IC Thermal Analyzer for Versatile 3-D Structures Using Multigrid Preconditioned Krylov Methods

DOI:
10.1145/2966986.2967045

Document Version
Accepted author manuscript

Published in:
2016 International Conference On Computer Aided Design

IC Thermal Analyzer for Versatile 3-D Structures Using Multigrid Preconditioned Krylov Methods

ABSTRACT

Thermal analysis is crucial for determining the propagation of heat and to track the formation of hot spots in advanced integrated circuit technologies. At the core of the thermal analysis for the integrated circuits is the numerical solution of the heat equation. Prior academic thermal analysis tools typically compute temperature by applying finite difference methods on uniform grids with time integration methods having fixed time step size. Additionally, the linear systems arising from the discretized heat equation are solved using direct methods based on matrix factorizations. Direct methods, however, do not scale well as the problem size increases. Moreover, most of the tools support only 2-D or a limited number of 3-D technologies. To address these issues, this paper presents a novel thermal analyzer with the ability to model both 2-D and 3-D circuit technologies. The analyzer solves the heat equation using the finite element method for the spatial discretization coupled with implicit time integration methods for advancing the solution in time. It also offers fully adaptive spatio-temporal refinement features for improved accuracy and computational efficiency. The resulting linear systems are solved by a multigrid preconditioned Krylov subspace iterative method, which gives superior performance for 3-D transient analyses. The analyzer is shown to accurately capture the propagation of heat in both the horizontal and vertical directions of integrated systems.

Keywords

Transient thermal analysis, Integrated circuits, Finite element method, Adaptive spatio-temporal refinement, Multigrid, Krylov solvers

1. INTRODUCTION

For advanced CMOS technologies below the 90 *nm* node, heat issues, incurred due to higher power, lead to performance degradation and excessive leakage currents [1]. Device technologies, such as FinFET and SOI, have been introduced to suppress leakage currents in order to maintain silicon technology scaling. However, they can be more susceptible to thermal issues due to self-heating and the low thermal conductivity of their constituent materials [2]. Designs of "post-Dennardian" scaling also face challenges of exponentially increasing power density [3, 4]. Emerging 3-D technologies (i.e., multi-tier systems with vertical interconnections) also suffer from thermal issues since 3-D ICs have complex heat dissipation paths and more active regions than a single tier in a package [1, 5, 6].

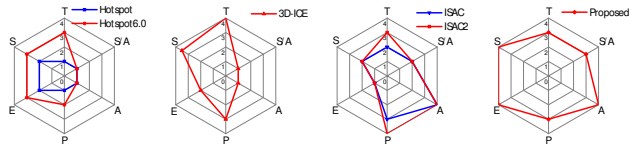
To deal with these challenges, early stage thermal characterization of ICs, (e.g., microarchitectures of processors) and the corresponding physical structures are required to provide a starting point for efficient thermal design eliminating thermal hazards. Fine-grain, or high resolution 3-D transient thermal analysis tools are promising for the simulation of modern and future ICs. Existing tools, e.g., [7, 8, 9, 10, 11], can be used to detect the formation of hot spots both in steady-state and transient regimes. These tools use compact models to describe heat transfer and are efficient only for a narrow class of problems, namely, simple small-scale problems where the spatial and time discretization remain fixed, resulting in linear systems that do not change at each time step. Additionally, they cannot accurately model various 3-D structures of future ICs. Therefore, this paper introduces a versatile thermal analyzer, which can accurately capture the flow of heat in the various 3-D structures of an IC.

The contributions of this paper to the problem of thermal analysis for 2-D and 3-D circuits are:

- the ability to model complex 2-D and 3-D circuit geometries,
- the fully automated mesh generation of such circuit geometries from a corresponding floorplan description,
- the use of second order accurate numerical schemes,
- the solution of the resulting linear systems using advanced preconditioned iterative methods,
- and the support of fully adaptive spatio-temporal refinement.

The proposed thermal analyzer uses the finite element method (FEM) [12] in conjunction with an implicit time stepping scheme to solve the heat equation. The FEM is a standard numerical technique that is well-suited to accurately capture the flow of heat in the complex geometries of an IC. The computation of transient thermal profiles with sufficient granularity requires the solution of a large, sparse linear system at each time step, which is then solved using multigrid preconditioned Krylov subspace methods [12, 13]. Such preconditioned iterative solvers offer faster solution times and use less memory [14] than sparse direct methods based on matrix factorizations [9]. The iterative solver also offers the potential for linear scaling with problem size.

Additionally, the proposed thermal analyzer offers fully adaptive spatio-temporal refinement features. This added flexibility gives our method computational advantages over other analyzers as discussed in Section 5.2. The main benefit



(a) Hotspot (b) 3D-ICE (c) ISAC (d) This work

Figure 1: Comparison of thermal analyzers.

of adaptive spatial refinement is that smaller problems can be solved without a loss of accuracy. In addition, with adaptive time stepping methods, the simulator can apply larger time steps during the periods of low (or no) power dissipation and smaller time steps to accurately capture the transient thermal effects during rapid changes in the power and temperature profile. This adaptation does not require user intervention and is inherently performed by our technique.

The rest of the paper is organized as follows. Section 2 discusses related works and the motivation for the proposed analyzer. Section 3 introduces the thermal model and FEM for computing transient thermal profiles of a given 2-D or 3-D IC. Section 4 describes the implementation details, which include a description of the iterative solver and preconditioner, and the adaptive spatio-temporal features. Section 5 presents the results of the numerical experiments and a discussion of the advantages of the proposed method. Section 6 offers some concluding remarks.

