Composable Thermal Modeling and Simulation for Architecture-Level Thermal Designs of Multicore Microprocessors

HAI WANG, University of Electronic Science and Technology of China
SHELDON X.-D. TAN, University of California, Riverside
DUO LI, Synopsys
ASHISH GUPTA, Intel
YUAN YUAN, University of Electronic Science and Technology of China

Efficient temperature estimation is vital for designing thermally efficient, lower power and robust integrated circuits in nanometer regime. Thermal simulation based on the detailed thermal structures no longer meets the demanding tasks for efficient design space exploration. The compact and composable model-based simulation provides a viable solution to this difficult problem. However, building such thermal models from detailed thermal structures was not well addressed in the past. In this article, we propose a new compact thermal modeling technique, called *ThermComp*, standing for thermal modeling with composable modules. *ThermComp* can be used for fast thermal design space exploration for multicore microprocessors. The new approach builds the composable model from detailed structures for each basic module using the finite difference method and reduces the model complexity by the sampling-based model order reduction technique. These composable models are then used to assemble different multicore architecture thermal models and realized into SPICE-like netlists. The resulting thermal models can be simulated by the general circuit simulator SPICE. *ThermComp* tries to preserve the accuracy of fine-grained models with the speed of coarse-grained models. Experimental results on a number of multicore microprocessor architectures show the new approach can easily build accurate thermal systems from compact composable models for fast architecture thermal analysis and optimization and is much faster than the existing HotSpot method with similar accuracy.

Categories and Subject Descriptors: J.6 [**Computer-Aided Engineering**]: Computer-Aided Design

General Terms: Design, Algorithms

Additional Key Words and Phrases: Composable, multicore, model reduction, thermal modeling

ACM Reference Format:
Wang, H., Tan, S. X.-D., Li, D., Gupta, A., Yuan, Y., 2013. Composable thermal modeling and simulation for architecture-level thermal designs of multicore microprocessors. ACM Trans. Des. Autom. Electron. Syst. 18, 2, Article 28 (March 2013), 27 pages.
DOI: http://dx.doi.org/10.1145/2442087.2442099

This work is supported in part by NSF grant under No. CCF-1255899, in part by NSF Grant under No. CCF-0902885, in part by Semiconductor Research Corporation (SRC) grant under No. 2013-TJ-2417.
The work was performed when H. Wang was at UC Riverside.

Authors' addresses: H. Wang, School of Microelectronics & Solid-State Electronics, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054 China; S. X.-D. Tan, Department of Electrical Engineering, University of California, Riverside, CA 92521; email: stan@ee.ucr.edu; D. Li, Synopsys Incorporated, Mountain View, CA 94043; A. Gupta, Intel Corporation, Chandler, AZ 85226; Y. Yuan, School of Automation, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054 China.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2013 ACM 1084-4309/2013/03-ART28 $15.00

DOI: http://dx.doi.org/10.1145/2442087.2442099

1. INTRODUCTION

Continuous process scaling and device density rising lead to rapid power density increase and adverse thermal effects. This problem becomes more severe as the VLSI technology scales to the nanometer ranges. Excessively high on-chip temperature can cause many severe problems such as reduced reliability of chips and elevated cooling cost of the packaging [Gunther et al. 2001; Brooks and Martonosi 2001]. Thermal management and related design problems continue to be identified by the Semiconductor Industries Association Roadmap [ITRS 2010] as one of the five key challenges during the next decade to achieve the projected performance goals of the semiconductor industry. Thus, accurate and efficient thermal modeling and analysis are vital for the thermal-aware VLSI design [Pedram and Nazarian 2006] to improve performance, reliability, power reduction as well as online temperature regulation techniques [Brooks and Martonosi 2001; Skadron et al. 2003].

Multicore techniques mitigate the exponential growth of temperature resulting from the exponential increase of power density [Li et al. 2006; Intel Corporation 2006; AMD Inc. 2006; Liao et al. 2010, 2011]. Multicore computing simply increases the total throughput via parallel computation with lower voltage and frequency to meet the thermal constraints. But thermal effects and the resulting reliability concerns, such as NBTI effect, are influenced by the placement of CPU cores and shared caches, loading of programs, and cooling solutions at the package level. So it is vital to accurately estimate the temperature during the floorplanning, architecture design [Huang et al. 2004; Skadron et al. 2003; Huang et al. 2006; Li et al. 2009, 2008; Yang et al. 2007] and the runtime [Wang et al. 2011] of the multicore microprocessors.

Traditional thermal analysis solves the thermal diffusion (partial differential) equation directly using numerical approaches such as FEM (finite element), FDM (finite difference) and CFD (computational fluid dynamics). These approaches are accurate given detailed thermal structures. However, the resulting equation sizes can be prohibitively large for design explorations, hence, thermal simulation starting from detailed thermal structures by solving thermal diffusion equations no longer meets the demanding design tasks for efficient design space exploration. As thermal effects become the first-class design constraints, efficient thermal analysis calls for more efficient solutions. The compact and composable model-based simulation provides a viable solution to this difficult problem. *Composable* is defined as the ability to build the basic module models, which are then used to assemble different large systems. We also call the assembling process as *composition*. In addition, the large system assembled from the basic modules is called the *composite model*. This strategy is similar to the model-based electronic circuit simulation method, such as SPICE, which no longer solves the basic Poisson's equations directly at the device level to obtain the voltage and current information. Similar to the SPICE device models of the CMOS and BJT transistors, we propose new composable thermal models, which can be reused and easily connected to build various thermal circuits and systems.

In this article, we address the emerging problem we have introduced by proposing a novel composable thermal modeling approach. We present the new approach in the context of fast thermal analysis and design for multicore microprocessors at the architecture level. The new approach, called *ThermComp*, which stands for thermal modeling with composable modules, builds the compact thermal models for the basic modules (CPU core and cache, for example) from detailed thermal models generated by the finite difference method. Then, it applies the sampling-based reduction technique to reduce the complexity of those models. To make the complexity reduction efficient for thermal models with many ports, port reduction by means of adjacent port merging has been introduced. Such port reduction naturally leads to a new two-grid discretization scheme where a uniform global coarse grid is used for all the boundary grids and a fine grid is used for the internal grids of each module. The coarse grid is also justified by the smooth thermal gradients at the boundary.

The composable modeling has three main advantages over traditional thermal modeling: first, it enables the reuse of the compact components in the thermal modeling process, especially for the multi/many-core systems: only few compact models need to be built for very large system with multiple identical components; second, the change in the design architecture only requires recomposing the compact components instead of rebuilding the whole model; third, it makes compact modeling and simulation of extremely large thermal system possible (in reasonable time and memory) because directly applying model order reduction on or simulation of the large system is impossible due to the memory and computational cost limitations. The proposed method tries to achieve both high efficiency and accuracy for thermal simulation and verification. Hence it will always help when the trade-off of efficiency and accuracy has been made in thermal-oriented chip design and optimization at the architecture and core levels.

Experimental results on a number of multicore microprocessor architectures show the new approach can easily build accurate composite thermal models from compact models of different modules for the fast architecture thermal analysis and optimization. The compact models lead to orders of magnitude speedup over the standard finite difference method with marginal error.

The rest of this article is organized as follows: Section 2 presents some existing works. Section 3 introduces the thermal modeling problem we are trying to solve. Section 4 presents the new composable thermal modeling method. Section 5 shows the experimental results and Section 6 concludes the article.

2. EXISTING WORKS

Many compact static and transient thermal modeling methods at different levels (parts, package, and board) have been proposed in the past. One popular approach is based on simple thermal resistance and capacitance networks subject to different thermal boundary conditions [Lasance et al. 1995; Christiaens et al. 1998; Augustin et al. 2005]. The traditional limitation of these methods is to determine appropriate RC values of elements, especially for the complex geometries and boundary conditions. The RC values are typically determined and optimized against field numerical or analytic results [Gerstenmaier and Wachutka 2002; Pape et al. 2004] and the measured data [Rencz et al. 2003]. A transient simulation technique using alternating direction implicit (ADI) method is proposed to enhance the simulation efficiency of the three dimensional thermal RC circuit [Wang and Chen 2002]. Another viable approach is by means of model order reduction of the large linear dynamic thermal systems after the spatial discretization. Existing approaches apply the multipoint Krylov subspace method [Codecasa et al. 2003] and the multivariate moment matching method [Codecasa et al. 2006]. Recently, many other fast thermal modeling and simulation techniques have been proposed, such as the compact thermal models based on the floorplan at architecture level [Huang et al. 2004; Skadron et al. 2003; Huang et al. 2006], the black-box behavioral thermal models [Li et al. 2009, 2008] and the spatially adaptive thermal modeling technique ISAC [Yang et al. 2007].

3. THERMAL SIMULATION AND NEW COMPOSABLE MODELING PROBLEM

In this section, we briefly present the basics of microprocessor thermal modeling. The finite difference method based thermal modeling technique is shown in Section 3.1. The boundary condition handlings are introduced next, in Section 3.2 and Section 3.3. Finally, the compact modeling technique and its corresponding model complexity reduction problem are shown in Section 3.4 and Section 3.5, respectively.

3.1. Thermal Simulation Problem from First Principles

At the circuit, package and board levels, the heat transfer phenomena is governed by the following heat differential equation [Cheng et al. 2000]:

$$\rho C_p \frac{\partial T(\vec{r}, t)}{\partial t} = \nabla \cdot [\kappa(\vec{r}, T) \cdot \nabla T(\vec{r}, t)] + g(\vec{r}, t), \tag{1}$$

which is subject to the following general thermal boundary condition (Robin's boundary condition)

$$\kappa(\vec{r}, T) \frac{\partial T(\vec{r}, t)}{\partial n_i} = h_i(T(\vec{r}, t) - T_{amb}). \tag{2}$$

In (1), T (K) is the temperature, ρ (Kg/m^3) is the density of the material, C_p $(J/Kg{\cdot}K)$ is the mass heat capacity, κ $(W/m{\cdot}K)$ is the thermal conductivity, and g (W/m^3) is the heat energy generation rate. In (2), n_i is the outward direction normal to the boundary condition i, h_i (W/m^2K) is the heat-transfer coefficient (for the convective interface), and T_{amb} is the ambient temperature surrounding the thermal systems. If $h_i = 0$, the boundary condition is adiabatic (isolated), otherwise, it is convective. Note that the thermal conductivity κ differs for different materials and also depends on the temperature.

