and G) and the chosen length of the time step. For clarity, we subdivided power P into two components, dynamic power $P_{D(n-1)}$ and leakage power $P_{L(n-1)}$. While dynamic power consumption $P_{D(n-1)} = (1/2)\alpha C_L V_{\mathrm{DD}}^2 f$, where C_L is the switching capacitance, α is the switching activity of output node, V_{DD} is the supply voltage, and f is the operation frequency of system, is weakly coupled with temperature variation, static power consumption is a strong function of temperature $P_{L(n-1)} = K_1T_{n-1}^2\exp(K_2/T_{n-1})$ [32], where K_1 and K_2 are the design/technology and fixed supply voltage constants, respectively. S_n is the output vector of temperatures at sensor locations, $r_{n-1} \sim \mathrm{N}(0,\mathcal{R}_{n-1})$ is the Gaussian process noise, and $u_n \sim \mathrm{N}(0,\mathcal{U}_n)$ is the Gaussian sensor noise (noise caused by fabrication variability, supply voltage fluctuation, cross coupling, etc.).

Because of unpredictability of workloads (power vector is unknown until runtime) and fabrication/environmental variabilities, the exact value of T_n at runtime is difficult to predict. To elevate the issue, on-chip sensors provide an observation vector S_n, which is essentially a subset of T_n plus sensor

noise u_n. In (11), $h(\cdot)$ is a transformation function determined by the sensor placement. Because of the sensors power/area overheads, their number and placement are highly constrained. Therefore, the problem of tracking the entire thermal profile (vector T_n) based on only a few limited sensor observations S_n is rather complex.

To extend the model for the nonlinear leakage-temperature function $f(\cdot)$, the most common way of applying the KF is in the form of the extended KF (EKF). In the EKF, the probability density function is propagated through a linear approximation of the system around the operating point at each time instant. These approximations, however, can introduce large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to suboptimal performance and sometimes divergence of the filter. In contrast, the UKF, which uses the unscented transform (UT) [33], [34], is using the statistical linearization technique to linearize a nonlinear function of a random variable through linear regression between k data points drawn from *a priori* distribution of the random variable. Because we are considering the spread of random variable, the UT is able to capture the higher order moments caused by the nonlinear transform better than the EKF Taylor series-based approximations [33]. The mean and covariance of the transformed ensemble can then be computed as the estimate of the nonlinear transformation of the original distribution. The UKF outperforms the EKF in terms of prediction and estimation error at an equal computational complexity for general state-space problems [34]. In addition, the UKF can easily be extended to filter possible power estimation noises, restricting the influence of the high-frequency component in power change on the modeling approach.

B. Temperature Estimation

The UKF estimates online the temperature during the normal operation in a predict-correct manner based on inaccurate information of temperature and power consumption. The measurement update incorporates the new measurements into the *a priori* estimate to obtain an improved *a posteriori* estimate of the temperature. A time and measurement update step is repeated for each run of the algorithm. In UKF, the initialization step uses the UT to generate the $2\mathcal{k}+1$ sigma points and appropriate weights $\mathcal{W}$ for the mean m and covariance $\mathfrak{Z}$ computations [34]. The first step in the time update phase is the propagation of the input domain points, which are referred to as sigma points [34], through the nonlinear function in the transition equation (12). Given an $\mathcal{k}$-dimensional distribution with covariance $\mathfrak{Z}$, the *a priori* estimate of a mean for the state vector is computed as a weighted average of the propagated sigma points (13). We compute the *a priori* error covariance from the weighted outer product of the transformed points (14). The covariance $\mathcal{R}_{n-1}$ is added to the end of (14) to incorporate the process noise. To compute the new set of sigma points, we need the square root matrix of the posterior covariance $\mathfrak{Z}_n = \mathcal{A}_n\mathcal{A}_n^T$. A Cholesky decomposition [35] is used for this step for numerical stability and guaranteed positive semidefiniteness of the state covariances [34]

$$T_{i|n} = f(T_{i|n-1}), \quad i = 0, \ldots 2\mathcal{k} \tag{12}$$

$$m_n^- = \sum_{i=0}^{2\mathcal{k}} \mathcal{W}_i^{(m)} T_{i|n} \tag{13}$$

$$\mathcal{A}_n^- = qr\left(\left[\sqrt{\mathcal{W}_i^{(c)}}(T_{i|n} - m_n^-)\ \sqrt{\mathcal{R}_{n-1}}\right]\right)$$

$$\mathcal{A}_n^- = \text{cholupdate}\left(\left[\mathcal{A}_n^-, (T_{0|n} - m_n^-),\ \text{sgn}\{\mathcal{W}_0^{(c)}\}\sqrt{\mathcal{W}_0^{(c)}}\right]\right) \tag{14}$$

where qr function returns only the lower triangular matrix. The weights are not time dependent and do not need to be recomputed for every time interval. The superscripts m and c on the weights refer to their use in mean and covariance calculations, respectively. Note that this method differs substantially from general sampling methods (e.g., Monte Carlo methods such as particle filters, which require orders of magnitude more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state.

The known measurement equation $h(\cdot)$ is used to transform the sigma points into a vector of respective (predicted) measurements (15). The *a priori* measurement vector is computed as a weighted sum of the generated measurements (16)

$$S_{i|n} = h(T_{i|n}), \quad i = 0, \ldots, 2\mathcal{k} \tag{15}$$

$$\mu_n^- = \sum_{i=0}^{2\mathcal{k}} \mathcal{W}_i^{(m)} S_{i|n}. \tag{16}$$

In the correction step, the computation of the Kalman gain (and, therefore the correction phase of the filtering) is based on the covariance of the measurement vector (17), where $\mathcal{U}_n$ is the measurement noise covariance, and the covariance of the state and measurement vectors (18). These are computed using the weights (which were obtained from the UT during the initialization step) and the deviations of the sigma points from their means

$$\mathcal{Z}_n = qr\left(\left[\sqrt{\mathcal{W}_i^{(m)}}(S_{i|n} - \mu_n^-)\ \sqrt{\mathcal{U}_n}\right]\right) \tag{17}$$

$$\mathcal{Z}_n = \text{cholupdate}\left(\left[\mathcal{Z}_n,\ (S_{0|n} - \mu_n^-),\ \text{sgn}\{\mathcal{W}_0^{(m)}\}\sqrt{\mathcal{W}_0^{(m)}}\right]\right)$$

$$\Xi_n = \sum_{i=0}^{2\mathcal{k}} \mathcal{W}_i^{(c)}(T_{i|n} - m_n^-)(S_{i|n} - \mu_n^-)^T. \tag{18}$$

The Kalman gain is then computed from these covariance matrices (19). We calculate the *a posteriori* estimate m_n in (20) as a combination of the *a priori* estimate of a mean of the state vector and a weighted difference between the measurement result S_n and its *a priori* prediction. The *a posteriori* estimate of the error covariance matrix is updated using (21)

$$\mathcal{K}_n = (\Xi_n / \mathcal{Z}_n^T)/\mathcal{Z}_n \tag{19}$$

$$m_n = m_n^- + \mathcal{k}_n[S_n - \mu_n^-] \tag{20}$$

$$\mathcal{A}_n = \text{cholupdate}(\mathcal{A}_n^-,\ \mathcal{k}_n\mathcal{Z}_n, -1) \tag{21}$$

where / denotes a back substitution operation as a superior alternative to the matrix inversion. Obtained values of m_n and

$\mathcal{A}_n$ become the input of the successive prediction–correction loop.