2. RELATED WORK AND MOTIVATION

Thermal analysis tools for integrated circuits have been previously introduced, for instance HotSpot 6.0 [9], 3D-ICE [10], and ISAC(2) [7]([15]). In ISAC, the heat equation is discretized using finite difference methods. Hotspot and 3D-ICE both invoke the analogy between heat flow, driven by temperature differences, and electrical current flow driven by voltage differences. The electro-thermal duality supports compact thermal models based on difference formulas. Such a spatial discretization can also be viewed as a finite difference method defined by the floorplan of the circuit. In each of these tools, the transient temperature profiles are computed using a time integration method. This process requires solving a linear system of equations at each time step. These tools offer fast transient thermal analysis, but are effective only for a limited range of problems, specifically, simple circuit geometries that lead to small-scale linear systems that do not change at each time step, i.e., the spatial and time discretizations are fixed. Practically, these tools can model 2.5-D geometries, where each layer of material cannot be freely discretized in the vertical direction. Such simplified models however, limit the applicability of these tools, where large-scale systems and 3-D systems are analyzed.

In any thermal analysis tool, both speed and accuracy of the computations are crucial. Furthermore, the thermal analysis of 3-D circuits must accurately capture and model the various substructures of the circuit, for example through-silicon vias (TSVs), which behave as heat conduits within the circuit stack. Two main ways to improve computational speed while not sacrificing the accuracy of computations, are adaptive mesh refinement and adaptive time stepping methods [7, 15]. The benefit of adaptive over uniform spatial refinement is that the same level of accuracy for the solution can be obtained at a lower computational cost. Similarly, adaptive time integration methods allow for fewer (and larger) time steps, without sacrificing accuracy.

T: Computation time		
1: Slow	2: Moderate	3: Fast
Parallel*		
S/A: Solvers and Accuracy		
Linear model*, Nonlinear model*		
Second order schemes*, Iterative solvers*		
A: Adaptivity		
Uniform spatial grid*, Nonuniform spatial grid*		
Adaptive spatial refinement*, Adaptive temporal refinement*		
P: Problem complexity		
1: 2-D	2: 2.5-D	3: 3-D
Atomic*		
E: Ease of use		
1: Hard	2: Medium	3: Easy
Visualization*		
S: Supported structure of ICs		
Planar IC*, Stacking tiers*		
Substructure (TSV, microfluid)*, Package/Board Level*		

Table 1: Definitions of notation used

A visual comparison of the computational features of the proposed analyzer against other available thermal analysis tools is shown in Fig. 1. The figures are generated based on the point system outlined in Table 1. Each tool is compared based on the following set of attributes: computation time, the solver used and accuracy of the tool, adaptivity features, the handled problem complexity, ease of use, and the circuit structures that are supported. For each property a tool possesses for a given attribute, a point is awarded.

Note that the proposed analyzer covers the largest area of the hexagon, and therefore exhibits the broadest set of computational capabilities. The only other tool that shares the same range of fully adaptive spatio-temporal refinement features is ISAC. Hotspot only offers adaptive time stepping on uniformly refined spatial grids. However, both ISAC and Hotspot employ explicit time stepping schemes and as a result their step size selection is constrained by the spatial discretization. This gives our proposed analyzer an advantage since we employ unconditionally stable implicit time stepping schemes, which do not impose any constraints on the time step size [16], apart from.

Additionally, the proposed analyzer employs preconditioned iterative methods for the solution of linear systems arising from the discretized problem. Both Hotspot and 3D-ICE use the Super-LU solver, a direct method that scales poorly with problem size. ISAC uses geometric multigrid solvers. The proposed analyzer uses state-of-the-art multigrid preconditioned iterative methods which converge rapidly, yielding improved solution times and iteration counts than the previously mentioned solvers.

More importantly, the proposed analyzer is the first academic tool with the capability to realistically model both 2-D and 3-D structures, such as multiple tiers, TSVs, and finned heat sinks. In this way, our analyzer offers a more comprehensive set of computational features with a more robust solution methodology, which allows for more complex and advanced thermal analyses to be performed.

3. THERMAL MODEL

Let $T(\mathbf{x}, t)$ denote the temperature (in [K]) at time t and spatial point $\mathbf{x} = (x, y, z)$ in an IC. The temporal and spatial evolution of the temperature are governed by

the heat equation with appropriate boundary and initial conditions, specifically,

$$(\rho c_p)\frac{\partial T(\mathbf{x},t)}{\partial t} - \nabla \cdot (\kappa \nabla T(\mathbf{x},t)) = f(\mathbf{x},t) \text{ in } \Omega, \quad (1a)$$

$$\kappa \nabla T \cdot \mathbf{n} = \eta(T_a - T) \text{ on } \partial\Omega, \quad (1b)$$

$$T(\mathbf{x},0) = T_a. \quad (1c)$$

The physical domain of the circuit is denoted by Ω and its boundary by $\partial\Omega$. The function $\nabla T = (T_x, T_y, T_z)$ is the temperature gradient and $\nabla \cdot T = T_x + T_y + T_z$ denotes the divergence operator, where, for example, $T_x = \frac{\partial T}{\partial x}$ denotes the partial derivative with respect to x. The physical parameters are the density ρ $[kg/m^3]$, specific heat c_p $[J/kgK]$, thermal conductivity κ $[W/mK]$, and thermal transmissivity η $[W/m^2K]$ at the boundaries. The function $f(\mathbf{x},t)$ $[W/m^3]$ is the power density dissipated by the active layer of the system. The Robin boundary condition described by (1b) represents Newton's law of cooling, i.e., the heat flux at the boundary is proportional (with constant heat transfer coefficient η) to the difference between the ambient temperature T_a and the temperature at the wall $\partial\Omega$. In our model, the Robin boundary conditions are applied to the top and bottom of the circuit. The four lateral walls are considered adiabatic, so that there is zero flux ($\eta = 0$) at these boundary faces, although the proposed method can handle Robin conditions over the entire boundary $\partial\Omega$. Furthermore, in this paper, we assume the parameters ρc_p, κ, and η are fixed with respect to temperature. The values of ρc_p and κ are assumed, however, to be different specific to the disparate floorplan components of the circuit, including, for example, TSVs and finned heat sinks.