For the finite difference based numerical analysis, a seven-point discretization scheme can be applied for (1) in three dimensions. The thermal structure will be decomposed into numerous rectangular parallelepipeds, which may be of nonuniform sizes and shapes. The adjacent elements interact with each other via heat diffusion and the elements interact with boundaries via specific boundary conditions. Each element may have power sources, temperature, and equivalent thermal capacitance and resistance to its adjacent elements. Assuming homogeneous material and temperature independent κ, $\nabla \cdot [\kappa(\vec{r}, T) \cdot \nabla T(\vec{r}, t)]$ becomes $\kappa \nabla^2 T(\vec{r}, t)$. Equation (1) becomes a linear partial differential equation

$$\rho C_p \frac{\partial T(\vec{r}, t)}{\partial t} = \kappa \left[\frac{\partial^2 T(\vec{r}, t)}{\partial x^2} + \frac{\partial^2 T(\vec{r}, t)}{\partial y^2} + \frac{\partial^2 T(\vec{r}, t)}{\partial z^2} \right] + g(\vec{r}, t). \tag{3}$$

After the space discretization, the temperature $T(x, y, z, t)$ at the grid point (i, j, k) is replaced by $T(i\Delta x, j\Delta y, k\Delta z, t)$. We denote $T(i\Delta x, j\Delta y, k\Delta z, t)$ by $T_{i,j,k}$ for the rest of the article. According to the central difference discretization, we will have the following discretized equation for grid (i, j, k)

$$\begin{aligned} \rho C_p M \frac{\partial T(x, y, z, t)}{\partial t} &= -2(G_x + G_y + G_z)T_{i,j,k} + G_x T_{i-1,j,k} \\ &\quad + G_x T_{i+1,j,k} + G_x T_{i,j-1,k} + G_x T_{i,j+1,k} \\ &\quad + G_x T_{i,j,k-1} + G_x T_{i,j,k+1} + M g_{i,j,k,t}, \end{aligned} \tag{4}$$

where Δx, Δy, Δz are the discretization steps along the x, y, z axes, $M = \Delta x \Delta y \Delta z$. $G_x = \kappa \Delta y \Delta z / \Delta x$, $G_y = \kappa \Delta x \Delta z / \Delta y$ and $G_z = \kappa \Delta x \Delta y / \Delta z$.

3.2. Adiabatic Thermal Condition for Composibility

The composable models need to be constructed such that the composite system connected by these modules is the same as the system constructed directly from the original structure. It can be proved that to make the thermal model composable, the *adiabatic* thermal conditions (special Neumann's boundary condition) [Cheng et al. 2000]

$$\kappa(\vec{r}, T) \frac{\partial T(\vec{r}, t)}{\partial n_i} = 0 \tag{5}$$

should be added at the thermal modules.

Specifically, for the *adiabatic* condition, there are no thermal exchanges between the boundary nodes and outside. When two modules are connected together, the connected boundary nodes of the two modules will become one node in the new system. Heat will flow via normal diffusion as there is no boundary between the two modules. The interface with *adiabatic* thermal conditions is called the *composable interface* in this article.

We remark that the isothermal or adiabatic thermal condition assumption actually is not a restriction for our modeling process. If we want to merge two thermal modules together, the merged interface must be adiabatic so that the merged thermal system will look like a system where normal thermal diffusions will happen at the merged boundaries and those merged boundaries will become internal parts of the whole system. For boundaries or interfaces, which are not merged, any thermal conditions can be applied and there is no restriction on those boundaries. Since the modules are always merged with other modules to build a complete thermal system, the adiabatic thermal interface or boundaries will never be the actual interfaces or boundaries for the complete thermal systems.

3.3. Equivalent Circuits for Other Boundary Conditions

For other boundaries, which interact with the outside directly via convection or other heat exchange mechanisms, a proper thermal condition (Robin's boundary condition) in terms of equivalent thermal resistance and independent source should be added at the ports as shown as follows.

To illustrate this, let us consider only one dimension (x direction). From (2), we carry out the discretization on the x direction

$$\begin{aligned} \kappa \frac{\partial T}{\partial x} &= h_x(T_{amb} - T) \\ \kappa \frac{T_0 - T_1}{\Delta x} &= h_x(T_{amb} - T_0). \end{aligned} \tag{6}$$

Here T_0 represents the temperature of a node just at the boundary and T_1 is the temperature of the adjacent node inside the module.

(6) is interpreted as the temperature relation between two branches (T_0, T_1) and (T_1, T_{amb}). Since we already know the thermal conductance between T_0 and T_1 through discretization, which is denoted as G_x, (6) readily becomes a three-branch KCL equation at T_0

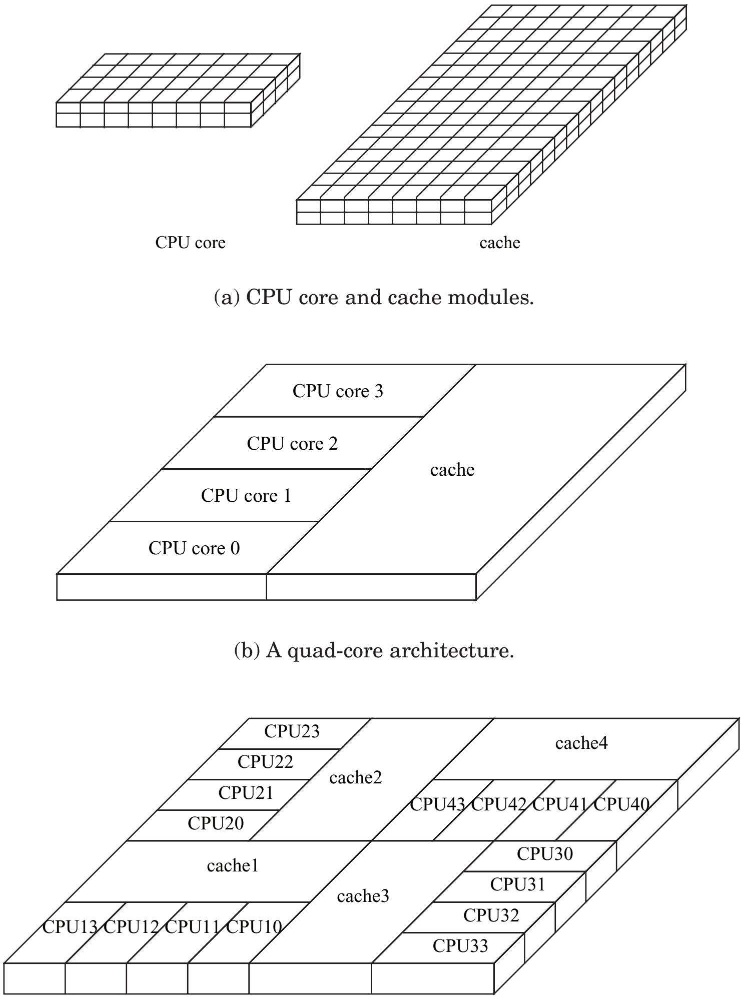
$$(T_0 - T_1)G_x = \frac{h_x \Delta x G_x}{\kappa} T_{amb} - \frac{h_x \Delta x G_x}{\kappa} T_0. \tag{7}$$

Then, an equivalent circuit is generated with a current source (with $\frac{h_x \Delta x G_x}{\kappa} T_{amb}$) and a resistor (with conductance $\frac{h_x \Delta x G_x}{\kappa}$) connected to the ground at T_0.

We can build a library where for each thermal module, such as the CPU core, there are different thermal conditions and different composable interfaces.

3.4. Simulation by Compact Thermal Modeling for Multicore Systems

A traditional thermal analysis method directly solves the discretized thermal system in (3). In order to capture the junction temperature of a multicore chip, a fine grid needs to be used, resulting in a very large thermal system. In this case, directly solving the system can be very expensive or even prohibitive. A solution is to build a reduced system using the model reduction techniques. However, building a reduced model from a very large scale system is still expensive. Moreover, a slight change in the multicore architecture requires redoing the whole finite difference discretization and rebuilding



(a) CPU core and cache modules.

(b) A quad-core architecture.

(c) A 16-core architecture.

Fig. 1. CPU core, cache, a quad-core architecture and a 16-core architecture.

the reduced model from the very large discretized system. As a result, it will be unacceptable for the thermal design exploration for various multicore architectures as the spatial discretization, reduction and simulation have to be done for every architecture during the optimization steps.

Instead of directly using (4) or its reduced model for thermal analysis, a more efficient composable thermal-model based approach is more desirable. As shown in Figure 1(a), we first build two compact composable models for CPU core and cache modules through the finite difference method and model reduction. Then, using the two models, we can build the multicore thermal system such as the quad-core system shown in Figure 1(b), the 16-core system shown in Figure 1(c) or other multicore thermal systems. We assume the CPU cores are the same for simplicity, but our approach can be easily extended to different CPU cores and other functional blocks. Figure 2 is the lateral structure view of a typical package for the multicore system. Typically the heat generated at the

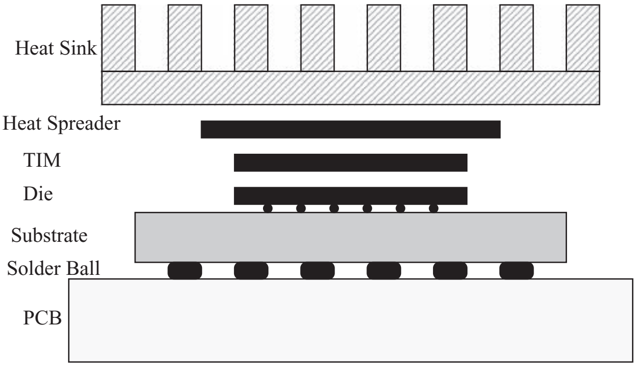


Fig. 2. The literal structure view of the multicore system package.

die is conducted from the bottom to the heat spreader and then to the heat sink. In this article, we do not directly consider those package structures. Instead, we capture the effects of those package structures on the die using proper thermal conditions. For brevity's sake, we assume the other sides of the die do not have heat exchange (adiabatic condition).

As discussed before, our goal is to build the compact composable thermal model for each module (CPU core and cache) so we can quickly build the composite models for different multicore architectures instead of building the whole thermal systems from scratch every time.

3.5. Model Complexity Reduction Problem

We already have detailed thermal models for the modules in the finite difference framework with certain boundary conditions. The modeling problem now is to build the compact thermal model for each module.

Specifically, if we have n discretized elements (grids) with specific boundary conditions, Equation (4) becomes a linear ordinary differential equation

$$C\frac{dT(t)}{dt} + GT(t) = Bg(t), \tag{8}$$

where $C \in \mathbb{R}^{n\times n}$ is the thermal capacitance matrix, $G \in \mathbb{R}^{n\times n}$ is the thermal conductance matrix. $B \in \mathbb{R}^{n\times p}$ is the position matrix for a total of p ports including the boundary ports and the power dissipation sources. $g(t) \in \mathbb{R}^{p\times 1}$ represents the power injections from the boundary ports and the power dissipation sources. The boundary port sources modeling has been discussed in Section 3.3 and the power dissipation sources modeling will be presented in Section 4.3.

To reduce the model complexity, model reduction techniques can be applied to (8). However, we will show that existing reduction methods do not work well for the thermal models with many ports in general, which will be addressed in Section 4.1.