IV. Reducing Computational Complexity

We introduce two techniques that significantly reduce the computational complexity of our model. One of the techniques includes techniques for fast numerical convergence, whereas the other provides fast and accurate MOR technique of dynamic IC thermal network.

The ODE in (3) needs to be numerically integrated in time as analytical solutions are not possible in general. Although many time marching numerical methods for solving ODEs are based on methods that do not require explicit differentiation, these methods are conceptually based on repeated Taylor series expansions around increasing time instants. Revisiting these roots and basing time marching on Taylor series expansion allows element-by-element time step adaptation by supporting the extrapolation of temperatures at arbitrary times.

The MOR enables us to find a low-dimensional but accurate approximation of the thermal network (10), which preserves the input–output behavior to a desired extent. In an asymptotic waveform evaluation (AWE) algorithm [36] explicit moment matching was used to compute the dominant poles via Padé approximation. As the AWE method is numerically unstable for higher order moment approximation, a more efficient solution of the numerical problem of AWE is to use a projection-based Krylov subspace MOR methods, such as the Padé via Lanczos method [37], or PRIMA [38]. These methods are, however, not efficient for circuits with many inputs and output terminals as the reducing costs are tied to the number of terminals; the number of poles of reduced models is also proportional to the number of terminals. In addition, PRIMA-like methods do not preserve structure properties like reciprocity of a network. Alternatively, MOR can be performed by means of singular value decomposition (SVD)-based approaches such as control-theory-based truncated balance realization (TBR) methods, where the weakly uncontrollable and unobservable state variables are truncated to achieve the reduced models [39]–[45]. The major advantage of SVD-based approaches over Krylov subspace methods lies in their ability to ensure the errors satisfy an *a priori* upper bound [43]. Furthermore, SVD-based methods typically lead to optimal or near optimal reduction results as the errors are controlled in a global way.

A. Modified Runge–Kutta Solver

We first designate numerical dissipation and boundary condition terms and treat them separately. We adopt a more stable semi-implicit treatment of the numerical dissipation terms, which is formally correct for the Crank–Nicolson scheme, but implies a modification of dissipation terms in (3) for the Runge–Kutta scheme. Using a discontinuity detector [46], the modified third-order Runge–Kutta predictor–corrector scheme, the spatially discrete system in (3) reads

$$\begin{aligned}(1+\Delta t\Omega(T_n))T_{(1)} &= T_n + \Delta t\Lambda(T_n)\\ (1+1/4\Delta t\Omega(T_{(1)}))T_{(2)} &= 1/4(3T_n + T_{(1)} + \Delta t\Lambda(T_{(1)}))\\ (1+2/3\Delta t\Omega(T_{(2)}))T_{(n+1)} &= 1/3(T_n + 2T_{(2)} + 2\Delta t\Lambda(T_{(2)}))\end{aligned} \tag{22}$$

where $\Lambda = C^{-1}G$ and $\Omega = C^{-1}P$ for two time instants T_n and T_{n+1}. Note that the terms designating boundary conditions are treated separately. To achieve fast convergence, the coefficients in the Runge–Kutta scheme have been optimized to damp the transients in the pseudotime integration as quickly as possible and to allow large pseudotime steps. In addition, the use of a point implicit Runge–Kutta scheme ensures that the integration method is stable. Convergence to steady state is further accelerated using a multigrid technique, e.g., the original fine mesh is coarsened a number of times and the solution on the coarse meshes is used to accelerate convergence to steady state on the fine mesh.

The boundary conditions in (5) also have to be written in terms of the discrete (in space and time) temperature. For the time marching between time indexes T_n and T_{n+1}, the form of the right-hand side depends, among other things, on the time-marching scheme chosen. The terms involved in the surface integral involve temperature and the spatial derivatives of temperature on the surfaces. We approximate these terms using the nearest neighbor temperatures only. Hence, the discrete form of the surface integral is of the form of a linear combination of the temperature at the center of the cell and the temperature at the center of the neighboring cells. The modified implicit Runge–Kutta scheme can not be used to compute neighbor temperatures at boundary condition, as it results in circular dependency problems. More specifically, T_n must be known before T_i is computed. Similarly, T_n depends on T_i. To solve this problem, we use the forward Euler method to extrapolate T_n. In addition, to increase efficiency, we employ backward Euler ($\theta = 1$, where the free parameter θ is used to control accuracy and stability of the scheme) and factor the matrix P_Q before the time stepping starts and then use forward and backward substitution in each time step

$$\begin{aligned}&\theta[P_{C,j}]_{n+1} + (1-\theta)[P_{C,j}]_{n+1}\\ &\qquad = \sum_{i=\mathcal{N}_f+1}^{\mathcal{N}} \Big[\int_{S_c} \mathcal{N}_j c_V \mathcal{N}_i^T \Delta z dS\Big]((T_{i|n+1} - T_{i|n})/\Delta t)\\ &\theta[P_{G,j}]_{n+1} + (1-\theta)[P_{G,j}]_{n+1}\\ &= \sum_{i=\mathcal{N}_f+1}^{\mathcal{N}} \Big[\int_{S_c} (\nabla\mathcal{N}_j) h_j g (\nabla\mathcal{N}_i)^T \Delta z dS\Big](\theta T_{i|n+1} + (1-\theta)T_{i|n})\end{aligned} \tag{23}$$

where we approximate the prescribed temperature rate rather than use its exact value.

B. Adjusted Balanced Stochastic Truncation

In this paper, we introduce a balanced stochastic truncation [47] in MOR of thermal networks to provide a uniform approximation of the frequency response for the original system over the whole frequency domain and to preserve phase information. The approach presented here produces orthogonal basis sets for the dominant singular subspace of the controllability and observability Gramians, which significantly reduces the complexity and computational costs of SVD, while preserving MOR accuracy and the quality of the approximations of the TBR procedure.

The balanced truncation [39] involves the explicit balancing of (10). This procedure is dangerous from a numerical point of view because the balancing transformation matrix Ψ tends to be highly ill conditioned. The square root method [45] is an attempt to cope with this problem by avoiding explicit balancing of the system. The method is based on the Cholesky factors of the Gramians instead of the Gramians themselves. In [48], the use of the Hammarling method was proposed to compute these factors. Recently, in [40] and [44], it has been observed that the solutions often have low numerical rank, which means that there is a rapid decay in the eigenvalues of the Gramians. Moreover, approximating directly the Cholesky factors of the Gramians and using these approximations to provide a reduced model, has a comparable cost to that of the popular moment-matching methods. It requires only matrix–vector products and linear solvers. For large systems with a structured transition matrix, this method is an attractive alternative because the Hammarling method can generally not benefit from such structures. In the original implementation, this step is the computation of exact Cholesky factors, which may have full rank.