The heat equation (1) is solved numerically using the FEM [12]. The FEM first partitions the domain into a union of non-overlapping smaller elements $\Omega = \cup_k \Omega_k$. The elements Ω_k are typically tetra or hexahedra with a characteristic length h called the mesh width. The finite element solution is computed by interpolating the discrete nodal values at the vertices of the partitioned domain. Specifically, the solution has the form $T_h(\mathbf{x}) = \sum_{j=1}^{n} T_j \phi_j(\mathbf{x})$, where $T_j \approx T(\mathbf{x}_j)$ are the n unknown nodal values, and $\phi_j(\mathbf{x})$ is a global Lagrangian basis function that equals 1 at node $\mathbf{x}_j$ and zero at every other node.

The n unknown nodal values $\{T_j\}_{j=1}^{n}$, also called degrees of freedom (DOF), can be collected into a column vector $\boldsymbol{\tau}_h = [T_1, \ldots, T_n]^T$. This vector of unknowns is determined by solving a set of n linear equations in n unknowns. For the steady state of (1a) (i.e., setting $\frac{\partial T}{\partial t} = 0$), this set of equations has the form

$$K\boldsymbol{\tau}_h = \mathbf{f}_h. \quad (2)$$

The size of the coefficient matrix $K \in \mathbb{R}^{n \times n}$ grows as the mesh is refined ($n \to \infty$ as $h \to 0$). The entries of the matrix K and the vector $\mathbf{f}_h$ for the steady state of (1a) are

$$K_{ij} = \int_\Omega \kappa \nabla \phi_j \cdot \nabla \phi_i \, d\mathbf{x} + \int_{\partial\Omega} \eta \phi_j \phi_i \, dS, \quad (3)$$

$$\mathbf{f}_{h,i} = \int_\Omega f \phi_i \, d\mathbf{x} + T_a \int_{\partial\Omega} \eta \phi_j \phi_i \, dS. \quad (4)$$

Note that by the definition of the basis functions the matrix K is sparse, since $K_{ij} \neq 0$ if and only if node i is connected to node j. The matrix K and vector f are assembled by looping over the elements and computing these integrals via appropriate quadrature rules.

For transient problems ($\frac{\partial T}{\partial t} \neq 0$), after discretizing in space using the FEM, the heat equation (1) becomes a system of n initial value problems (IVPs) of the form

$$M\dot{\boldsymbol{\tau}}_h + K\boldsymbol{\tau}_h = \mathbf{f}_h, \quad (5)$$

where M is a mass matrix with entries $M_{ij} = \int_\Omega \phi_j \phi_i \, d\mathbf{x}$.

To solve this system of IVPs the time derivative can be approximated, for instance, by a first or second order backward difference formula (BDF method). For further details on various time integration methods see, e.g., [16].

Regardless of the time stepping scheme, a large, sparse linear system must be solved at each time step.

Higher order methods, such as a second order BDF method, can take larger time steps than the first order method with the same level of temporal accuracy and appreciable computational savings in time. We demonstrate the computational gains of using this higher order formula in Section 5.

The repeated assembly and solution of these linear systems accounts for the majority of the computational effort in transient thermal analysis. Strategies to address this issue have been considered before, for example in [11], where Alternating Direction Implicit (ADI) methods are used. If the resulting linear system does not change between time steps, i.e., there is no spatial refinement or adaptive time stepping, a single LU factorization of the system matrix can be computed at the first step, and then at subsequent time steps the solves are performed using backward and forward substitution with the factors. This is how transient simulations in Hotspot are computed and is one of the reasons for the fast solve times [9]. However, this is also the primary limitation of that strategy. In order to account for all the spatio-temporal changes there must be a sufficiently refined spatial grid and small time step size throughout the entire simulation, which can be a severe restriction and cause excessive computational overhead and accuracy issues.

Alternatively, with adaptive spatio-temporal features, the resulting linear systems must be solved at each time step. However, the size of the resulting linear systems and the number of time steps, (i.e., the number of repeated linear solves) are typically much smaller that in Hotspot [9] leading to potentially significant computational savings. In addition, such an approach systematically supports more complex structures that better fit the physical structure of planar or 3-D integrated systems (cf. Section 5).

4. IMPLEMENTATION DETAILS

This section describes the implementation details of the thermal analyzer. In Section 4.1 the tool flow is presented. In Section 4.2, a description of the hierarchical XML file which encodes the geometry and thermal properties of the IC is provided. In Section 4.3, the FEM implementation used in the analyzer is presented and in Section 4.4 the corresponding linear solver is discussed. In Sections 4.5 and 4.6, the adaptive spatio-temporal refinement features of the method are described.

4.1 Tool Flow

The proposed thermal analyzer has five main stages in the tool chain: structure definition, mesh generation, mesh optimization, thermal simulation by the FEM solver, and

visualization. In the structure definition stage, the users can describe the circuit geometry in an .xml file, which has a hierarchical structure similar to the design hierarchy of the IC. This hierarchical structure allows the user to provide detailed information needed for the simulation, or to select the required granularity level for the simulation. In the mesh generation stage, the mesh file, which discretizes the computational domain for the thermal analysis, is generated using the well-known open-source mesh tool GMSH [17]. However, the mesh generated by GMSH is not appropriate for the IC structure due to duplicated definition of physical volumes. Thus, in the mesh optimization stage, we developed a new tool to surpass these limitations, which removes the duplicate definitions of the physical volumes generated by GMSH and addresses floating point rounding problems for the nodal coordinates of the mesh.

The created mesh is an input to the FEM solver in the thermal simulation stage. The power density source function is provided from a power trace file. The power trace file is generated by architecture simulators coupled with power trace converters, or power estimators for ICs, such as PrimetimePX. In the visualization stage, all the results are plotted in Paraview [18], an open-source visualization tool, which allows the users to view cross-sectional views of the IC to observe the full 3-D thermal flow simulated by the proposed analyzer.