4. NEW COMPOSABLE THERMAL MODELING METHOD

In this section, the new composable thermal modeling method is presented. A two-grid discretization scheme is introduced in Section 4.1 to reduce the number of ports of each module in order to enhance the model order reduction efficiency. The stability and the property of the new discretization is shown in Section 4.2. Next, in Section 4.3, three power dissipation models are presented and their handling by the composable thermal modeling method is studied. Section 4.4 introduces the thermal model order reduction technique which generates smaller model for faster simulation speed. Finally, we show

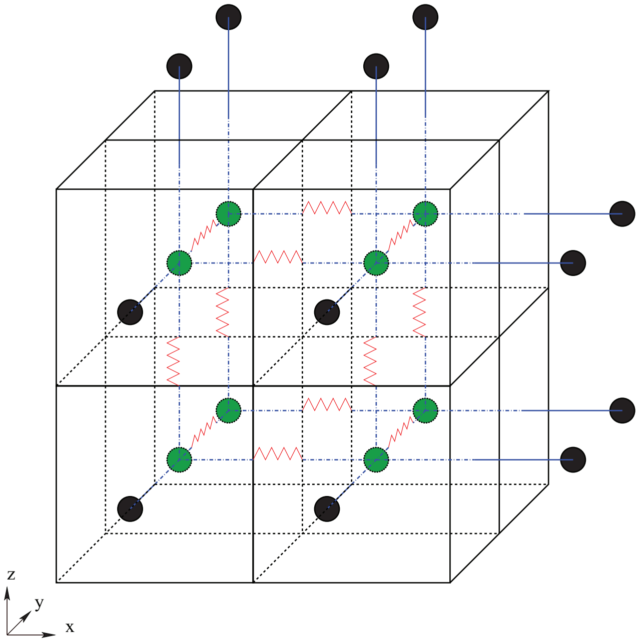


Fig. 3. A 2 × 2 × 2 meshed structure case. The nodes and the ports are represented by light solid circles and dark solid circles, respectively.

how to realize the reduced thermal model into SPICE subcircuit for easier composition in Section 4.5 and how to retrieve the internal node temperature from the reduced thermal model in Section 4.6.

4.1. Two-Grid Scheme for Discretization

To build compact thermal models from detailed models generated by the finite difference method (or other discretization schemes), one viable way is to partition the original thermal structure into many building blocks and perform reduction on these modules. In case of the multicore processor example shown in Figure 1(b) and (c), the natural building blocks are the CPU core module and the cache module.

However, such a simple strategy does not work well in practice. The main reason is that many ports will be generated for such thermal modules since each grid in one direction will generate one port as shown in Figure 3. As a result, the number of ports can be huge if the mesh is large.[1] Existing model order reduction techniques such as the Krylov subspace or the Gramian based methods do not work well when the number of ports is large because the projection space and the computational time are tightly determined by the port counts. Reducing the number of ports is vital for building the compact thermal models.

The many ports problem mentioned already can be solved by assuming the adjacent boundary nodes are isothermal. In this way, we can reduce the number of ports by merging some adjacent ports into one port. Meanwhile, such port merging or reduction needs to ensure the composability of the thermal modules for fast thermal design exploration. So the port reduction needs to follow geometrical patterns such that the resulting thermal modules can be easily assembled to build different thermal systems.

[1] The internal power dissipation sources also contribute to the number of ports. But as shown in Section 4.3, only one port is enough to represent all the power dissipation sources in a module.

To achieve the two goals: reducing the port number and ensuring composability, the port merging should equivalently lead to a coarser global grid among all the modules so they can be easily assembled based on this global grid. In the following, we use a $2 \times 2 \times 2$ meshed structure example (in finite difference scheme) shown in Figure 3 to illustrate the idea.[2]

For this meshed structure, there are 8 nodes (cubes) denoted by light solid circles and 24 ports by dark solid circles. Please note, for this special case, every node is on the boundary and is the vertex of the cube. Thus, each of them has 3 ports connected. We also indicate the connection between the nodes by an equivalent thermal resistance. Each node also has an equivalent thermal capacitance connected to the ground, which is not displayed for simplicity. If we apply the adiabatic conditions to all the 6 interfaces, the G, C, T, g and B matrices are shown in (9)–(13) respectively.

$$G = \kappa \left[\begin{array}{cccccccc}
3 & -1 & -1 & & -1 & & & \\
-1 & 3 & & -1 & & -1 & & \\
-1 & & 3 & -1 & & & -1 & \\
 & -1 & -1 & 3 & & & & -1 \\
-1 & & & & 3 & -1 & -1 & \\
 & -1 & & & -1 & 3 & & -1 \\
 & & -1 & & -1 & & 3 & -1 \\
 & & & -1 & & -1 & -1 & 3
\end{array}\right] \tag{9}$$

$$C = \rho C_p I_{8\times8} \tag{10}$$

$$T(t) = \left[T_{1,1,1}\ \ T_{2,1,1}\ \ T_{1,2,1}\ \ T_{2,2,1}\ \ T_{1,1,2}\ \ T_{2,1,2}\ \ T_{1,2,2}\ \ T_{2,2,2}\right]^T \tag{11}$$

$$\begin{aligned} g(t) = [\, & g_{1,1,0}\ \ g_{2,1,0}\ \ g_{1,2,0}\ \ g_{2,2,0}\ \ g_{1,1,3}\ \ g_{2,1,3}\ \ g_{1,2,3}\ \ g_{2,2,3}\ \ g_{1,0,1}\ \ g_{2,0,1}\ \ g_{1,0,2}\ \ g_{2,0,2} \\ & g_{1,3,1}\ \ g_{2,3,1}\ \ g_{1,3,2}\ \ g_{2,3,2}\ \ g_{0,1,1}\ \ g_{0,2,1}\ \ g_{0,1,2}\ \ g_{0,2,2}\ \ g_{3,1,1}\ \ g_{3,2,1}\ \ g_{3,1,2}\ \ g_{3,2,2}\,]^T \end{aligned} \tag{12}$$

$$B = \left[\begin{array}{cccccccccccccccccccccccc}
1 & & & & & & & & 1 & & & & & & & & 1 & & & & & & & \\
 & 1 & & & & & & & & 1 & & & & & & & & & & & 1 & & & \\
 & & 1 & & & & & & & & & & 1 & & & & & 1 & & & & & & \\
 & & & 1 & & & & & & & & & & 1 & & & & & & & & 1 & & \\
 & & & & 1 & & & & & & 1 & & & & & & & & 1 & & & & & \\
 & & & & & 1 & & & & & & 1 & & & & & & & & & & & 1 & \\
 & & & & & & 1 & & & & & & & & 1 & & & & & 1 & & & & \\
 & & & & & & & 1 & & & & & & & & 1 & & & & & & & & 1
\end{array}\right]. \tag{13}$$

$T_{i,j,k}$ is the temperature of the node at (i, j, k) in Figure 3 where i, j and k are indexes in the x, y, z directions, respectively. $g_{i,j,k}$ is the power injection into the boundary port at (i, j, k). (i, j, k) can be the position of a node (with $T_{i,j,k}$) or a port (with $g_{i,j,k}$). Notice that $I_{8\times8}$ is an 8×8 unit matrix. So we have a symmetric positive definite (s.p.d.) matrix G and a unit matrix C. Matrix G will still be s.p.d. when we have power dissipation sources since the power dissipation sources will only change the B matrix. The s.p.d.

[2] For simplicity, in this example, we ignore the internal power sources whose effects will be discussed later. Also, we do not show the thermal capacitors and only the ports on the three front faces are shown.

property can be preserved during the projection based reduction, allowing the reduced models to be further optimized and realized. It will be shown in Section 4.5.

In this simple case, we have more ports than nodes. As a result, existing model reduction methods such as the Krylov subspace based method do not work. For instance, even with only one moment matching, one will end up with a larger reduced model (24 × 24 matrices, which are also full matrices!) than the original one (8 × 8 sparse matrices). Although for the large real finite difference thermal model, the port number is one order of magnitude smaller than the grid number, it still greatly degenerates the model reduction efficiency.

In general, if the number of grids in the x, y and z directions are X, Y and Z, the number of nodes in the detailed thermal model is XYZ, while the number of ports is

$$f_{port} = 2(XY + XZ + YZ), \tag{14}$$

which indicates the number of ports grows quadratically with the grid count. It is impossible to build very compact thermal models if the port number is not reduced.

To reduce the number of ports, one idea is to reduce the number of grids at the boundaries using a large grid size assuming the boundary surface of the large grid is isothermal. We can simply merge the adjacent boundary nodes into one node and then build the new finite difference matrices. But this scheme requires the construction of the new G and C matrices in (8). A better way is to just merge the adjacent ports of the boundary nodes into one port assuming all the involved nodes are connected to the same port. Using the example in Figure 3, if we merge 4 adjacent ports into 1 port in each boundary face as show in Figure 4, the port number will be reduced to 6. The B and $g(t)$ matrices after this boundary merge are shown as

$$B_m = \begin{bmatrix} 1 & & 1 & & 1 & \\ 1 & & 1 & & & 1 \\ 1 & & & 1 & 1 & \\ 1 & & & 1 & & 1 \\ & 1 & 1 & & 1 & \\ & 1 & 1 & & & 1 \\ & 1 & & 1 & 1 & \\ & 1 & & 1 & & 1 \end{bmatrix} \tag{15}$$

$$g_m(t) = [g_{x,y,0}, g_{x,y,3}, g_{x,0,z}, g_{x,3,z}, g_{0,y,z}, g_{3,y,z}]^T, \tag{16}$$

where the subscripts x, y and z represent the merged indexes in the x, y and z directions, respectively. Such a port reduction also leads to a larger grid size at the boundaries, as shown in Figure 4.

The boundary merging will introduce errors because some adjacent nodes with different temperatures are merged together. The errors, however, are relatively small when the power dissipation is uniformly distributed inside the module and will be larger when the power dissipation distribution has a large gradient just next to the boundary. As shown in our experiment, even in the latter case, the errors are still within 2-3°C and the large errors only appear at the boundaries. In order to further improve the accuracy, a finer boundary grid can be used through merging less ports into one port. This will result in a less compact reduced model as the model reduction efficiency is the trade-off in this case.

4.2. Stability and Property of the New Discretization

In the new two-grid discretization, we only change the right-hand side matrix B without altering the G and C matrices in (8), so the resulting thermal matrices will still be

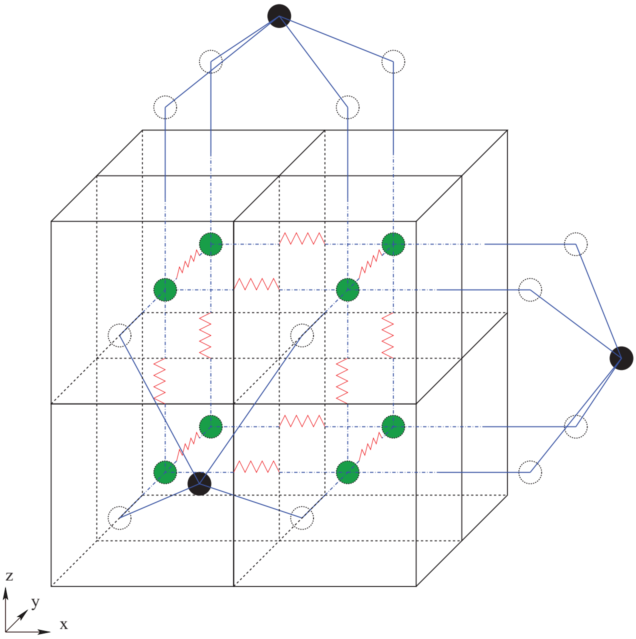


Fig. 4. A 2 × 2 × 2 meshed structure case where the boundary faces (ports) are merged. The original ports are shown as the hollow circles and the new ports are represented by the dark solid circles.

symmetric positive definite. This is important for more accurate reduction and passivity preservation during the reduction process.