To guarantee the passivity of the reduced model and simplify the computational procedure, we first convert original descriptor systems into standard state-space equations by mapping $\mathcal{C} \rightarrow I$, $\mathcal{G} \rightarrow \mathcal{C}^{-1}\mathcal{G}$, and $\mathcal{B} \rightarrow \mathcal{C}^{-1}\mathcal{B}$. If we define $\Phi(s) = \Gamma(s)\Gamma^T(-s)$, and let W be a square minimum spectral factor of Φ, satisfying $\Phi(s) = W^T(-s)W(s)$, a state-space realization $(\mathcal{G}_W, \mathcal{B}_W, \mathcal{L}_W)$ of $W(s)$ can be obtained as

$$\mathcal{G}_W = \mathcal{G}\ \mathcal{B}_W = \mathcal{B} + Y\mathcal{L} \quad \mathcal{L}_W^T = \mathcal{L}^T - \mathcal{B}_W^T X \tag{24}$$

where Y is the controllability Gramian (e.g., the low-rank approximation to the solution) of Γ given by Lyapunov equation

$$\mathcal{G}Y + Y\mathcal{G}^T + \mathcal{B}\mathcal{B}^T = 0 \tag{25}$$

and X is the observability Gramian of W, being the solution of the Riccati equation

$$X\mathcal{G} + \mathcal{G}^T X + \mathcal{L}F\mathcal{L}^T + X\mathcal{B}_W M^{-1}\mathcal{B}_W^T X = 0 \tag{26}$$

where $F \in \mathcal{R}^{p\times p}$ is symmetric positive semidefinite and $M \in \mathcal{R}^{m\times m}$ is symmetric positive definite. In the iterative procedure, we approximate the low-rank Cholesky factors Ξ and Θ, such that $\Theta^T\Theta \approx X$ and $\Xi^T\Xi \approx Y$. We obtain the observability Gramian X by solving the Riccati equation (26) with a Newton double step iteration

$$\begin{aligned}(\mathcal{G}^T - Z^{(z-1)}\mathcal{B}_W^T)X^{(z)} + X^{(k)}(\mathcal{G} - \mathcal{B}_W Z^{(z-1)^T}) \\ = -\mathcal{L}^T F\mathcal{L} - Z^{(z-1)}MFZ^{(z-1)^T} \\ Z^{(z)} = X^{(z)}\mathcal{B}_W M^{-1}\end{aligned} \tag{27}$$

where the feedback matrix $Z = X\mathcal{B}_W M^{-1}$, for $z = 1, 2, 3,\ldots$, which generates a sequence of iterates $X^{(z)}$. This sequence converges toward the stabilizing solution X if the initial feedback Z_0 is stabilizing, i.e., $G - BZ^{(0)T}$ is stable. If we partition Ψ and Ψ^{-1} as $\Psi = [\mathcal{J}U]$ and $\Psi^{-1} = [OV]^{-1}$ then $I_l = O\mathcal{J}$ is the identity matrix, $\Pi = \mathcal{J}O$ is a projection matrix, and O and $\mathcal{J}$ are the truncation matrices. In the related balancing model reduction methods, the truncation matrices

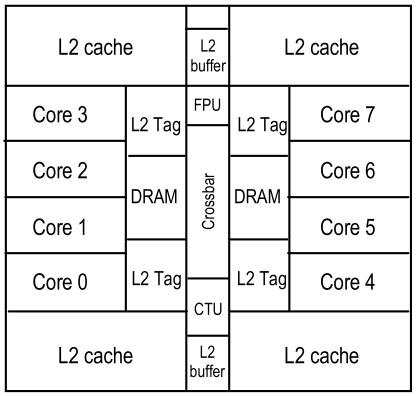

Fig. 5. UltraSparc T1 architecture floorplan.

O and $\mathcal{J}$ can be determined knowing only the Cholesky factors of the Gramians Y and X. If we let $\Xi^T\Theta = U\Sigma V^T$, where $\Sigma = \text{diag}(\sigma_1,\ldots,\sigma_l)$, be SVD of $\Xi^T\Theta$, then we can calculate the truncation matrices $O = \Sigma^{-1/2}V^T\Theta$ and $\mathcal{J} = \Xi^T U\Sigma^{1/2}$. Under a similarity transformation of the state-space model, both parts can be treated simultaneously after a transformation of the system $(\widehat{C}, \widehat{\mathcal{G}}, \widehat{\mathcal{B}}, \widehat{\mathcal{L}}^T)$ with a nonsingular matrix $\Psi \in \mathcal{R}^{m\times m}$ into a stochastically balanced system

$$\widehat{C} = \mathcal{J}^T C O\ \widehat{\mathcal{G}} = \mathcal{J}^T\mathcal{G}O\ \widehat{\mathcal{B}} = \mathcal{J}^T\mathcal{B}\ \widehat{\mathcal{L}} = \mathcal{L}O \tag{28}$$

where $\widehat{C}, \widehat{\mathcal{G}} \in \mathcal{R}^{l\times l}$, $\widehat{\mathcal{B}} \in \mathcal{R}^{l\times p}$, and $\widehat{\mathcal{L}} \in \mathcal{R}^{p\times l}$ are of the order l much smaller than the original order m, if controllability Gramian Y satisfies $\Psi^{-1}Y\Psi^{-T} = \Psi^T X\Psi$. SVDs are arranged so that the diagonal matrix containing the singular values has the same dimensions as the factorized matrix and the singular values appear in nonincreasing order.

V. Experimental Results

A. Experimental Setup

The chip architecture determines the complexity of processing versus storage versus communication elements and thus the thermal peak of these elements. A chip with complex PEs (e.g., wide issue and multithreaded) will require larger storage elements (e.g., large multilevel caches and register files) as well as sophisticated communication elements (e.g., multilevel wide buses, networks with wide link channels, deeply pipelined routers, and significant router buffering). On the other extreme, there are chip architectures where PEs are single ALUs serviced by a few registers at ALU input/output ports, interconnected with simple single-stage routers with a little buffering. Application characteristics dictate how these elements are used, and hence influencing the thermal profile of the chip. In this paper, as a platform for analyzing the absolute and relative thermal impact of all the components of a chip, we use a two-die stack consisting of 300-μm-thick dies with a 30 mm $\times$ 10 mm cross section and an architecture resembling UltraSparc T1 architecture [49] (Fig. 5), stacked together through a thermally resistive interface material. Tiles are interconnected through a wormhole routed 3-D mesh network consisting of seven-port routers with two TSV-based vertical links. Alongside enabling stacking, the use of a 3-D mesh