4.2 Versatile Structures in XML

The hierarchical structure of the IC is encoded in an XML file, which gives the added benefit of describing the subdomain structures of the circuit. We have extended the open source XML parser [19] such that the physical traits and thermal properties of the disparate components of the circuit are modeled to automatically generate the computational mesh and initialize the FEM solver for thermal simulations.

The proposed analyzer differs from previous academic and open-source tools, in that the mesh generation process is completely automated and able to represent advanced structures such as TSVs and microbumps. It natively supports 3-D structures by defining multiple-tiers in the vertical direction. The analyzer also has the ability to support more complicated structures, such as heat sinks with fins. Furthermore, users have the ability to create their XML files defining the physical dimensions and thermal properties of their own circuit and then generate the corresponding computational domain. Further details relating to the usage of the XML structure are provided in the Appendix.

4.3 FEM Implementation

The core of the thermal simulator is implemented in C++ using functionalities of the finite element library OOMPH-Lib (Object Oriented Multi-Physics Library), developed at the University of Manchester [20]. Several new features are implemented to support the capabilities of the proposed thermal analyzer. The physical domains can be discretized using either first order bi-linear elements or second order bi-quadratic elements. The time integration methods employed by the analyzer are first and second order implicit backward differentiation formulas. In experiments, the asymptotic order of the spatial discretization and the time integration method are matched. The computations with a first order scheme, i.e., with bi-linear elements and a first order BDF method, offer slightly faster simulation times but at the cost of lower accuracy. Thermal analysis for complex structures, however, requires second order methods for satisfying accuracy as shown in Section 5.1. Though the implementation of the thermal analysis uses a previously developed FEM library, a number of new functionalities and adaptations are made. In particular, the capability to read in and supply power density values from user supplied power trace files is new as well as the implementation of the Robin boundary conditions (1b).

4.4 Linear Solver

The software library platform used to build the simulator is designed to solve more complicated nonlinear problems and employs Newton's method to solve the nonlinear systems of equations arising from the finite element discretizations. In our case, the problem is linear and the Newton iteration converges in one step. Though it may seem counterintuitive to solve a linear problem using a nonlinear solver, in reality the problem parameters are temperature dependent, i.e., $c_p = c_p(T)$, $\kappa = \kappa(T)$, making the proposed model in (1) nonlinear. Although this kind of model is beyond the scope of this paper and is left as future work, the built in nonlinear functionality allows the proposed methodology to be extended to cope with the more physically relevant cases of temperature-dependent thermal parameters and large temperature differences in 3-D.

The resulting linear systems, i.e., (2) for steady-state analysis, or the time discretized version of (5) for transient analysis, are large, sparse and symmetric positive definite (SPD). Therefore, preconditioned Krylov subspace iterative methods are used as the solver. Specifically, the Krylov solver is the conjugate gradient (CG) method [21] with an algebraic multigrid (AMG) preconditioner [22]. The preconditioned CG method is the solver of choice for SPD linear systems. The method only requires computationally inexpensive sparse matrix-vector products. Moreover, the CG method is based on a three-term recurrence to create the orthogonal Krylov basis, i.e., only three vectors need to be stored at each iteration of the algorithm, giving the CG algorithm linear storage costs. The fixed storage costs of the conjugate gradient method make this technique more suitable for solving the resulting linear systems than sparse direct methods based on matrix factorizations. To improve the convergence of the CG method, it is preconditioned with the AMG method with standard Ruge-Stüben coarsening [23]. These solves are implemented by calling the software library hypre [24].

4.5 Adaptive Mesh Refinement

Temperatures can vary significantly across the area of a circuit. To accurately model features of the thermal profile, the computational grid must be sufficiently refined locally. For uniformly refined meshes, achieving this level of accuracy requires globally small mesh sizes, resulting in unnecessarily large linear systems, and longer solve times. To minimize computational effort without losing accuracy, the mesh can be adaptively refined only where the temperature varies significantly, i.e., where there are large temperature gradients. The elements to be refined are determined from computed error estimates depending on the temperature gradient.

In the proposed method, these estimates are computed by the Z2 error estimator [25]. The idea behind this estimator is to compute a more accurate representation of the heat flux (or gradient) of the finite element solution, i.e., $\nabla\tilde{T}_h$. The more accurate flux is computed by projecting the computed finite element flux function, $\nabla T_h = \sum_{j=1}^{n} T_j \nabla\phi_j(\mathbf{x})$, onto a set of continuous basis functions defined on small patches of the domain. The mesh is refined if the difference in the fluxes is greater than a user prescribed tolerance ε_{max}, i.e., $|\nabla\tilde{T}_h - \nabla T_h| > \varepsilon_{max}$. In addition, already refined elements can be merged (unrefined) when the error estimate satisfies $|\nabla\tilde{T}_h - \nabla T_h| < \varepsilon_{min}$, where ε_{min} is the user prescribed minimum error tolerance. This functionality is extremely useful in maintaining accuracy of the solution while keeping a moderate size of the linear systems.

Note that when spatial adaptivity is used, more than one linear system is solved at each time step. This situation is because the fully adaptive spatio-temporal method first advances the solution to the next time step, i.e., $T_H^k \rightarrow T_H^{k+1}$, where H denotes the mesh width on the initial coarser grid. At this point, the error estimate for T_H^{k+1} is computed. If the refinement of the initial coarse grid solution T_H^{k+1} is warranted by this estimate, the grid is refined, otherwise this solution is accepted and the simulation proceeds with T_H^{k+1}. If the grid is refined, the problem needs to be reassembled and solved to obtain the solution T_h^{k+1}, where T_h now denotes the solution on the adaptively refined grid. This procedure is then repeated for all time steps. This reassembly moderately increases the computational cost per time step, but offers flexibility of finding the optimal spatial discretization which produces the solution of the problem with a prescribed level of accuracy.