For the finite difference method, if h is the space difference (assume it is the same for all the three dimensions), the time step Δt cannot be too large to ensure the numerical stability of the integration. As a result, Δt will be restricted by the following equation (when the explicit time discretization method is used) [Ozisik 1994]:

$$0 < \Delta t < \frac{1}{2\beta\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)} \tag{17}$$

where $\beta = \frac{\kappa}{\rho C_p}$. For the new method, this is not an issue. First, we typically use implicitly discretization methods such as the Backward Euler, Trapezoidal or Gear methods, which do not restrict the time step sizes and are always stable for a stable system. Even if we use the explicit time discretization method such as the Forward Euler method, the resulting model is still stable (assume the simulation of the original model without reduction is stable)—since we increase the space difference at the boundary, it will give a larger upper bound for Δt. As a result, if we still use the same Δt for the model without port merging, the simulation will be definitely stable.

4.3. Power Dissipation Modeling

Each module has its own power dissipation property, approximately characterized as the power dissipation distribution within its 3D structure. In this article, the power dissipation is modeled in three ways. The first model assumes the power inside a module is uniformly distributed. We call it the *uniform model*. The second model assumes the power is concentrated in some power centers, and is called the *center model*. The third model, called the *distribution model*, follows the measured or given power distribution across the module. The first two models are simple but less accurate as some simplifications are made. Specifically, the uniform model generally gives optimistic junction temperature predictions while the center model generally gives pessimistic ones. Although the distribution model is more accurate, it requires the detailed power distribution, which may be unavailable. Please note that the first two models are just two special cases of the third model: the uniform model is the distribution model with a uniform distribution and the center model is the distribution model with zeros everywhere except for the power centers.

In our composable thermal model, all of the three power dissipation models can be easily handled with a simple formulation. Specifically, the power dissipation of a module is represented at the right hand side of our thermal model as $B_{pw} \times g_{pw}(t)$. $B_{pw} \in \mathbb{R}^{n\times 1}$ represents the power distribution in the module and is one column of the $B \in \mathbb{R}^{n\times p}$ matrix in (8) (the other columns in B are for the boundary condition power sources). $g_{pw}(t)$ is the total power dissipation in the module and is an element of $g(t)$. Each element of B_{pw} corresponds to a finite difference node and its value is the weight of the power dissipation of the finite difference node against the total power dissipation of the module. For the uniform model, each element of B_{pw} is $1/n$ and for the center model, all elements have the value 0 except for the power center nodes whose value is $1/n_c$ where n_c is the number of the power centers. The element values of the distribution model are set according to the provided power distribution. Given the power distribution function as $f(x, y, z)$, the ith element of B_{pw}, denoted as b^i_{pw}, is calculated as

$$b^i_{pw} = \frac{\int_{x_i^-}^{x_i^+}\int_{y_i^-}^{y_i^+}\int_{z_i^-}^{z_i^+} f(x,y,z)\,dz\,dy\,dx}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y,z)\,dz\,dy\,dx}, \tag{18}$$

where x_i^-, y_i^- and z_i^- are the starting coordinates of the ith finite difference element and x_i^+, y_i^+ and z_i^+ are its ending coordinates.

We use a one dimensional module (x direction) as an example to illustrate the proposed idea. Assume the module is positioned on $x \in [0, 1]$ and discretized into three nodes. For the uniform model, $B_{pw} = [1/3, 1/3, 1/3]^T$ and $B_{pw} \times g_{pw}(t) = [g_{pw}(t)/3, g_{pw}(t)/3, g_{pw}(t)/3]^T$ indicate the total power $g_{pw}(t)$ is equally divided into three parts which are injected into three finite difference nodes separately. For the center model with only one power center at $x = 0.5$, there will be no power injection into the first and the third nodes while all the power flows into the second node. As a result, the B_{pw} vector is $[0, 1, 0]^T$. For the distribution model, assume we have the simple power distribution $f(x) = x$, the B_{pw} vector should be $[1/9, 1/3, 5/9]^T$ according to (18).

In summary, all the three power dissipation models can be handled by our thermal model using the same formulation. Additionally, no matter which power dissipation model is used, there will be only one power dissipation handling column B_{pw} inside B in our thermal model. As a result, the model reduction efficiency will not be degraded.

4.4. Thermal Model Reduction

Reducing the complexity of linear dynamic systems by means of model order reduction has been studied intensively for parasitic electronic circuits in the past [Antoulas 2005; Tan and He 2007]. For compact thermal modeling, the Krylov subspace based approaches have been applied to reduce the large models [Codecasa et al. 2003, 2006]. In this article, we apply a more accurate sampling-based reduction technique [Willcox and Peraire 2002; Phillips and Silveira 2005; Wang et al. 2012], which is based on the globally accurate balanced truncation realization reduction scheme.

For a dynamic thermal system described in (8), the goal is to find a subspace represented by the projection matrix $V \in \mathbb{R}^{n\times k}$, resulting in the approximation

$$T \approx V\tilde{T}, \tag{19}$$

where $\tilde{T} \in \mathbb{R}^{k\times 1}$ is the state in the reduced model and $k \ll n$. This is not an equation but an approximation because $\tilde{T}$ has a smaller dimension than T and the information in some dimensions may have been lost. The resulting thermal equation becomes

$$\tilde{C}\frac{d\tilde{T}(t)}{dt} + \tilde{G}\tilde{T}(t) = \tilde{B}g(t), \tag{20}$$

where

$$\tilde{G} = V^T GV,\ \tilde{C} = V^T CV,\ \tilde{B} = V^T B \tag{21}$$

with $\tilde{G} \in \mathbb{R}^{k\times k}$, $\tilde{C} \in \mathbb{R}^{k\times k}$, $\tilde{B} \in \mathbb{R}^{k\times p}$. The resulting reduced model should preserve the port behaviors of the original system.

One way to obtain the projection matrix V is by means of the truncated balanced realization (TBR) method. TBR first performs the coordinate changes such that the controllability and observability (described by their corresponding Gramians X and Y) are the same for every state. In this way, the balanced weak states can be truncated by picking only the dominant eigenspace of XY as the projection matrix V. However, the computational cost to obtain the Gramians is $O(n^3)$, which makes it too expensive for integrated circuits problems and thus an efficient Gramian approximation technique is highly appreciated.

Recently, some fast TBR methods have been proposed [Willcox and Peraire 2002; Phillips and Silveira 2005] to mitigate the high computational cost of the standard TBR method, where the Gramians are approximated using the Monte-Carlo sampling approach. Specifically, the Gramian, denoted as X,[3] is explicitly written as the following integral in the frequency domain

$$X = \int_{-\infty}^{+\infty} (j\omega C + G)^{-1} BB^T (j\omega C + G)^{-H} d\omega, \tag{22}$$

where the superscript H denotes the Hermitian transpose. The dominant eigenspace of X is used as the projection matrix V [Phillips and Silveira 2005] for model order reduction. As discussed before, computing the exact values of X in (22) is extremely expensive for large scale systems. A cheaper computation of the Gramian X is achieved by evaluating the definite integral in (22) by summation approximation, using numerical quadrature methods [Iserles 1996] as discussed in the following.

The integral of a function $f(x)$ can be approximated as summation by numerical quadrature methods as

$$\int_a^b f(x)dx \approx \sum_{k=1}^{m} w_k f(x_k), \tag{23}$$

where w_k, $k = 1, 2, \ldots, m$, are referred to as the quadrature point weights, while the interpolation points x_k, $k = 1, 2, \ldots, m$, are called the quadrature points. The selections of w_k and x_k depend on the quadrature methods, such as Newton-Cotes, Gaussian quadrature rules or even random-based Monte Carlo method [Iserles 1996].

[3] For thermal system, which can be interpreted as the equivalent RC circuit, both the controllability Gramian X and observability Gramian Y are identical.

According to (23), the Grammian X in the integral form (22) can be also approximated as $\hat{X}$ in summation form like the following

$$X \approx \hat{X} = \sum_{k=1}^{m} w_k z_k z_k^H, \tag{24}$$

where

$$z_k = z(j\omega_k) = (j\omega_k C + G)^{-1} B \tag{25}$$

is called the kth snapshot of the system in the frequency domain (let ω_k be the kth sample point in frequency) and w_k is the weight from a specific numerical quadrature method.

For conveniences, $\hat{X}$ in (24) can be equivalently written in matrix form as

$$\hat{X} = ZW^2 Z^H, \tag{26}$$

where

$$Z = [z_1, z_2, \ldots, z_m] \tag{27}$$

and W is a diagonal matrix with diagonal entries $w_{kk} = \sqrt{w_k}$.

As the last step, the dominant eigenspace of X needs to be computed to serve as the projection matrix V. Because from (26), there is

$$\hat{X} = (ZW)(ZW)^H, \tag{28}$$

the eigenspace of $\hat{X}$ is the same as the left singular space of ZW. In order to save computational cost, instead of explicitly performing eigenvalue decomposition on $\hat{X}$, singular value decomposition (SVD) on ZW is usually performed as

$$ZW = VSU. \tag{29}$$

Then V, which gives the dominant eigenspace of $\hat{X}$, is used as the projection matrix. Finally, the reduced model can be easily computed using (21), which completes the model reduction process.

4.5. Circuit Realization and Model Generation

We have obtained the reduced matrices of the composable models. The next step is to generate the composite system using these models. One method is directly composing the module matrices into the whole system matrices. However, the reduced composable model has complex structures and many new node connections will be introduced by composing the two or more modules directly at the matrix level. As a result, direct matrix level composition requires reformulating the structure of the composable model matrices, which is very complicated and error-prone. Instead, we can package the composable model as a black box and use it for easy composition of the whole system. In this article, we realize the reduced composable model matrices into a SPICE netlist and package it as a subcircuit. SPICE will automatically build the composite system matrices from the hierarchical netlist.