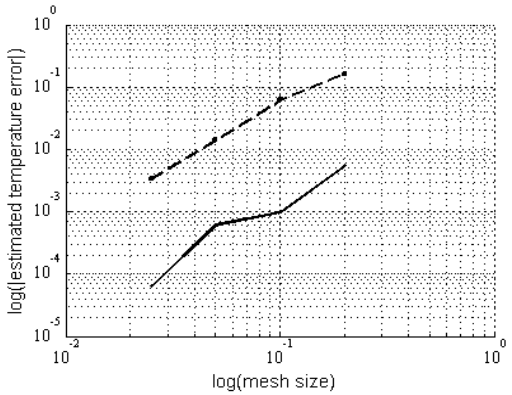

Fig. 6. Temperature error versus mesh size for the proposed (bold line) and generalized finite-element method (dashed line).

results in lower end-to-end packet latencies when compared with planar meshes with the same number of nodes and under identical traffic conditions. The experiments were executed on a 64-b Linux server with two quadcore Intel Xeon 2.5-GHz CPUs and 16-GB main memory. Values regarding thermal resistance, silicon thickness, and copper layer thickness have been derived from [49] and its floorplan and power/area distribution ratio of each element from [50], respectively.

Basic Math application from the MiBench benchmark [51] is selected and run on datasets provided by [52]. Switching activities were obtained using SimpleScalar [53]. The calculation was performed in a numerical computing environment [54]. Thermal profile has been estimated as in [21]. Thermal conductance matrix is generated for time equal to temperature check cycle, which improves effective utilization of instantaneous temperature margin [24]. The power is dissipated in each die in hotspots of variable extension (minimum size $= 100$ μm in this paper), while the structure is thermally isolated on the sides. Heat sink and package thermal resistances are assumed to be 2 and 20 K/W, respectively. Thermal conductivity of silicon is taken to be 148 W/(mK) and that of copper interconnect 383 W/(mK). In comparison with the heat sink and package resistances, the silicon resistance is $\sim$0.02 K/W.

B. Reducing Computational Complexity

For thermal profile comparison purposes [21], we implemented generalized finite-element method, which can be found in several commercially available software packages (e.g., Hotspot [55] and ANSYS [56]). The accuracy of a discretization concerns the rate of convergence as function of mesh size. The truncation error consists of the discretization applied to the exact solution. Fig. 6 shows that the numerical accuracy of the proposed Galerkin method with l-adaptive error control is one to two orders of magnitude more accurate for comparable mesh size than corresponding generalized finite-element method. Furthermore, we compared modified Runge–Kutta solver with Euler (as in Hotspot [55]) and Newmark (in ANSYS [56]). As shown in Table I, the proposed method offers increased accuracy, while simultaneously increases solution efficiency. Theoretically, the modified

TABLE I
ACCURACY COMPARISON AT TOP SURFACE

x-y[mm]	Euler [55]		Newmark [56]		**This Paper**	
	CPU	Err.%	CPU	Err.%	CPU	Err.%
1.17-3.42	1.76s	2.364	2.43s	0.375	2.04s	0.042
2.42-4.16	1.93s	2.147	2.54s	0.463	2.08s	0.028
3.86-4.28	1.86s	2.267	2.47s	0.428	2.29s	0.034
5.28-3.76	2.13s	2.325	2.63s	0.364	2.58s	0.041
6.68-4.54	1.87s	2.134	2.38s	0.448	2.86s	0.043
8.14-4.18	1.94s	2.246	2.45s	0.564	2.72s	0.037
9.64-3.86	1.98s	2.185	2.52s	0.474	2.47s	0.043

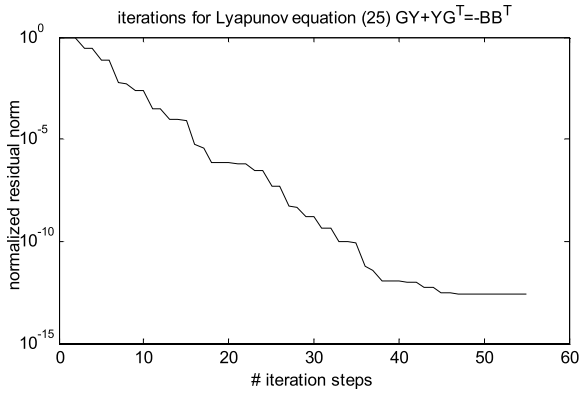

Fig. 7. Convergence history of residual form. Convergence is obtained after 46 iterations.

third-order Runge–Kutta scheme can reach accuracy of $O(\Delta_t^4)$ [46]. On the other hand, the accuracy of Euler method is $O(\Delta_t^2)$. The errors in Euler scheme are dominated by the deterministic terms as long as the step size is large enough. In more detail, the error of the method behaves like $O(\alpha^2 + \varepsilon\alpha + \varepsilon^2\alpha^{1/2})$, when ε is used to measure the smallness of the temperature and α is the time step. The smallness of the temperature also allows special estimates of the local error terms, which can be used to control the step size. An efficient implementation of the Newmark methods for linear problems requires that direct methods (e.g., Gauss elimination) be used for the solution in the system of algebraic equations. When a step size should be updated, the prediction of the new step size has to be made such that the prescribed accuracy can be achieved with the least cost. The rate of convergence of the global error in the Newmark integration can be $O(\Delta_t^2)$. Correspondingly, the rate of convergence of the local error should achieve $O(\Delta_t^3)$. Suppose that the current time step is α, then we have $O(\kappa\alpha^3)$, where κ is a constant depending on the exact solution.

Using the balanced stochastic truncation MOR technique for indirect sensing, we obtain low-dimensional but accurate approximation of the thermal network (11). The convergence history for solving the Lyapunov equation (25) with respect to the number of iteration steps is plotted in Fig. 7. Convergence is obtained after 46 iterations. The total CPU time needed to solve the Lyapunov equation according to the related tolerance for solving the shifted systems is 0.27 s. Note further that saving iteration steps means that we save large amounts of memory especially in multiple input and output systems where

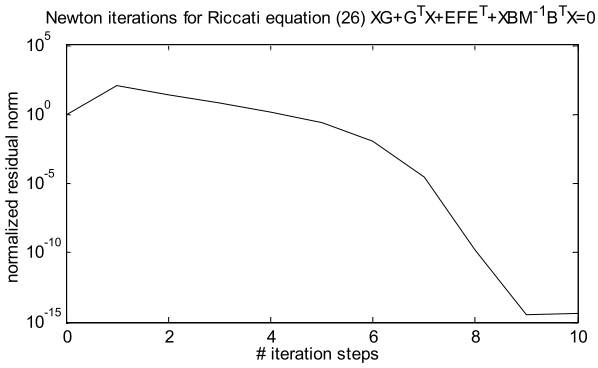

Fig. 8. Convergence history of the normalized residual form of the Newton double step iteration (27) for solving the Riccati equation (26).