4.6 Adaptive Time Stepping

Another important technique for increasing the computational efficiency of transient thermal simulations is adaptive time stepping. In these methods, the time step size Δt is either increased or decreased based on the local truncation error estimate of the time integrator. Let $\mathbf{e}$ denote the local truncation error (LTE) [16] and ε_t a prescribed LTE tolerance. Then, a standard estimate for computing the adaptive time step is

$$\Delta t_{new} = \left(\frac{\varepsilon_t}{\|\mathbf{e}\|_2}\right)^{1/(\alpha+1)} \Delta t, \tag{6}$$

where α is the order of the time integration method and $\|\mathbf{e}\|_2 = \sqrt{\frac{1}{n}(\sum_i^n e_i^2)}$ is the standard Euclidean norm.

5. RESULTS AND DISCUSSION

In this section, the features of the proposed thermal analyzer are demonstrated in several numerical experiments. The first experiment gives an error analysis and run-time comparison between direct and iterative solvers as the mesh is uniformly refined for an illustrative 3-D problem. This example demonstrates the faster solution times possible using iterative methods and shows the gain in accuracy possible, while solving smaller linear systems, using second order methods.

The second set of experiments demonstrates the performance of the analyzer for realistic 3-D circuit geometries that cannot be tackled or solved accurately by the existing solvers. For these test problems, the various features of the analyzer, i.e., the adaptive spatio-temporal refinement capabilities are shown. All of the experiments are performed on an Intel i7 4790 processor with 32 GB DRAM and the CentOS 7 operating system. The computed thermal profiles are visualized in the open-source software Paraview [18].

5.1 Numerical Experiments

For the error and run-time analysis, the heat equation (1a) is solved with $(\rho c_p) = \kappa = 1$ on a unit cube. To compute the error, the solution is assumed to have the form

$$T(x,y,z,t) = \alpha e^{-\beta t} e^{-((x-1/2)^2+(y-1/2)^2+(z-1/2)^2)/\sigma}, \tag{7}$$

with $(\alpha,\beta,\sigma) = (10, 0.01, 0.05)$. For this problem, the temperature at the boundary is fixed (as opposed to flux boundary conditions). The heat equation (1a) is solved using the procedure outlined in Section 3 for five time steps with $\Delta t = 0.001$ seconds. The power source function is given by $f(\mathbf{x},t) = \frac{\partial T(\mathbf{x},t)}{\partial t} - \nabla\cdot(\nabla T(\mathbf{x},t))$. The error, $e_h = T - T_h$ is then computed for a sequence of uniformly refined grids.

	dofs	h	$t = 0.001$			$t = 0.005$			Setup (s)
			error	iter	time (s)	error	iter	time (s)	
D	29791	2^{-5}	0.00886943		18.9	0.00793845		0.060	1.29
	250047	2^{-6}	0.00222486		2573	0.00193098		1.62	10.3
I	29791	2^{-5}	0.00886943	2	0.19	0.00770112	2	0.19	1.68
	250047	2^{-6}	0.00222486	3	1.28	0.00193098	3	1.27	11.32

Table 2: Uniform refinement error and run-time analysis for 1st order scheme.

	dofs	h	$t = 0.001$			$t = 0.005$			Setup (s)
			error	iter	time (s)	error	iter	time (s)	
D	250047	2^{-5}	0.000129		5533	0.000130		3.0	50
	2048383	2^{-6}	-		-	-		-	-
I	250047	2^{-5}	0.000129	3	3.2	0.000130	2	3.0	53.7
	2048383	2^{-6}	1.62968e-5	3	28	1.62967e-5	3	28	436

Table 3: Uniform refinement error and run-time analysis for 2nd order scheme.

For this problem, both first and second order schemes are compared. The results of the error and run-time analysis as the mesh is refined uniformly for first and second order methods are given in Tables 2 and 3, respectively.

In these tables, the setup time for both methods, as well as the solution times at the first and final time steps are reported. The intermediate solution times for both methods are similar to their respective final solution times. For iterative solvers the number of iterations to converge to a relative residual tolerance of 10^{-6} is given. Note that the solution time at the first time step, where the LU factorization is computed using a direct method, is orders of magnitude larger than for iterative methods. In particular, observe that for first order schemes, the linear solve at the first time step for a linear system with 250,047 unknowns takes 1.28 seconds to solve using the proposed preconditioned iterative solver and over 2500 seconds using a direct method. Note that for second order methods this result is further exaggerated. The linear system with over 2 million degrees of freedom takes 28 seconds to solve using preconditioned iterative methods, whereas the direct solver is unable to even compute the LU factorization. This clearly illustrates the advantage of using iterative solvers, especially for linear systems arising from 3-D problems.

Furthermore, note that as the mesh is refined for both first and second order methods, the subsequent solves using the upper and lower triangular factors at each time step are

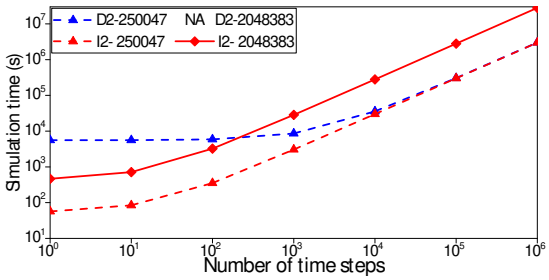


Figure 2: Numerical results.

of the same order as the iterative solvers. Based on these results, in Fig. 2, the extrapolated total simulation time as the number of time steps increases is shown. The red and blue lines denote, respectively, the simulation times for the direct and iterative solvers. It is important to observe that as the problems are refined, the blue lines are below the red lines, i.e., the iterative methods are faster. Moreover, due to the constant solve time per time step, iterative methods offer the possibility for linear scaling as the number of time steps increases, in contrast to direct methods where the solution times grow superlinearly. Furthermore, note that higher order methods exhibit significantly lower errors for smaller sized linear systems. In particular, the linear system corresponding to the 2nd order scheme with 250,047 unknowns yields an error that is an order of magnitude lower than the 1st order scheme. As a result, all further simulations are run using second order numerical schemes.