Before realization, we can first make the reduced thermal system simpler by diagonalizing $\tilde{G}$ and $\tilde{C}$ which are originally dense matrices. Since the congruence transformation is used in (21), the reduced matrices $\tilde{G}$ and $\tilde{C}$ are symmetric and $\tilde{G}$ is positive definite. As a result, the reduced model can be further diagonalized by a generalized eigen-decomposition of a definite matrix pencil $\tilde{C} - \lambda\tilde{G}$ [Bai et al. 2000]. With the eigenvector matrix $P \in \mathbb{R}^{k\times k}$ of $\tilde{C} - \lambda\tilde{G}$, we can perform the coordinate change of (20) as

$$\tilde{T} = P\hat{T} \tag{30}$$

and generate

$$\hat{C}\frac{d\hat{T}(t)}{dt} + \hat{G}\hat{T}(t) = \hat{B}g(t), \tag{31}$$

where

$$\hat{G} = P^T\tilde{G}P,\ \hat{C} = P^T\tilde{C}P,\ \hat{B} = P^T\tilde{B}. \tag{32}$$

$\hat{G}$ and $\hat{C}$ are two *diagonal* matrices here. The diagonalized system in (31) can be realized into RC circuits with controlled sources, and then simulated using SPICE-type simulators [Codecasa et al. 2003].

Next, all the reduced composable models are realized into the SPICE compatible format using SPICE *.subckt* command. Specifically, the i-th diagonal element in the diagonal matrix $\hat{G}$, denoted as $\hat{G}_{ii}$, is realized into a resistor from the i-th node to the ground with the value $1/\hat{G}_{ii}$; the i-th diagonal element in the diagonal matrix $\hat{C}$, denoted as $\hat{C}_{ii}$, is realized into a capacitor from the i-th node to the ground with the value $\hat{C}_{ii}$. Realization of the $\hat{B}$ matrix is more complicated. The element in the i-th row and j-th column in the dense matrix $\hat{B}$, denoted as $\hat{B}_{ij}$, is realized into two components: one component is a current controlled current source (CCCS) in parallel with the i-th resistor and capacitor. The j-th independent current source (power source) is the corresponding control current with $\hat{B}_{ij}$ as the current gain. The other component is a voltage controlled voltage source (VCVS) in series with the j-th independent current. The i-th node voltage is the corresponding control voltage with $\hat{B}_{ij}$ as the voltage gain.

We would like to demonstrate a simple example to illustrate the circuit realization process. Given a small 2 × 2 diagonalized system with system matrices $\hat{C}$, $\hat{G}$, and $\hat{B}$ as the following

$$\hat{C} = \begin{bmatrix} \hat{C}_{11} & 0 \\ 0 & \hat{C}_{22} \end{bmatrix} \quad \hat{G} = \begin{bmatrix} \hat{G}_{11} & 0 \\ 0 & \hat{G}_{22} \end{bmatrix} \quad \hat{B} = \begin{bmatrix} \hat{B}_{11} & \hat{B}_{12} \\ \hat{B}_{21} & \hat{B}_{22} \end{bmatrix} \tag{33}$$

the realized circuit is shown in Figure 5. The two thermal nodes are shown on the left side with temperatures $\hat{T}_1$ and $\hat{T}_2$ respectively. The two independent current sources (power sources) are shown on the right side with values g_1 and g_2.

After the realization process, we can easily build different multicore architectures (their thermal circuits) on top of these basic thermal building-block modules in SPICE netlists. All the boundary conditions in terms of equivalent resistances and sources will be added once the architectures are generated.

Although we are realizing matrices into the SPICE netlist, which needs to be reconstructed into matrices in the simulation stage, the overhead is very small since the matrices are reduced and have small sizes. In addition, circuit realization will only be performed for the small number of basic modules, and will only be performed once at the composable model library building stage.

4.6. Internal Node Temperature Retrieval Technique

Model reduction presented in Section 4.4 and the diagonalizing technique introduced in Section 4.5 will destroy the internal state structures and only keep the port behaviors (This is indeed what model reduction tries to achieve). Specifically, the final diagonalized reduced model (31) has the state $\hat{T}$ with k dimensions, while the state of the original finite difference model (8), T, has n dimensions. From the reduced model, we can only observe the port temperatures, but the temperatures of the original finite difference nodes cannot be observed directly.

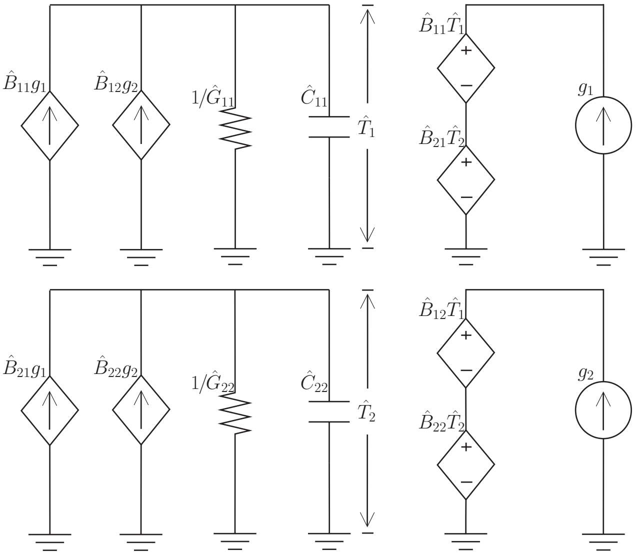


Fig. 5. A simple circuit realization example.

Fortunately, the original internal node information can be approximately recovered from the reduced model. Remembering (19) and (30), we have

$$\bar{T} = V\tilde{T} = VP\hat{T}, \tag{34}$$

where

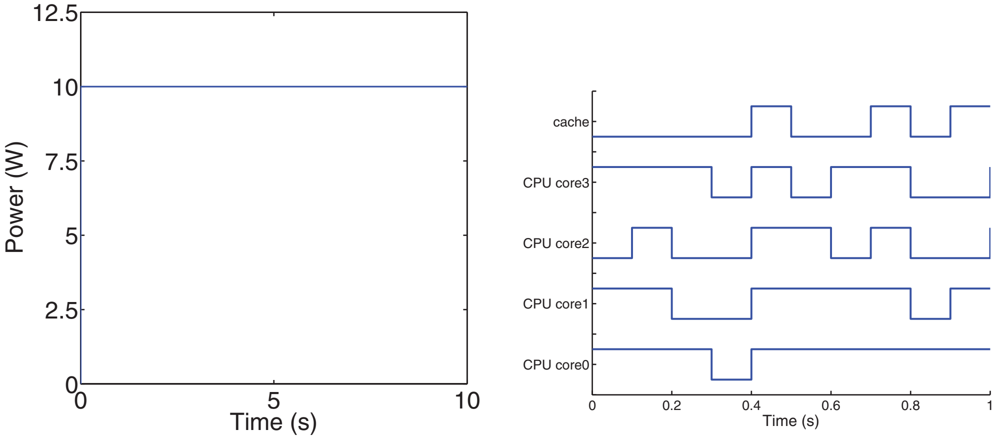
$$\bar{T} \approx T \in \mathbb{R}^{n\times 1} \tag{35}$$

is the original state approximation. As a result, the original state T can be approximately recovered as $\bar{T}$ from the reduced state $\hat{T}$ through formulation (34).

5. EXPERIMENTAL RESULTS

The proposed method, *ThermComp*, has been implemented in MATLAB. First, we build the finite difference models and generate their reduced composable models for a single CPU module and a single cache module using the two-grid discretization based finite difference method and the sampling based model reduction technique. Then, we compose multicore systems using the reduced composable CPU core and cache modules. The original finite difference models of the corresponding multicore systems are built directly (without port merge process, composition and model reduction) for the comparison purpose. Finally, thermal transient simulation is performed using HSPICE on a Linux server with Intel quad-core CPU and 16GB memory to obtain the temperature distribution for both the original and the reduced composite thermal systems.

To build the composable models for CPU core and cache modules, we set up the size of the discretization grid as $32 \times 16 \times 3$ for CPU core module ($8mm \times 4mm \times 0.75mm$) and $64 \times 32 \times 3$ for cache module ($16mm \times 8mm \times 0.75mm$), in order to keep both CPU core and cache modules sharing the same discretization step value $\Delta x = \Delta y = \Delta z = 0.25mm$. This is because the length and width of cache module are twice of the CPU core while the height is the same. Here, we choose the thermal conductivity $\kappa = 149W/(m^{\circ}C)$, material density $\rho = 2300Kg/m^3$ and specific heat $c_p = 700J/(Kg^{\circ}C)$. The lateral and top surfaces are supposed to be adiabatic ($h_i = 0W/m^2K$) and the heat exchanges with



(a) Power input source type 1: step input for steady state analysis, all the CPU and cache modules share the same input.

(b) Power input source type 2: random power traces for transient analysis.

Fig. 6. Power input waveform.

the ambient through the bottom surface ($h_i = 10000W/m^2K$), which is convective to model the heat spreader effects.

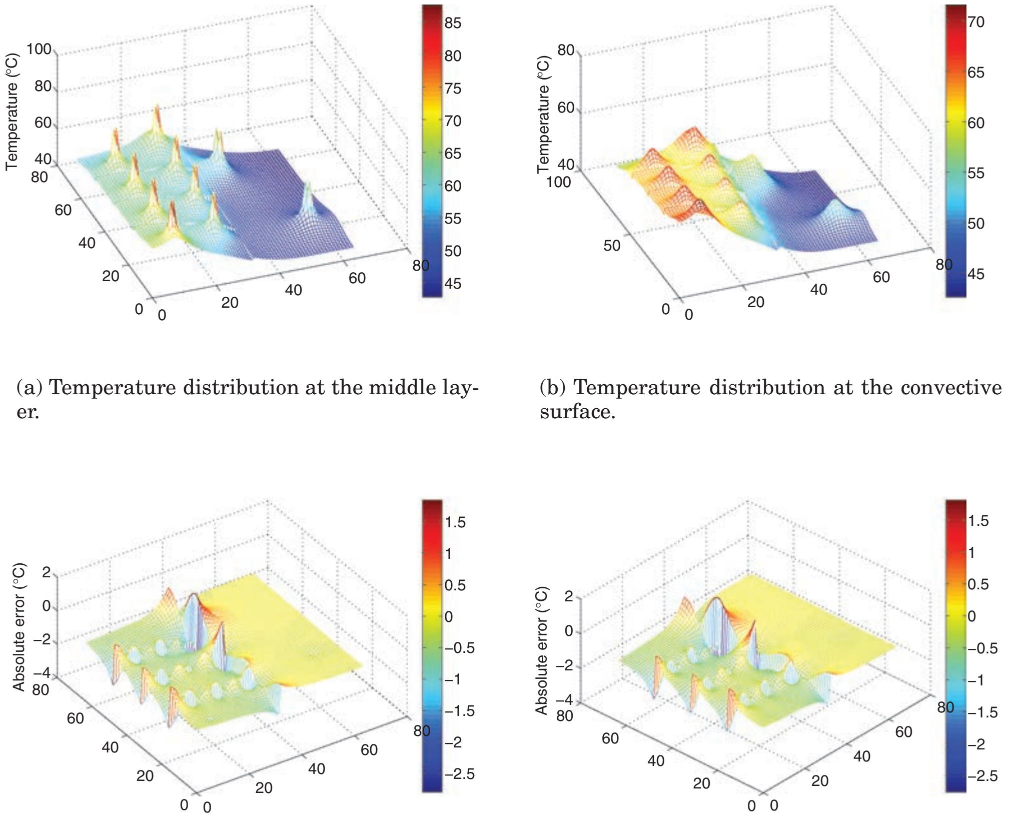
Two types of power dissipation waveforms used in the experiment are shown in Figure 6. The one in (a) is the type 1 total power input in one module, used for steady state analysis. All the basic modules (CPU and cache) share this same total input. In order to test the composable model in more realistic cases, we have also used power traces from SPEC [Henning 2000] benchmarks by running architecture level power simulator Wattch [Brooks et al. 2000]. The cc3 clock-gating power estimation model is used in this experiment. It is known that Wattch cannot handle leakage power well. Since leakage power reversely depends on temperature, in order to accurately estimate leakage power, a power-temperature simulation loop needs to be run until convergence. For simplicity without harming the model verification accuracy, we did not estimate the leakage power in details but simply feed the power trace from Wattch for model verification.