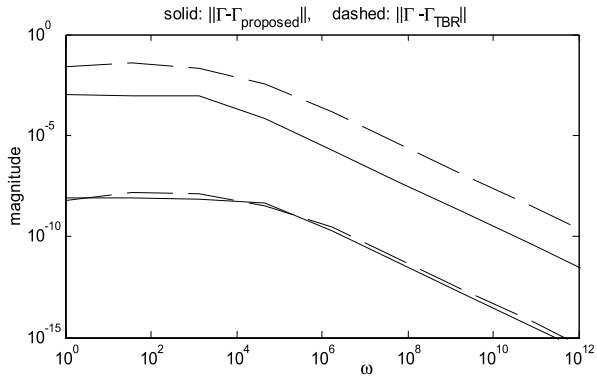

Fig. 9. Bode magnitude plot of the approximation errors.

the factors are growing by p columns in every iteration step. The convergence history of the Newton double step iteration (27) for solving the Riccati equation (26) is shown in Fig. 8. Because of symmetry, the matrices F and M can be factored by a Cholesky factorization. Hence, the equations to be solved in (27) have a Lyapunov structure similar to (25). In this algorithm, the (approximate) solution of the Riccati equation is provided as a low-rank Cholesky factor product [57] rather than an explicit dense matrix. The algorithm requires much less computation compared with the standard implementation, where Lyapunov is solved directly by the Bartels–Stewart or the Hammarling method.

The CPU time needed to solve the Riccati equation inside the iteration is 0.77 s. Fig. 9 shows a comparison with the TBR method [40]. When very accurate Gramians are selected, the approximation error of the reduced system is very small compared with the Bode magnitude function of the original system. The lower two curves correspond to the highly accurate reduced system; the proposed MOR technique delivers a system of lower order. For the lower curve, the CPU time of the proposed method is 11.47 s versus 19.64 s for the TBR method. The upper two denote $k = 15$ reduced orders; the proposed technique delivers two orders of magnitude better accuracy. The reduced order is chosen in dependence of the descending ordered singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$, where r is the rank of factors that approximate the system Gramians.

For m variation sources and l reduced parameter sets, the full parameter model requires $O(m^2)$ simulation samples and thus has a $O(m^6)$ fitting cost. On the other hand, the proposed parameter reduction technique has a main computational cost attributable to the $O(m + l^2)$ simulations for sample data collection and $O(l^6)$ fitting cost significantly reducing the required sample size and the fitting cost. The CPU time of the proposed method for $k = 15$ reduced order is 8.35 s. The TBR method requires 14.64-s CPU time.

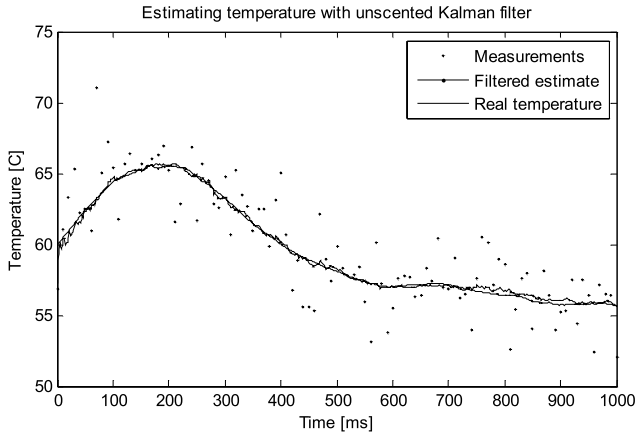

Fig. 10. Sensor measurements, actual, and estimated temperatures.

C. Temperature Tracking

In the experiments, the temperature values of the grid cell containing the sensors are observable, whereas the temperature at other grid cells are estimated with proposed UKF. We assumed 16×16 chip gridding granularity. Furthermore, for thermal tracking, we assumed that the sensors are uniformly scattered on the chip. The number of samples and the sample locations is varied. No specific sensor technology is assumed in this paper. The readings from the temperature sensors initiate the estimation algorithm. The transformation matrix $h(\cdot)$ in (11) is determined by the sensor placement. Gaussian noise is superimposed on the actual temperature values to model the inaccuracies of real thermal sensors, such as supply voltage fluctuation, fabrication variability, cross coupling, and so on. Processes generating these noises are assumed to be stationary between different successive prediction–correction steps. Actual temperatures at the sensor locations and locations of interest are obtained with the proposed Galerkin method and acquired results compared with HotSpot [55] and ANSYS [56], as in Fig. 6. In this sense, the measurement error designates the temperature difference between sensors readings and real temperature at locations of interest in the observed grid cell.

We compare the accuracy of our approach with that of the KF [14] and EKF [16]. In KF dynamic model, function $f(\cdot)$ in (11) is a linear Gaussian model. Such a model does not account for the nonlinear temperature-circuit parameters dependency and, therefore its usability in the practical applications is restricted. Furthermore, because of the inaccuracy of its linear model, the standard KF relies excessively on the accuracy of sensor input. The temperature estimates derived from the KF are nonanticipative in the sense that they are only conditional to sensor measurements obtained before and at the time step n.

TABLE II
ERROR STATISTICS FOR LIMITED NUMBER OF SENSORS

# Sensors	Measurement Errors (°C)		KF [14] Estimation Errors (°C)		EKF [16] Estimation Errors (°C)		**UKF** Estimation Errors (°C)	
	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)
2	5.41	5.89	3.16	4.17	1.82	3.01	0.42	0.68
3	5.09	5.27	3.15	3.92	1.71	2.92	0.34	0.72
4	4.34	5.17	3.65	3.26	1.78	2.14	0.45	1.04
5	4.16	4.61	3.27	3.89	1.51	1.89	0.23	0.76
6	3.76	4.23	2.87	4.06	1.87	2.31	0.41	0.63

However, after we have obtained measurements, we could compute temperature estimates of T_{n-1}, T_{n-2},..., which are also conditional to the measurements after the corresponding state time steps. With the Rauch–Tung–Striebel smoother, more measurements and information are available for the estimator. Therefore, these temperature estimates are more accurate than the nonanticipative measurements computed by the KF. The EKF approximates the nonlinearities with linear or quadratic functions or explicitly approximate the filtering distributions by Gaussian distributions. In UKF, the UT is used for approximating the evolution of Gaussian distribution in nonlinear transforms. Fig. 10 shows that the proposed method always keeps track of the actual temperature with high accuracy for a randomly chosen chip location that does not coincide with the sensor location. For clarity, we only depicted UKF tracking. There is no observable difference between the reduced and original model results, which suggests high accuracy of MOR. With (11), we simulated the thermal profile of the test processor for a total duration of 600 s (the simulation starts at room temperature). This is assumed to be the real chip temperature and is used to measure estimation accuracy. We examine the mean absolute error and the standard deviation of the error as the location of interest. These values are averaged over all the locations of interest.