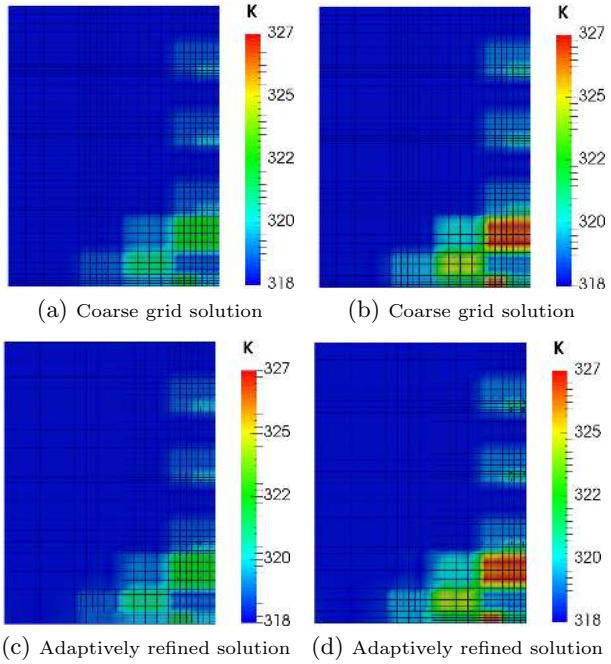
5.2 3-D Thermal Analysis

In this section, two experiments are presented that illustrate the features and functionalities of the analyzer. The first experiment considers a simple case study based on the Nehalem processor, the microarchitecture of which is assumed to be implemented at a 45 *nm* technology node [26]. To generate the power trace of this microarchitecture, we used the Sniper multicore simulator 6.1 [27] to simulate benchmarks of Splash-2 [28] and Parsec-2.1 [29]. McPAT 1.0 [30] converts the trace to power trace as input for the thermal simulation. Corresponding parameters of the physical structure and thermal coefficients are obtained from [31, 9, 32]. The structure of the circuit in this case study consists of 5 material layers, an active layer with different subdomains defined by the Nehalem architecture, a silicon substrate layer, TIM layer, spreader, and a heat sink. In this problem, a simple circuit geometry is assumed so that the computational domain is a rectangular cuboid. These experiments also illustrate the fully adaptive spatio-temporal features of the model.

In the second experiment, a similar but more realistic structure is considered. This circuit has the same 5 layer structure, however this time both the heat spreader and heat sink are larger than the active and silicon substrate layers and overhang the circuit. Additionally, the heat sink is endowed with fin structures. This highlights the versatility of the analyzer to model complex 3-D structures, which is not available in existing thermal analysis tools.

5.2.1 *Rectangular Cuboid Circuit*

For this set of experiments, the thermal simulation was first run using a fixed time step size $\Delta t = 10^{-6}$ s for 500 time-steps ($t_{end} = 0.0005$ s) with a fixed grid, consisting



(a) Coarse grid solution (b) Coarse grid solution

(c) Adaptively refined solution (d) Adaptively refined solution

Figure 3: Rectangular cuboid thermal profiles.

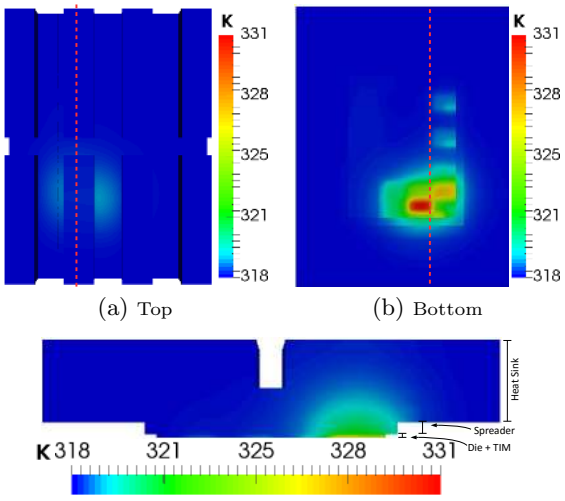
of 25,725 unknowns. This simple, small-scale problem serves as an illustrative example for demonstrating the adaptive capabilities of the proposed analyzer. Each circuit component of the chip is considered to operate at full power capacity when active. Snapshots of the bottom face of the active layer (located in the bottom layer) at time $t = 2.5e-4$ s and $t = 5.0e-4$ s are plotted in Figs. 3(a) and 3(b), respectively.

The same simulation is also performed, again with 500 time-steps of fixed step size $\Delta t = 10^{-6}$ s, however for this case the grid was adaptively refined. The refinement parameters are $\epsilon_{max} = 10^{-2}$ and $\epsilon_{min} = 10^{-6}$. The initial coarse mesh was the same as the first simulation, consisting of 25,725 temperature unknowns. At the final time step, the adaptively refined grid consists of $59,023$ unknowns. The adaptively refined thermal map at times $t = 2.5e-4$ s and $t = 5.0e-4$ s are plotted in Figs. 3(c) and 3(d), respectively. As expected, the refinement occurs in regions of the circuit where the power and variation in temperature are highest.

Lastly, the same structure is simulated to a final time $t_{end} = 5.0e-4$ s on a fixed grid but with adaptively chosen time-steps. The fixed grid is the same as the first simulation, consisting of 25,725 unknowns. The reduction in the number of total time steps taken is presented in Table 4 for two different temporal error tolerances. The more the error tolerance is relaxed, the fewer time steps are taken, yielding improved simulation times. The minimum and maximum temperatures are also reported, illustrating that even with the large reduction in the number of time steps, which is provided by the adaptive time stepping of the the proposed method, there is essentially no loss of accuracy.