As discussed in Section 4.3, this total power may concentrate in some power centers, distribute across the module uniformly or following a distribution according to the power dissipation model used. In this experiment, we use both uniform model and center model for the power dissipation modeling. For the center model, four power centers are put into a module at its middle layer. We emphasis that *ThermComp* is able to handle all of the three power dissipation models (uniform, center and distribution power models). The waveforms in (b) is the type 2 total power inputs for different modules, used for transient analysis. As shown in the figure, different modules have different power input waveforms.

We have composed a quad-core CPU as shown in Figure 1(b) using the CPU core module and the cache module shown in Figure 1(a).

5.1. Comparison with Finite Difference Simulation

We first apply power source type 1 for the steady state analysis and use the center power dissipation model. There are four power centers in each module and all of them are very close to the boundaries in order to validate the port merge process. Figure 7(a)



(a) Temperature distribution at the middle layer.

(b) Temperature distribution at the convective surface.

(c) Temperature error at the middle layer.

(d) Temperature error at the convective surface.

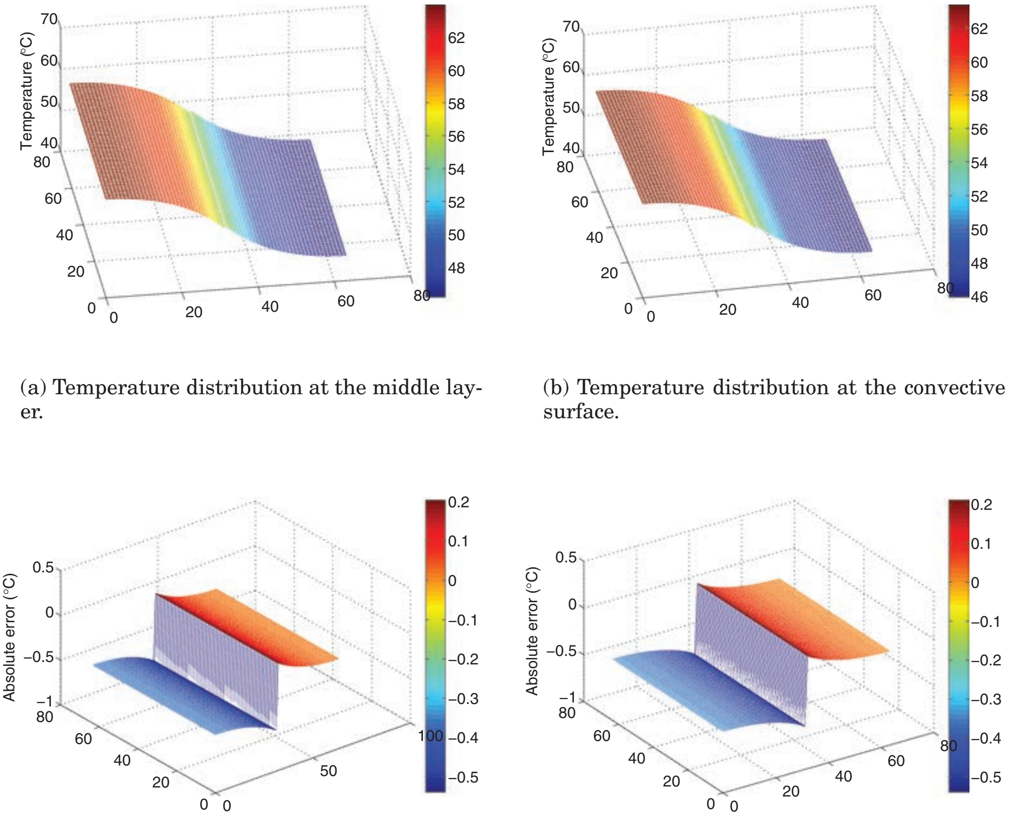
Fig. 7. Composed quad-core microprocessor temperature distribution with type 1 power input and center power dissipation model.

and (b) show the temperature distribution at the middle layer and the convective surface layer (the bottom surface) of the die, respectively. All of the internal nodes can be observed thanks to the internal node retrieval technique used. As we can see, the highest temperatures are on the left hand side and the hot spots appear at the power centers. This is expected as the CPU cores are on the left hand side.

The error plots of *ThermComp* are shown in Figure 7(c) and (d). We can see although the power centers generated large gradients at the boundaries, the largest temperature error is $2.5^{\circ}C$. The errors are almost $0^{\circ}C$ elsewhere other than at the boundaries. This indicates the high accuracy of our composite model.

Next, we change the power dissipation model to the uniform one, i.e., the power dissipation is uniformly distributed within the module. Figure 8(a) and (b) show the temperature distributions at the middle layer and the convective surface. It is observed that the temperature decreases from the CPU modules to the cache module smoothly and the convective surface has a temperature higher than the middle layer. This is because of the uniform power dissipation in the modules.

The errors of *ThermComp* with the uniform power distribution are shown in Figure 8(c) and (d). Since there are relatively small temperature differences on the boundaries, the errors of the port merge process are very small, within $0.5^{\circ}C$ as



(a) Temperature distribution at the middle layer.

(b) Temperature distribution at the convective surface.

(c) Temperature error at the middle layer.

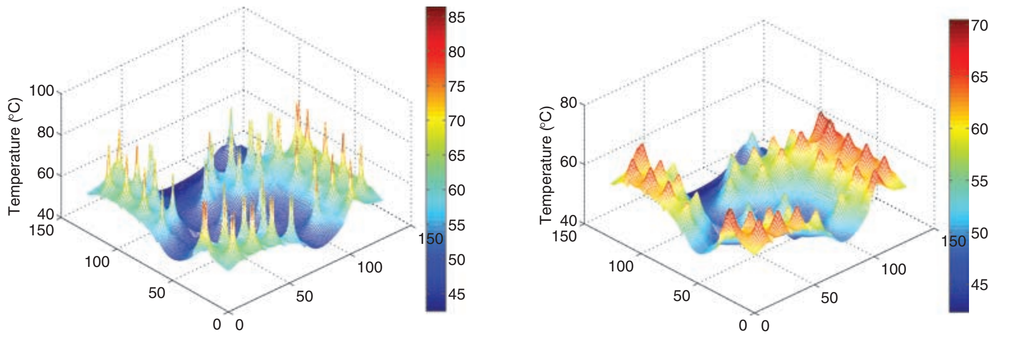
(d) Temperature error at the convective surface.

Fig. 8. Composed quad-core microprocessor temperature distribution with type 1 power input and uniform power dissipation model.

shown in the figures. A nonsmooth transition in the error between two modules (CPU core and cache) can be seem in the figure. This phenomenon is due to the boundary merging. As discussed in Section 4.1, several thermal nodes are merged together and the information of the temperature variations among these boundary thermal nodes are lost. As a trade-off between the simulation accuracy and reduction efficiency, this effect can be minimized by merging less boundary nodes.

We have also composed the 16-core architecture, which is shown in Figure 1(c), using the CPU core and cache in Figure 1(a). Figure 9 shows the temperatures at the middle layer and convective surface given input type 1 with the center power model. The temperature distributions with the uniform power model are shown in Figure 10. The highest temperatures are on the right hand side as we have more CPU modules there. The middle has the lowest temperatures as many cache modules are located close to the center. Similar to the quad-core case, the uniform power distribution model gives a relatively smoother temperature distribution.

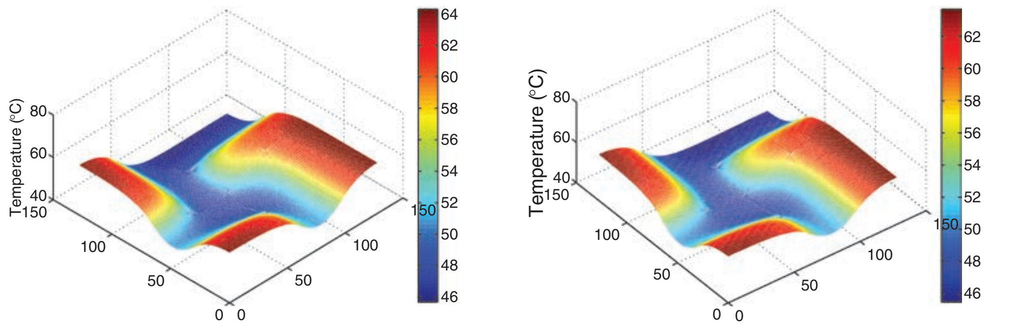
Next, we analyze the dynamic thermal behavior of the 16-core architecture using the power input type 2 shown in Figure 6(b) with the center power model. Figure 11 compares the accuracy between the original finite difference method and the composite model at several randomly picked power centers. The two transient results are almost



(a) Temperature distribution at the middle layer.

(b) Temperature distribution at the convective surface.

Fig. 9. 16-core microprocessor temperature distribution with type 1 power input and center power dissipation model.



(a) Temperature distribution at the middle layer.