High precision of temperature tracking (within 0.5 °C for mean and 1.1 °C for standard deviation) for various cases, ranging from two to six sensors, respectively, placed at an arbitrary location around the hotspot, is shown in Table II. In ICs, the placement of the sensors is constrained to areas where there is enough spatial slack due to the limitations such as additional channels for routing and input/output. For thermal sensors, if one sensor per router is not affordable for large on-chip networks, the network can be partitioned into regions and multiple adjacent routers within a region could share the same sensor. The proposed technique is able to estimate the temperature at the locations far away from the limited number of sensors. As anticipated, the Kalman techniques are relatively independent of the relative position of the sensor and the location of interest. The UKF obtain almost identical accuracy (variations of <0.3 °C) across the examined range significantly outperforming KF and EKF, especially when the number of sensors is small. This difference is shown in Fig. 11. Note that 1 °C accuracy translates to 2-W power savings [58]. The state vector representing temperatures at different grid cells at time n in (11) and function $f(\cdot)$ is determined by the circuit parameters and the chosen length of time steps.

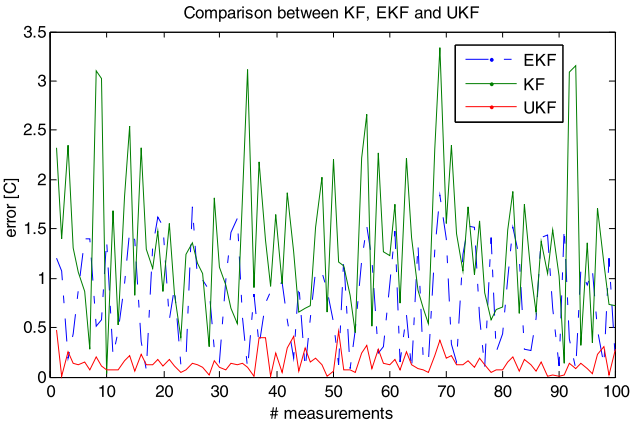

Fig. 11. Error comparison between KF, EKF, and UKF.

Table III shows statistics of the measurement and estimation errors for different sizes of time steps. The chosen time step is at 10^{-4} s and multiplied by powers of two. Thermal profile transition in 3-D ICs is a very slow process and a noticeable temperature variation takes at least several hundred milliseconds to change; accordingly, a few millisecond overheads for reading noisy thermal sensors will not impact the effectiveness of dynamic thermal management unit. High precision within 1.1 °C for both mean and standard deviation is obtained even with a large time step size. The average error in Table IV (across all the chip locations) of each method is reported as we vary the sensor noise level as defined in (11). As we increase the noise level, the estimation accuracy generated by KF and EKF degrades more rapidly in contrast to UKF, which generates accurate thermal estimates (within 0.8 °C) under all the examined circumstances. The improved performance of the UKF compared with the EKF is due to two factors, namely the increased time-update accuracy and the improved covariance accuracy. In the UKF case, the covariance estimation is very accurate, which results in a different Kalman gains in the measurement-calibration equation and hence the efficiency of the measurement-calibration step. The advantage of EKF over UKF is its relative simplicity compared with its performance. Nevertheless, because EKF is based on a local linear approximation, its accuracy is limited in highly nonlinear systems. In addition, the filtering model is restricted in the sense that only Gaussian noise processes are allowed and thus the model cannot contain, for example, discrete-valued random variables.

TABLE III
ERROR STATISTICS FOR DIFFERENT TIME STEPS

Step Size (10^{-4}s)	Measurement Errors (ºC)		KF [14] Estimation Errors (ºC)		EKF [16] Estimation Errors (ºC)		**UKF** Estimation Errors (ºC)	
	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)
128	5.12	6.31	2.94	4.15	1.18	1.45	0.47	0.78
512	5.24	6.23	3.55	4.76	1.24	1.62	0.64	0.74
1028	5.41	6.16	4.65	5.01	1.73	1.94	0.86	0.96
2056	6.04	7.12	5.77	6.34	2.07	2.25	1.12	0.98

TABLE IV
ERROR STATISTICS FOR DIFFERENT NOISE SETTINGS

Sensor noise (%)	Measurement Errors (ºC)		KF [14] Estimation Errors (ºC)		EKF [16] Estimation Errors (ºC)		**UKF** Estimation Errors (ºC)	
	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)	Error (μ)	Error (σ)
2.5	5.27	6.24	3.23	4.46	1.36	2.65	0.36	0.65
5.0	6.36	6.85	4.15	5.13	2.16	3.22	0.42	0.68
7.5	8.12	6.45	6.78	6.93	2.37	3.56	0.56	0.76
10.0	9.32	9.86	7.14	8.44	2.86	3.45	0.54	0.84

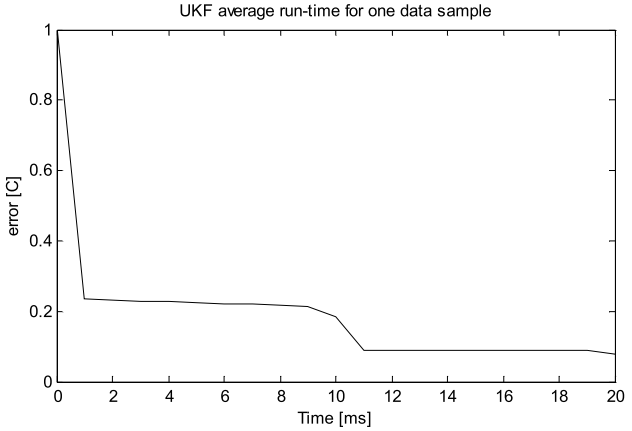

Fig. 12. Runtime overhead of the UKF recursive regression.

The Gaussian restriction also prevents handling of hierarchical models or other models, where significantly non-Gaussian distribution models would be needed. The EKF also formally requires the measurement model and dynamic model functions to be differentiable. Even when the Jacobian matrices exist and could be computed, the actual computation and programming of Jacobian matrices are error prone and hard to debug. On the other hand, UKF is not based on local linear approximation; UKF uses a bit further points in approximating the nonlinearity.

The computational load increases when moving from the EKF to the UKF if the Jacobians are computed analytically [the average runtime of EKF versus UKF (Fig. 12) is ∼16 and 19 ms for one measurement, respectively]. However, for higher order systems, the Jacobians for the EKF are computed using finite differences. In this case, the computational load for the UKF is comparable with the EKF. Effectively, the EKF builds up an approximation to the expected Hessian by taking outer products of the gradient. The UKF, however, provides a more accurate estimate through direct approximation of the expectation of the Hessian. Another distinct advantage of the UKF occurs when either the architecture or error metric is such that differentiation with respect to the parameters is not easily derived as necessary in the EKF. The UKF effectively evaluates both the Jacobian and Hessian precisely through its sigma point propagation, without the need to perform any analytic differentiation.