ε_t	Time steps	$[T_{min}, T_{max}]$
10^{-3}	50	$[318.00, 326.98]$
10^{-4}	396	$[318.00, 326.88]$

Table 4: Decrease in the number of adaptive time steps for different temporal error tolerances.



(a) Top (b) Bottom

(c) Cross-section

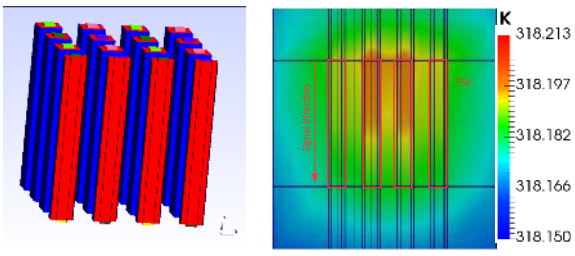
Figure 4: Fin structure thermal profiles.

5.2.2 *Fin Structure*

For this experiment, a realistic circuit geometry is investigated. A thermal simulation is run for 1000 seconds with fixed $\Delta t = 1$ s on a fixed grid consisting of 72,445 unknowns. The larger time steps and simulation time are chosen to fully demonstrate the vertical propagation of heat in the IC over a longer period of time. This experiment accurately shows the detailed vertical flow of heat in the circuit and package, which other current thermal analysis tools have difficulty modeling. The results at the end of the simulation are plotted in Fig. 4. Three views of the circuit are provided. The bottom view in Fig. 4(b) illustrates the formation of a hot spot in the active layer of the chip. The top view of the circuit in Fig. 4(a) shows how the heat from this hotspot has propagated through to the fins of the heat sink. Lastly, a cross-section along the x-direction of the circuit in Fig. 4(c) shows the vertical propagation of the hotspot in the interior of the circuit. Often the heat sink is modeled as several thermal resistors connected in parallel and collapsed at a single node on the outer face of the heat sink. Such a model cannot capture the diffusion of heat within the volume of the heat sink, which is depicted in Fig. 4(c).

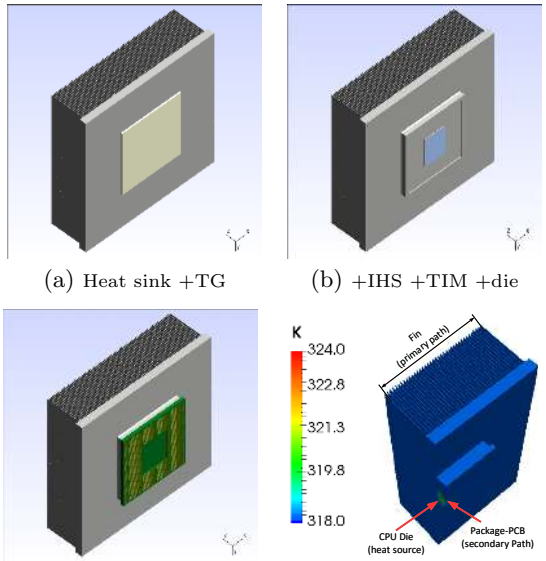
5.2.3 *TSV Structure*

The analyzer can model detailed heat characteristics of vertical channels. Here is an example of a thermal simulation performed on part of a 3-D circuit with signal TSVs. Fig. 5 shows an example of 16 TSVs in 45 nm technology. The simulation is executed with a time step $\Delta t = 10^{-6}$ s for 500 time steps, where each TSV act as vertical signal channel and has its own switching activity during the thermal simulation. In Fig. 5(a), the metal filling of the TSVs (assumed to be copper in this example) is shown by the red color and the liner of the TSVs (SiO_2 in this example) is depicted by the blue color. Inverters are assumed to drive the TSVs and the load of each TSV is also a 1x drive strength inverter. Fig. 5(b) shows a cross-section of the resulting thermal map. This example demonstrates that the flow of heat within each TSV (which depends on the corresponding switching activity) is captured, instead of treating this part of the 3-D stack as a homogeneous material with an effective thermal conductivity (which depends on the number of TSVs within this part of the circuit). Consequently, our tool accurately describes the thermal behavior of any TSV distribution, also enabling appropriate thermal analysis of 3-D ICs.



(a) TSV structure (b) TSV thermal

Figure 5: A TSV example.



(a) Heat sink +TG (b) +IHS +TIM +die

(c) +Package-PCB (d) Thermal map.

Figure 6: A package example.

5.2.4 *Complex Structure*

Fig. 6 shows an example of the Intel Xeon processor (Nehalem Architecture) in a FCLGA (flip-chip land grid array) package referring to the thermal reference guide [33]. Figs. 6(a)-6(c) show the assembly of a heat sink with 72 fins and TG (thermal grease), a CPU die, and package components. The thermal simulation for this IC was performed with $\Delta t = 1$ s for 500 time steps. Fig. 6(d) shows a cross-section along x= 7.2 mm of the IC at the final time step of the simulation. The thermal map shows the secondary heat dissipation path of the package-PCB, which further illustrates the capabilities of the proposed analyzer to model complex heat dissipation paths.

6. CONCLUSION

In this paper, we presented a flexible thermal simulator based on the FEM which uses fast multigrid preconditioned Krylov subspace iterative solvers. The preconditioned iterative solvers offer near linear scaling in simulation times as the mesh is refined. Users can easily configure the simulator with the desired 3-D structure of an IC with various physical parameters and import the corresponding

power information from microarchitecture simulators and power estimators. By deploying spatio-temporal adaptivity, the proposed method provides accurate thermal profiles within acceptable computation times.

7. REFERENCES

[1] M. M. Waldrop, "The Chips Are Down for Moore's Law," *Nature News*, vol. 530, no. 7589, p. 144, 2016.

