PACT: An Extensible Parallel Thermal Simulator for Emerging Integration and Cooling Technologies

Zihao Yuan , *Member, IEEE*, Prachi Shukla, *Member, IEEE*, Sofiane Chetoui , Sean Nemtzow,
Sherief Reda , *Senior Member, IEEE*, and Ayse K. Coskun , *Senior Member, IEEE*

Abstract—Thermal analysis is an essential step that enables co-design of the computing system (i.e., integrated circuits and computer architectures) with the cooling system (e.g., heat sink). Existing thermal simulation tools are limited by several major challenges that prevent them from providing fast solutions to large problem sizes that are necessary to conduct standard-cell level thermal analysis or to evaluate new technologies or large chips. To overcome these challenges, we introduce a SPICE-based parallel compact thermal simulator (PACT) that achieves fast and accurate, standard cell to architecture-level, steady-state, and transient parallel thermal simulations. PACT utilizes the advantages of multicore processing (OpenMPI) and includes several solvers to speed up both steady-state and transient simulations. PACT can be easily extended to model a variety of emerging integration and cooling technologies by simply modifying the thermal netlist. In addition, PACT can also be used with popular architecture-level performance and power simulators. In comparison to a state-of-the-art finite-element method (FEM)-based simulator (COMSOL), PACT has a maximum error of 2.77% and 3.28% for steady-state and transient thermal simulations, respectively. Compared to a popular compact thermal simulator, HotSpot, PACT demonstrates a speedup of up to 1.83× and 186× for steady-state and transient simulations, respectively. We also show the applicability and extensibility of PACT through modeling emerging integration and cooling technologies, such as monolithic 3-D integrated circuits and liquid cooling via microchannels, and full-system simulation integration on a 2.5-D system with silicon-photonic network-on-chips (PNoCs).

Index Terms—Compact thermal models (CTMs), SPICE, standard-cell level thermal simulation, thermal simulation.

I. INTRODUCTION

OVER the last few decades, chip temperature has become one of the most important criteria for designing high-performance, cost-effective, and reliable integrated circuits (ICs). Increased power consumption and temperature not only

degrade the performance of a chip but also generate larger subthreshold leakage power and cause reliability challenges [1]. Therefore, thermal analysis is an essential procedure for designing any chip. Conventional thermal analysis relies on the finite-element method (FEM)-based multiphysics simulators (e.g., COMSOL and ANSYS). However, such commercial simulators are computationally expensive and experience long solution times along with large memory requirements [2]. These limitations make commercial simulators unsuitable for evaluating numerous design alternatives or running time scenarios. Therefore, having fast and accurate thermal analysis is crucial for chip design and thermal optimization.

To address the fast thermal analysis needs, researchers have developed tools using compact thermal modeling methods [3]–[7]. Compact thermal models (CTMs) are built based on the well-known duality between thermal and electric properties. In a CTM, the chip is represented as a network of thermal nodes, and the chip temperature is modeled based on an equivalent resistor-capacitor (RC) network of these thermal nodes. A second-order heat diffusion equation is represented using a first-order ordinary differential equation (i.e., an RC equation), which simplifies the boundary conditions and lowers the complexity [3]. The equivalent RC network is then solved using differential solvers to acquire the temperature of each node.

We identify several challenges in existing compact thermal simulators [3]–[5], [7]. First, these thermal simulators target architecture-level thermal simulations only and do not perform standard-cell level thermal simulations. For standard-cell designs, fine-granularity thermal simulation is necessary for an accurate temperature estimation. To demonstrate the necessity of standard-cell level simulation, we select a high power design (Sparc) from OpenROAD [8] and carry out steady-state thermal simulations at various granularities. Fig. 1 shows that architecture-level thermal simulation (e.g., 32×32, 64×64, and 128×128) cannot achieve the same accuracy as standard-cell level simulation (e.g., 256×256, 512×512, and 1024×1024), with a maximum temperature inaccuracy of 3.28 °C and a thermal gradient inaccuracy of 3.56 °C. For thermally aware circuit or policy design (e.g., thermally aware dynamic voltage frequency scaling [9]), such accuracy losses will lead to suboptimal designs or even failures.