VI. CONCLUSION

Accurate temperature estimation is one of the foremost steps in the evaluation of successful high-performance 3-D IC designs. However, because of the temperature sensors power/area overheads and the limitations such as additional channels for routing and input/output, their number and placement are highly constrained to areas where there is enough spatial slack. Therefore, the problem of tracking the entire thermal profile based on only a few limited sensor observations is rather complex. This problem is further aggravated due to unpredictability of workloads and fabrication/environmental variabilities. Within this framework, in this paper, to improve thermal management efficiency, we present methodology based on UKF for accurate temperature estimation at all the chip locations while simultaneously countering sensor noise. As the results show, the proposed method generates accurate thermal estimates (within 1.1 °C) under all the examined circumstances. In comparison with KF and EKF, the UKF consistently achieves a better level of accuracy at limited costs. In addition, to provide significant reductions on the required simulation samples for constructing accurate models, we introduce a balanced stochastic truncation MOR. The proposed approach produces orthogonal basis sets for the dominant singular subspace of the controllability and observability Gramians, which exploits lowrank matrices and avoids large-scale matrix factorizations, significantly reducing the complexity and computational costs of Lyapunov and Riccati equations, while preserving MOR accuracy and the quality of the approximations of the TBR procedure.

ACKNOWLEDGMENT

The authors acknowledge the contributions of M. Berkelaar and S. S. Kumar from the Delft University of Technology, Delft, The Netherlands, A. Aggarwal of ASLM Holdings and R. S. Jagtap of ARM Holdings.

REFERENCES

[1] A. W. Topol, D. C. L. Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Ieong, "Three-dimensional integrated circuits," *IBM J. Res. Develop.*, vol. 50, nos. 4–5, pp. 491–506, Jul. 2006.

[2] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. P. Zhang, K. Bazargan, and S. Sapatnekar, "Placement and routing in 3D integrated circuits," *IEEE Design Test Comput.*, vol. 22, no. 6, pp. 520–531, Nov./Dec. 2005.

[3] S. Im and K. Banerjee, "Full chip thermal analysis of planar (2D) and vertically integrated (3D) high performance ICs," in *Proc. IEEE IEDM*, Jul. 2000, pp. 727–730.

[4] J. Torresola, C. Chia-pin G. Chrysler, D. Grannes, R. Mahajan, R. Prasher, and A. Watwe, "Density factor approach to representing impact of die power maps on thermal management," *IEEE Trans. Adv. Packag.*, vol. 28, no. 4, pp. 659–664, Nov. 2005.

[5] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. IEEE Int. Conf. Comput.-Aided Design*, Nov. 2004, pp. 306–313.

[6] T.-Y. Chiang, S. J. Souri, C. O. Choi, and K. C. Saraswat, "Thermal analysis of heterogeneous 3D ICs with various integration schemes," in *Proc. IEEE IEDM*, Dec. 2001, pp. 681–684.

[7] T. T. Wang, Y. M. Lee, and C. C. P. Chen, "3D thermal ADI—An efficient chip-level transient thermal simulator," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1434–1445, Dec. 2002.

[8] K. J. Scott, "Electrostatic potential Green's functions for multi-layered dielectric media," *Philips J. Res.*, vol. 45, no. 3, pp. 293–324, 1990.

[9] A. Vincenzi, A. Sridhar, M. Ruggiero, and D. Atienze, "Fast thermal simulation of 2D/3D integrated circuits exploiting neural networks and GPUs," in *Proc. IEEE ISLPED*, Aug. 2011, pp. 151–156.

[10] A. M. Niknejad, R. Gharpurey, and R. G. Meyer, "Numerically stable Green function for modeling and analysis of substrate coupling in integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 4, pp. 305–315, Apr. 1998.

[11] B. Wang and P. Mazumder, "Fast thermal analysis for VLSI circuits via semi-analytical Green's function in multi-layer materials," in *Proc. IEEE ISCAS*, vol. 2. May 2004, pp. 409–412.

[12] N. Allec, Z. Hassan, L. Shang, R. P. Dick, and R. Yang, "ThermalScope: Multi-scale thermal analysis for nanometer-scale integrated circuits," in *Proc. IEEE ICCAD*, Nov. 2008, pp. 603–610.

[13] A.M. Ionescu, G. Reimbold, and F. Mondon, "Current trends in the electrical characterization of low-k dielectrics," in *Proc. IEEE Int. Semicond. Conf.*, Oct. 1999, pp. 27–36.

[14] Y. Zhang, A. Srivastava, and M. Zahran, "Chip level thermal profile estimation using on-chip temperature sensors," in *Proc. IEEE ICCD*, Oct. 2008, pp. 432–437.

[15] R. Cochran and S. Reda, "Spectral techniques for high resolution thermal characterization with limited sensor data," in *Proc. 46th IEEE DAC*, Jul. 2009, pp. 478–483.

[16] S. Sharifi, C.-C. Liu, and T. S. Rosing, "Accurate temperature estimation for efficient thermal management," in *Proc. 9th IEEE ISQED*, Mar. 2008, pp. 137–142.

[17] Y. Zhang, A. Srivastava, and M. Zahran, "Chip level thermal profile estimation using on-chip temperature sensors," in *Proc. IEEE Int. Conf. Comput. Design*, Jan. 2008, pp. 1065–1068.

[18] H. Jung and M. Pedram, "A stochastic local hot spot alerting technique," in *Proc. IEEE Asia South Pacific Design Autom. Conf.*, Apr. 2008, pp. 468–473.

[19] S. Sharifi and T. S. Rosing, "Accurate direct and indirect on-chip temperature sensing for efficient dynamic thermal management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 10, pp. 1586–1599, Oct. 2010.

[20] Y. Zhang and A. Srivastava, "Adaptive and autonomous thermal tracking for high performance computing systems," in *Proc. IEEE DAC*, Jun. 2010, pp. 68–73.

[21] A. Zjajo, N. van der Meijs, and R. van Leuken, "Thermal analysis of 3D integrated circuits based on discontinuous Galerkin finite element method," in *Proc. IEEE 13th ISQED*, Mar. 2012, pp. 117–122.

[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," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[23] Y. Xie, "Processor architecture design using 3D integration technology," in *Proc. IEEE 3rd Int. Conf. VLSI Design*, Jan. 2010, pp. 446–451.

[24] A. Aggarwal, S. S. Kumar, A. Zjajo, and R. van Leuken, "Temperature constrained power management scheme for 3D MPSoC," in *Proc. IEEE Int. Workshop Signal Power Integr.*, Dec. 2012, pp. 7–10.

[25] S. S. Kumar, A. Aggarwal, R. Jagtap, A. Zjajo, and R. van Leuken, "A system level methodology for interconnect aware and temperature constrained power management of 3D MP-SOCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2013.2273003.

[26] J. Lienhard, *A Heat Transfer Textbook*. Cambridge, MA, USA: Phlogiston, 2006.

[27] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 8, pp. 1479–1492, Aug. 2008.

[28] S. O. Memik, R. Mukherjee, M. Ni, and J. Long, "Optimizing thermal sensor allocation for microprocessors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 3, pp. 516–527, Mar. 2008.