[2] C. Xu, S. K. Kolluri, K. Endo, and K. Banerjee, "Analytical Thermal Model for Self-Heating in Advanced FinFET Devices with Implications for Design and Reliability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 1045–1058, 2013.

[3] A. Danowitz, K. Kelley, J. Mao, J. P. Stevenson, and M. Horowitz, "CPU DB: Recording Microprocessor History," *Communications of the ACM*, vol. 55, no. 4, pp. 55–63, 2012.

[4] M. B. Taylor, "Is Dark Silicon Useful?: Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse," in *ACM Design Automation Conference*, Jun. 2012, pp. 1131–1136.

[5] J. H. Lau, "Evolution, Challenge, and Outlook of TSV, 3D IC Integration and 3D Silicon Integration," in *International Symposium on Advanced Packaging Materials*, 2011, pp. 462–488.

[6] S. Borkar, "3D Integration for Energy Efficient System Design," in *ACM Design Automation Conference*, 2011, pp. 214–219.

[7] Y. Yang, C. Zhu, Z. Gu, L. Shang, and R. P. Dick, "Adaptive Multi-domain Thermal Modeling and Analysis for Integrated Circuit Synthesis and Design," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2006, pp. 575–582.

[8] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC Thermal Simulation and Modeling via Efficient Multigrid-based Approaches," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 9, pp. 1763–1776, 2006.

[9] 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 Transactions on Very Large Scale Integration Systems*, vol. 14, no. 5, pp. 501–513, 2006.

[10] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs," *IEEE Transactions on Computers*, vol. 63, pp. 2576–2589, 2014.

[11] T.-Y. Wang and C. C.-P. Chen, "3-D Thermal-ADI: A Linear-Time Chip Level Transient Thermal Simulator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1434–1445, 2002.

[12] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics*. Oxford University Press, 2005.

[13] V. Simoncini and D. B. Szyld, "Recent Computational Developments in Krylov Subspace Methods for Linear Systems," *Numerical Linear Algebra with Applications*, vol. 14, no. 1, pp. 1–59, 2007.

[14] J. Boyle, M. Mihajlović, and J. Scott, "HSL_MI20: An Efficient AMG Preconditioner for Finite Element Problems in 3D," *International journal for numerical methods in engineering*, vol. 82, no. 1, pp. 64–98, 2010.

[15] Z. Hassan, N. Allec, F. Yang, L. Shang, R. P. Dick, and X. Zeng, "Full-Spectrum Spatial–Temporal Dynamic Thermal Analysis for Nanometer-Scale Integrated Circuits," *IEEE Transactions on Very Large Scale Integration*, vol. 19, no. 12, pp. 2276–2289, 2011.

[16] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems*. SIAM, 2007, vol. 98.

[17] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[18] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., 2015.

[19] L. Thomason, "TinyXML2," http://www.grinninglizard.com/tinyxml2docs/index.html, 2010.

[20] M. Heil and A. Hazel, "oomph-lib – An Object-Oriented Multi-Physics Finite-Element Library," in *Fluid-Structure Interaction*, ser. Lecture Notes in Computational Science and Engineering, H.-J. Bungartz and M. Schäfer, Eds. Springer Berlin Heidelberg, 2006, vol. 53, pp. 19–49.

[21] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[22] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed. SIAM, Philadelphia, 2000.

[23] U. M. Yang *et al.*, "BoomerAMG: A Parallel Algebraic Multigrid Solver and Preconditioner," *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177, 2002.

[24] R. Falgout and U. Yang, "hypre: A Library of High Performance Preconditioners," *Computational Science - ICCS 2002*, pp. 632–641, 2002.

[25] O. C. Zienkiewicz and J. Z. Zhu, "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 1331–1364, 1992.

[26] R. Kumar and G. Hinton, "A family of 45nm IA processors," in *IEEE International Solid-State Circuits Conference*, Feb. 2009, pp. 58–59.

[27] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An Evaluation of High-level Mechanistic Core Models," *ACM Transactions on Architecture and Code Optimization*, vol. 11, no. 3, p. 28, 2014.

[28] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in *ACM International Symposium on Computer Architecture*, Jun. 1995, pp. 24–36.

[29] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark Suite: Characterization and Architectural Implications," in *ACM International Conference on Parallel Architectures and Compilation Techniques*, Oct. 2008, pp. 72–81.

[30] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *IEEE/ACM International Symposium on Microarchitecture*, Dec. 2009, pp. 469–480.

[31] R. Powell, C. Y. Ho, and P. E. Liley, "Thermal Conductivity of Selected Materials," DTIC Document, Tech. Rep., 1966.

[32] E. Bauer and J. Carlos, "Nehalam Floorplan," 2014, http://www.contrib.andrew.cmu.edu/~eob/.

[33] Intel, "Intel Xeon Processor 5500/5600 Series Thermal/Mechanical Design Guide," http://www.intel.com, 2010.

APPENDIX

A basic description of a component by using our XML tool is given in Code A1. The structure description has a basic name and three major definitions: `<component>` for hierarchical definition of sub-domain components, `<position>` for definition of position and geometry, and `<parameter>` for physical parameters in running thermal simulation. The main component of the example is "name0", which defines position and parameter. Meanwhile, "name0" has two sub-domain components "subname1" and "subname2" to define "name0". Note that if the sub-domain of name0 is not defined by either "subname1" or "subname2", the undefined domain will be automatically defined by "name0". Since "subname1" and "subname2" are in the same level of the hierarchy, the definition of both domains should be mutually exclusive to avoid duplicate definitions.

Code A1: Basic XML example

```
<name0>
        <component>
                <subname1>
                        <component></component>
                        <position></position>
                        <parameter></parameter>
                </subname1>
                        <component></component>
                        <position></position>
                        <parameter></parameter>
                <subname2>
                </subname2>
        </component>
        <position></position>
        <parameter></parameter>
</name0>
```