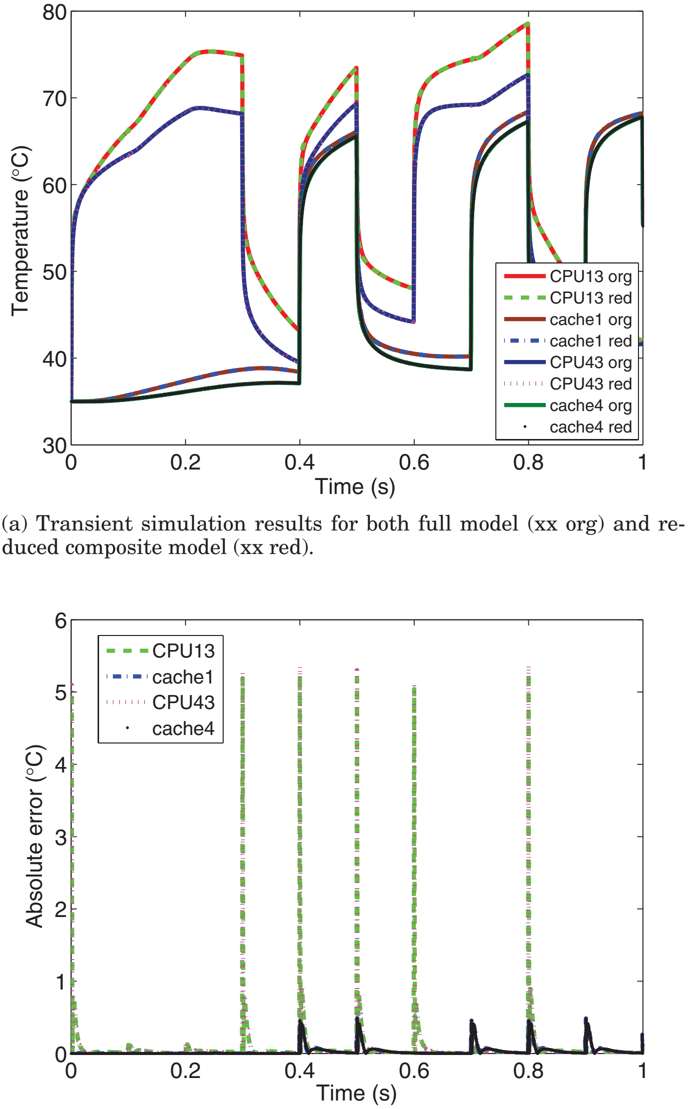
(b) Temperature distribution at the convective surface.

Fig. 10. 16-core microprocessor temperature distribution with type 1 power input and uniform power dissipation model.

the same and the large errors at the fast switch points are actually due to the small errors in the time axis. Since the waveforms at some time points switch very fast (like the ideal step), a tiny timing error will lead to a large temperature difference. But those differences do not represent any meaningful errors in practice.

Another dynamic thermal behavior analysis of the 16-core architecture is performed using a power trace obtained from power estimator Wattch [Brooks et al. 2000] by running the SPEC [Henning 2000] bzip2 benchmark. The bzip2 benchmark power trace, which has more complex transient behavior, is applied to each module using the center power model. Figure 12 shows the transient results from the original finite difference model and the composite model. Similar to results from the previous experiment, the composite model generated accurate transient waveforms.

The CPU time results for different configurations of modules are shown in Table I. In the table, xx *org* means the original models directly built from finite difference method and xx *red* means the reduced systems built from composable models (*ThermComp*). It can be seen that with the reduced composite thermal models, we can achieve about two orders of magnitude speedup for the transient simulation of the multicore systems. In



(a) Transient simulation results for both full model (xx org) and reduced composite model (xx red).

(b) Error plot of the transient simulation.

Fig. 11. Transient simulation accuracy comparison with the full model at some power centers for the 16-core architecture, using power input source type 2.

addition, the reduced composite thermal models lead to much smaller memory footprint than the original models.

5.2. Comparison with HotSpot Program

To further illustrate the advantages of the proposed method, we compared the new modeling technique with existing thermal modeling program HotSpot [Huang et al. 2006]. To perform accuracy comparison between the two programs, we have configured HotSpot to have the same settings as *ThermComp*. HotSpot's heat sink and heat spreader are set to be very thin (10^{-7}m in thickness) in order to neglect

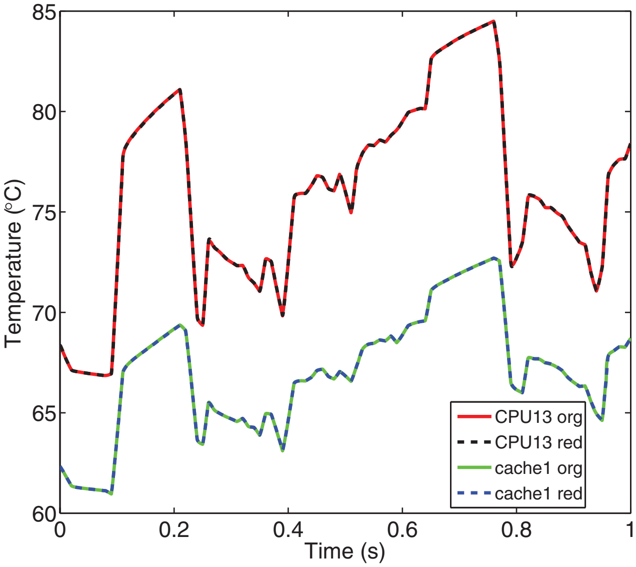


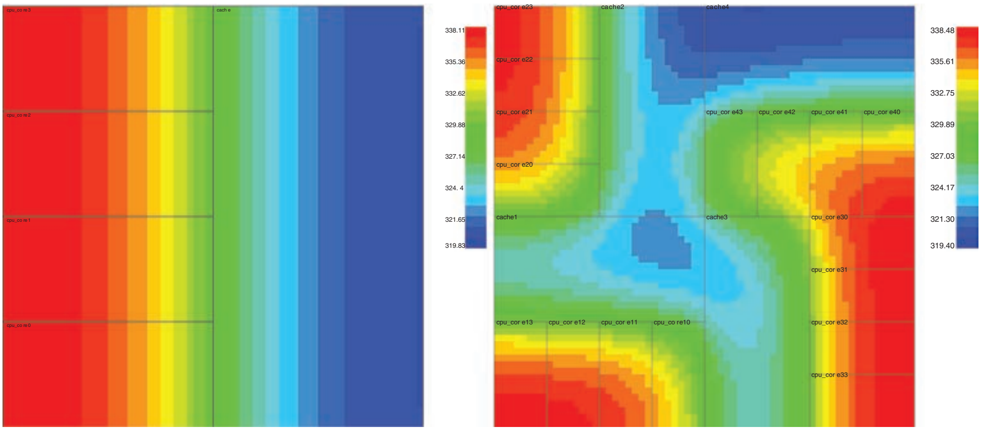
Fig. 12. Transient simulation accuracy comparison with the full model at some power centers for the 16-core architecture, using power input from bzip2 benchmark.

Table I. CPU Time Comparisons between the Original Models (xx *org*) and Reduced Thermal Systems using Composable Models (xx *red*)

1	2	3	4	5	6	7	8
Circuit	#Node	#Elem	Run time (s)		Memory	Speedup	
			Trans	Total	(mb)	Trans	Total
CPU core org	3475	12340	4.7	10.9	32.6		
CPU core red	225	448	0.03	0.05	0.7	157	218
Cache org	13093	48232	31.9	100	30		
Cache red	2145	4288	0.2	3.2	6.2	160	31
2 CPU cores org	6949	24678	9.3	33.4	68		
2 CPU cores red	449	894	0.05	0.1	1.3	186	334
4 CPU cores org	13897	49352	42.3	85.8	28		
4 CPU cores red	897	1784	0.1	0.35	2.6	423	245
Quad-core chip org	26989	97582	167	520	93		
Quad-core chip red	3041	6070	0.48	6.9	8.5	348	75
16-core chip org	107953	390312	392	24146	224		
16-core chip red	12161	24264	15.5	114	28	25	212

their effects. HotSpot [Huang et al. 2006] steady state results of the quad-core and 16-core architectures using the power input in Figure 6(a) are shown in Figure 13. Being a uniform power distribution based method, HotSpot gives very similar results compared to *ThermComp* models with uniform power distribution. The maximum difference between ThermComp and HotSpot is within $1°C$.

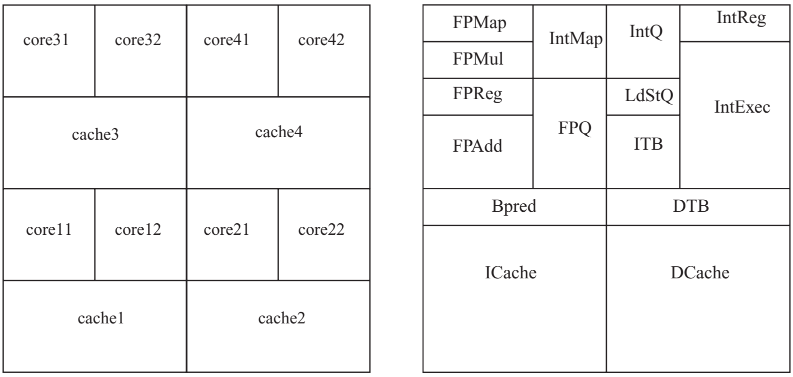
In addition, in order to further verify our method using more realistic CPU architecture with general distributed power model case, we have built an 8-core CPU model with the core structure similar to the alpha ev6 architecture as shown in Figure 14. The power trace of the 8-core CPU is obtained by running Wattch power estimator with SPEC benchmarks. For the composite model, in order to verify the distributed power model, we treat each core as a module, and extract the power distribution by



(a) Temperature distribution of the quad-core microprocessor.

(b) Temperature distribution of the 16-core microprocessor.

Fig. 13. HotSpot steady state results of the quad-core and 16-core microprocessors.



(a) The basic structure of the 8-core alpha chip, composed of 8 cores and 4 caches.

(b) The detailed structure of each core, similar to the alpha ev6 processor.

Fig. 14. The architecture of the 8-core alpha chip.

running different SPEC benchmarks and let the power distribution follow the average power ratio among all the functional blocks. The temperature distribution is obtained by running bizp2 benchmark on the bottom left dual-core structure (core11, core12, and cache1), mcf benchmark on the bottom right dual-core structure (core21, core22, and cache2), swim benchmark on the top left dual-core structure (core31, core32, and cache3), and mgrid benchmark on the top right dual-core structure (core41, core42, and cache4). The steady state results of the HotSpot results on the same 8-core structure is shown in Figure 15. The steady state results from the composite model with the distributed power model are shown in Figure 16. Comparing Figure 15 and Figure 16, except for the errors on the boundaries of the modules, the other parts of the temperature distribution from the composable model matches the HotSpot results very well.