[29] B.-H. Lee and T. Kim, "Optimal allocation and placement of thermal sensors for reconfigurable systems and its practical extension," in *Proc. IEEE Asia South Pacific Design Autom. Conf.*, Mar. 2008, pp. 703–707.

[30] F. Liu, "A general framework for spatial correlation modeling in VLSI design," in *Proc. IEEE DAC*, Jun. 2007, pp. 817–822.

[31] N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, "Leakage current: Moore's law meets static power," *IEEE Comput.*, vol. 36, no. 12, pp. 68–75, Dec. 2003.

[32] L. He, W. Liao, and M. Stan, "System level leakage reduction considering the interdependence of temperature and leakage," in *Proc. 41st Annu. IEEE/ACM DAC*, Jun. 2004, pp. 12–17.

[33] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no.3, pp. 401–422, Mar. 2004.

[34] R. van der Merwe and E.A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Proc. IEEE ICASSP*, vol. 6. May 2001, pp. 3461–3464.

[35] G. Golub and C. van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.

[36] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.

[37] P. Feldmann and R.W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995.

[38] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

[39] B. C. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Autom. Control*, vol. 26, no. 1, pp. 17–31, Feb. 1981.

[40] J. Li and J. White, "Efficient model reduction of interconnect via approximate system Grammians," in *Proc. IEEE 36th Int. Conf. Comput. Aided Design*, Nov. 1999, pp. 380–384.

[41] J. R. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proc. IEEE DAC*, Jun. 2002, pp. 52–57.

[42] J. R. Phillips and L. M. Silveira, "Poor man's TBR: A simple model reduction scheme," in *Proc. IEEE Design, Autom. Test Eur. Conf.*, Jun. 2004, pp. 938–943.

[43] N. Wong and V. Balakrishnan, "Fast positive-real balanced truncation via quadratic alternating direction implicit iteration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1725–1731, Sep. 2007.

[44] T. Penzl, "A cyclic low-rank Smith method for large sparse Lyapunov equations," *SIAM J. Sci. Comput.*, vol. 21, no. 4, pp. 1401–1418, 2000.

[45] M. G. Safonov and R. Y. Chiang, "A Schur method for balanced-truncation model reduction," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 729–733, Jul. 1989.

[46] L. Krivodonova, "Limiters for high-order discontinuous Galerkin methods," *J. Comput. Phys.*, vol. 226, no. 1, pp. 879–896, Sep. 2007.

[47] M. Green, "Balanced stochastic realizations," *Linear Algebra Appl.*, vol. 98, pp. 211–247, Jan. 1988.

[48] M. S. Tombs and I. Postlethwaite, "Truncated balanced realization of stable, non-minimal state-space systems," *Int. J. Control*, vol. 46, no. 4, pp. 1319–1330, 1987.
[49] A. Leon, K. Tam, J. Shin, D. Weisner, and F. Schumacher, "A power efficient high-throughput 32-thread SPARC processor," in *Proc. IEEE ISSCC*, Feb. 2006, pp. 295–304.
[50] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. IEEE Design, Autom. Test Eur. Conf.*, Apr. 2007, pp. 1–6.
[51] *MiBench* (2010) [Online]. Available: http://www.eecs.umich.edu/mibench/
[52] G. Fursin, J. Cavazos, M. O'Boyle, and O. Temam, "MiDataSets: Creating the conditions for a more realistic evaluation of iterative optimization," in *Proc. Int. Conf. High-Perform. Embedded Architectures Compil.*, Jan. 2007, pp. 245–260.
[53] *SimpleScalar* (2011) [Online]. Available: http://www.simplescalar.com/
[54] *MatLab* (2012) [Online]. Available: http://www.mathworks.com/
[55] K. Skadron, K. Sankaranarayanan, S. Velusamy, D. Tarjan, M.R. Stan, and W. Huang, "Temperature-aware micro-architecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.
[56] *Ansys 12.0* (2009) [Online]. Available: http://www.ansys.com
[57] T. Reis and T. Stykel, "PABTEC: Passivity-preserving balanced truncation for electrical circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 9, pp. 1354–1367, Sep. 2010.
[58] E. Rotem, J. Hermerding, C. Aviad, and C. Harel, "Temperature measurement in the Intel Core Duo processor," in *Proc. IEEE Int. Workshop Thermal Investigations ICs*, Jul. 2006, pp. 23–27.

Amir Zjajo (M'02) received the M.Sc. and D.I.C. degrees from Imperial College London, London, U.K., in 2000, and the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2010, all in electrical engineering.

He joined Philips Research Laboratories, Eindhoven, in 2000, as a Research Staff Member with the Mixed-Signal Circuits and Systems Group. From 2006 to 2009, he was with Corporate Research of NXP Semiconductors, Eindhoven, as a Senior Research Scientist. In 2009, he joined the Delft University of Technology, Delft, The Netherlands, as a Faculty Member with the Circuit and Systems Group. He has published more than 60 papers in referenced journals and conference proceedings, and holds more than 10 U.S. patents or patents pending. He is the author of *Low-Voltage High-Resolution A/D Converters: Design, Test and Calibration* (Springer, 2011, Chinese translation, 2013) and *Stochastic Process Variations in Deep-Submicron CMOS: Circuits and Algorithms* (Springer, in press). His current research interests include mixed-signal circuit design, signal integrity and timing, and yield optimization of VLSI.

Dr. Zjajo serves as a member of Technical Program Committee of IEEE Design, Automation and Test in Europe Conference, IEEE International Symposium on Circuits and Systems, and IEEE International Mixed-Signal Circuits, Sensors, and Systems Workshop.

Nick van der Meijs (M'87) received the M.Sc. and Ph.D. degrees from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 1985 and 1992, respectively.

He is currently an Associate Professor with the Circuits and Systems Group, Department of Micro Electronics and Computer Engineering, TU Delft. He is responsible for the content, organization, and quality of the B.Sc. and M.Sc. curricula in electrical engineering and computer engineering with TU Delft. He has co-authored some 100 papers on various topics, including design frameworks, interconnect optimization, and parasitics modeling. He was one of the lead developers of the SPACE 2-D and 3-D parasitic layout to circuit extractor. He is a regular reviewer for various EDA and design methodology conferences and journals, and he has served as a Topic Chair on multiple at conferences. He and his research group currently work both on modeling of parasitic effects in advanced integrated circuits and on circuit level design methods and tools for dealing with variability.

Rene van Leuken (M'80) was born in The Netherlands in 1955. He received the Ph.D. degree in electrical engineering from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 1988.

He is a Professor with the Circuit and Systems Group, TU Delft. He has been involved in many research projects, including ESPRIT, FP6, FP7, JESSI, MEDEA, and recently in MEDEA+ and ENIAC/CATRENE. He has published papers in all major conferences and workshops proceedings. His current research interests include high level system design, design automation, system design optimization, and DSP engines.

Dr. van Leuken is a member of the PATMOS Program Committee.