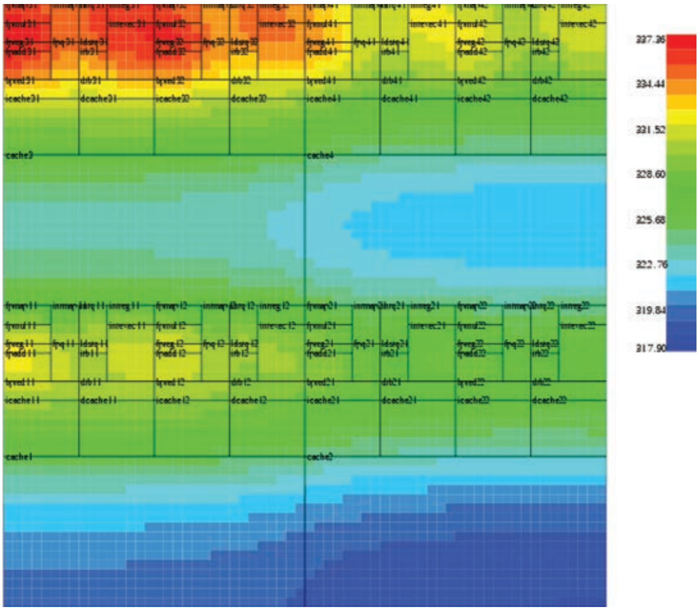
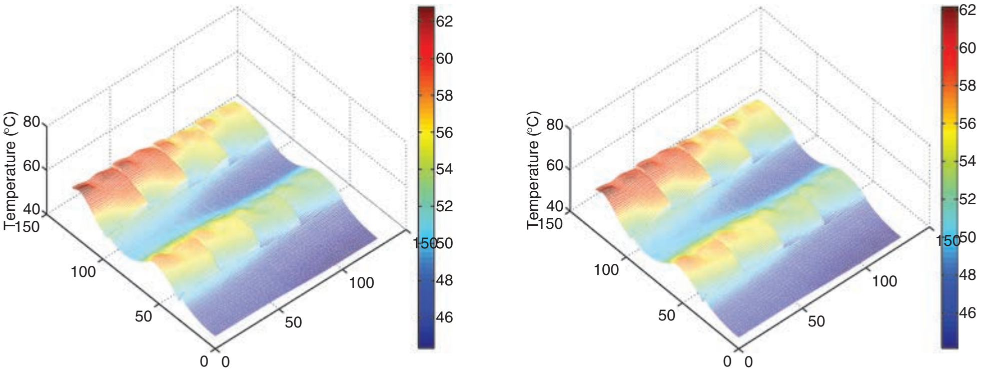


Fig. 15. HotSpot steady state results of the 8-core microprocessor.



(a) Temperature distribution at the middle layer.

(b) Temperature distribution at the surface.

Fig. 16. Steady state results of the 8-core microprocessor with the composable thermal model with the distributed power model.

The CPU time comparison with HotSpot using the power trace from Figure 6(a) is shown in Table II. In the steady state case, we have configured the heat sink and heat spreader to a smaller value in order to neglect their effects in HotSpot. However, doing this will cause the transient simulation hard to converge and result in a very long transient simulation time. In order to make a fair comparison, in the transient simulation case, we have relaxed the thickness of both the heat sink and heat spreader to the default values. Since HotSpot grid mode restricts the grid dimensions to be powers of two, we make HotSpot node number to be smaller than *ThermComp* node number in the case where we cannot make them the same. *ThermComp* is faster than

Table II. CPU Time Comparison between ThermComp and HotSpot

1	2	3	4	5
Circuit	ThermComp		HotSpot	
	node #	total time (s)	node #	total time (s)
Quad-core chip	26989	6.9	16384	86
16-core chip	107953	114	65536	586

HotSpot even with larger node number (higher resolution) as shown in the table. It is also noticed that *ThermComp* achieved less speedup with the larger 16-core chip case than with the smaller Quad-core case. The major reason is on the $\hat{B}$ matrix in the sparsified reduced model (31). In the sparsification process introduced in Section 4.5, $\hat{G}$ and $\hat{C}$ matrices are diagonalized, making the number of non-zeros grow in linear form as the reduced system size k grows. However, $\hat{B}$ matrix remains dense, which means the number of controlled sources will grow quadratically (size of $\hat{B}$ is the product of the reduced system size and the port count ($k \times p$). As a result, the simulation improvement will go down when the systems with more cores are simulated. Although the performance of *ThermComp* is held back by the dense $\hat{B}$ matrix, it is still at least 5 times faster than HotSpot in the largest testing case 16-core chip.

6. CONCLUSION

In this article, we have proposed a new compact thermal modeling technique for architecture level thermal design space exploration for multicore microprocessors. The new approach, called *ThermComp*, builds the models from the first principles by the finite difference method for each basic module and reduces the model complexity by the sampling based model order reduction technique. To improve the reduction efficiency, we try to merge the boundary nodes of modules, which lead to different space discretizations for the whole thermal system. *ThermComp* tries to preserve the accuracy of fine-grained models with the speed of coarse-grained models. The resulting reduced models are then realized into equivalent RC circuits for easy assembling and simulation by circuit level SPICE simulators. Experimental results on a number of multicore microprocessor architectures show that the new approach can easily build accurate thermal circuits from the composable thermal models. The reduced composite models lead to orders of magnitude speedup over the standard finite difference models and are much faster than the HotSpot method with similar accuracy.

REFERENCES

AMD INC. 2006. Multi-core processors—the next evolution in computing. White paper. http://multicore.amd.com.

ANTOULAS, A. C. 2005. *Approximation of Large-Scale Dynamical Systems*. SIAM.

AUGUSTIN, A., MAJ, B., AND KOSTKA, A. 2005. A structure oriented compact thermal model for multiple heat source ASICs. *Mircroelectron. J. 36*, 8, 700–704.

BAI, Z., DEMMEL, J., DONGARRA, J., RUHE, A., AND VAN DER VORST, H. 2000. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM.

BROOKS, D. AND MARTONOSI, M. 2001. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the International Symposium on High-Performance Computer Architecture*. 171–182.

BROOKS, D., TIWARI, V., AND MARTONOSI, M. 2000. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the International Symposium on Computer Architecture*. 83–94.

CHENG, Y.-K., TSAI, C.-H., TENG, C.-C., AND KANG, S.-M. 2000. *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers.

CHRISTIAENS, F., VANDEVELDE, B., BEYNE, E., MERTENS, R., AND BERGHMANS, J. 1998. A generic methodology for deriving compact dynamic thermal models, applied to the PSGA package. *IEEE Trans. Compon. Packag. Manuf. Technol., Part A 21*, 4, 565–576.

Codecasa, L., D'Amore, D., and Maffezzoni, P. 2003. An arnoldi based thermal network reduction method for electro-thermal analysis. *IEEE Trans. Compon. Packag. Manuf. Technol. 26*, 1, 186–192.

Codecasa, L., D'Amore, D., and Maffezzoni, P. 2006. Boundary condition independent compact models of dynamic thermal networks with many heat sources. In *Proceedings of the 10th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronics Systems*. 685–689.

Gerstenmaier, Y. C. and Wachutka, G. 2002. Rigorous model and network for transient thermal problems. *Microelectron. J. 33*, 719–725.

Gunther, S., Binns, F., Carmean, D., and Hall, J. 2001. Managing the impact of increasing microprocessor power consumption. *Intel Tech. J.*

Henning, J. L. 2000. SPEC CPU 2000: Measuring CPU performance in the new millennium. *IEEE Computer 1*, 7, 28–35.

Huang, W., Ghosh, S., Velusamy, S., Sankaranarayanan, K., Skadron, K., and Stan, M. R. 2006. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. VLSI Syst. 14*, 5, 501–513.

Huang, W., Stan, M., Skadron, K., Sankaranarayanan, K., Ghosh, S., and Velusamy, S. 2004. Compact thermal modeling for temperature-aware design. In *Proceedings of the Design Automation Conference.* 878–883.

Intel Corporation. 2006. Intel multi-core processors, making the move to quad-core and beyond. White paper. http://www.intel.com/multi-core.

Iserles, A. 1996. *A First Course in the Numerical Analysis of Differential Equations* 3rd Ed. Cambridge University.

ITRS 2010. International technology roadmap for semiconductors (ITRS), 2010 update. http://public. itrs.net.

Lasance, C., Vinke, H., Rosten, H., and Weiner, K.-L. 1995. A novel approach for the thermal characterization of electronic parts. In *Proceedings of the IEEE 11th Annual Semiconductor Thermal Measurement and Management Symposium*. 1–9.

Li, D., Tan, S. X.-D., Pacheco, E. H., and Tirumala, M. 2008. Parameterized transient thermal behavioral modeling for chip multiprocessors. In *Proceedings of the International Conference on Computer Aided Design*. 611–617.

Li, D., Tan, S. X.-D., Pacheco, E. H., and Tirumala, M. 2009. Architecture-level thermal characterization for multi-core microprocessors. *IEEE Trans. VLSI Syst. 17*, 10, 1495–1507.

Li, Y., Lee, B. C., Brooks, D., Hu, Z., and Skadron, K. 2006. CMP design space exploration subject to physical constraints. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*. 15–26.

Liao, G., Zhu, X., and Bhuyan, L. N. 2011. A new server I/O architecture for high speed networks. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*.

Liao, G., Zhu, X., Larsen, S., Bhuyan, L. N., and Huggahalli, R. 2010. Understanding power efficiency of TCP/IP packet processing over 10GbE. In *Proceedings of the 18th Symposium on High-Performance Interconnects*. 32–39.

Ozisik, M. N. 1994. *Finite Difference Methods in Heat Transfer*. Taylor & Francis, Inc.

Pape, H., Schweitzer, D., Janssen, J. H., Morelli, A., and Villa, C. M. 2004. Thermal transient modeling and experimental validation in the European project PROFIT. *IEEE Trans. Compon. Packag. Manuf. Technol. 27*, 3, 530–538.

Pedram, M. and Nazarian, S. 2006. Thermal modeling, analysis, and management in VLSI circuits: Principles and methods. *Proc. IEEE 94*, 8, 1487–1501.

Phillips, J. R. and Silveira, L. M. 2005. Poor man's TBR: a simple model reduction scheme. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 24*, 1, 43–55.

Rencz, M., Farkas, G., Székely, V., Poppe, A., and Courtois, B. 2003. Boundary condition independent dynamic compact models of packages and heat sinks from thermal transient measurements. In *Proceedings of the 5th Electronics Packaging Technology Conference*. 479–484.

Skadron, K., Stan, M. R., Huang, W., Velusamy, S., Sankaranarayanan, K., and Tarjan, D. 2003. Temperature-aware microarchitecture. In *Proceedings of the International Symposium on Computer Architecture*. 2–13.

Tan, S. X.-D. and He, L. 2007. *Advanced Model Order Reduction Techniques in VLSI Design*. Cambridge University Press.

Wang, H., Tan, S. X.-D., Liao, G., Quintanilla, R., and Gupta, A. 2011. Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management. In *Proceedings of the International Conference on Computer Aided Design*.

WANG, H., TAN, S. X.-D., AND RAKIB, R. 2012. Compact modeling of interconnect circuits over wide frequency band by adaptive complex-valued sampling method. *ACM Trans. Des. Autom. Electron. Syst. 17*, 1, 5:1–5:22.

WANG, T. Y. AND CHEN, C. C. 2002. 3-D thermal-ADI: a linear-time chip level transient thermal simulator. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 21*, 12, 1434–1445.

WILLCOX, K. AND PERAIRE, J. 2002. Balanced model reduction via the proper orthogonal decomposition. *AIAA J.*

YANG, Y., GU, Z. P., ZHU, C., DICK, R. P., AND SHANG, L. 2007. ISAC: Integrated space and time adaptive chip-package thermal analysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 16*, 1, 86–99.

Received June 2011; revised March 2012, August 2012; accepted October 2012