Another challenge with the existing compact simulators is that they cannot tackle large and complex problems (e.g., standard-cell level design problems or multilayered chips such as in monolithic 3-D integration [10]) as the simulation time rises dramatically when problem size increases. One reason

TABLE I
SOLVERS, COOLING METHODS, AND INPUTS OF PACT AND OF EXISTING COMPACT THERMAL SIMULATORS. BE: BACKWARD EULER SOLVER;
TRAP: A HYBRID SOLVER OF BE AND THE TRAPEZOIDAL METHOD; FULL INDUSTRIAL DESIGN: REAL-WORLD STANDARD-CELL DESIGNS
SUCH AS THOSE FROM OPENROAD

| Simulator	Steady-state	Transient	Cooling	Inputs
HotSpot [5]	SuperLU	Explicit RK4	NA	Block/architecture-level floorplan and power
3D-ICE [6]	SuperLU	Backward Euler	Liquid cooling	Block/architecture-level floorplan and power
ThermalScope [7]	Gauss Seidel	Trapezoidal	NA	Block/architecture-level floorplan and power
PACT	KLU,KSparse SuperLU,AztecOO, Belos	TRAP, BE, Gear	Liquid cooling and easily extensible	Block/architecture-level floorplan and power and full industrial design
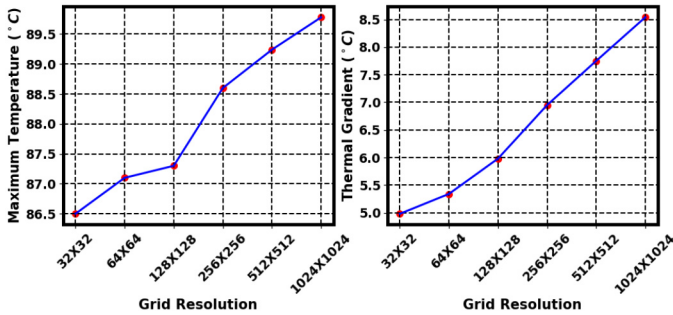
Fig. 1. Temperature profiles for a standard-cell design at various grid resolutions.

for this is that thermal simulators are typically designed to be sequential and cannot easily be parallelized. In addition, the solvers embedded in these simulators are often not efficient enough to perform fine granularity thermal simulations. For example, HotSpot [3] uses explicit adaptive fourth-order Runge–Kutta (adaptive RK4) to conduct transient thermal analysis and this method suffers from numerical instability [11]. Such forward Euler methods may converge slowly for transient simulation (e.g., on the order of days for a standard-cell level chip model), depending on the granularity of the chip as well as the thickness of the chip layers.

A third challenge is that existing compact thermal simulators are either dedicated to a specific cooling technology or it is difficult and time-consuming to extend them for emerging integration and cooling technologies, such as microchannel-based two-phase cooling, thermoelectric coolers (TECs), or two-phase vapor chambers [2], [6], [12]. As a result, research that proposes models for such novel cooling methods frequently rolls out customized software packages (e.g., [4], [6], [7], [12], [13]), resulting in a fragmented space of thermal modeling tools. We summarize the solvers, cooling methods, and inputs of popular compact thermal simulators in Table I.

This article introduces a SPICE-based[1] parallel compact thermal simulator (PACT) that enables speedy and accurate thermal analysis for processors. Recent advances in SPICE [14]–[16] solve many computational challenges associated with modeling electric circuits, and PACT leverages these improvements toward thermal modeling and analysis. Unlike the existing thermal simulators that cannot easily solve standard-cell level simulation problems, PACT supports parallel computing with various types of solvers to provide fast and accurate standard-cell level to architecture-level[2] thermal analysis, regardless of the problem size. In

addition, users can easily extend PACT to model various emerging integration and cooling technologies by adding dependent/independent sources, resistors, and capacitors. The main contributions of this article are as follows:

1) We design and implement PACT to enable fast and accurate parallel thermal simulations.[3] PACT aims to address the fragmentation in the thermal modeling tool space and provides a single tool that is able to conduct efficient thermal evaluation from a standard-cell level to the architecture-level, for a variety of chip integration and cooling technologies. Our ambitious goal with PACT is to release a thermal simulator that provides speedy and accurate thermal simulations and, at the same time, caters to a vast number of (future) designers and technologies with different needs and goals, without requiring a substantial redesign of the tool.

2) To enable standard-cell level thermal simulation, we interface PACT with OpenROAD [8], an end-to-end silicon compiler. This interface allows the evaluation of thermal behavior of full standard-cell level industry designs directly. To speed up standard-cell level thermal simulations, PACT is able to utilize the parallelism in modern computing systems and conduct parallel simulations. We further build a 2.5-D silicon photonic network-on-chip (PNoC) simulation framework [17] as an example to show that PACT is compatible with popular architectural performance and power simulators [18], [19] and is able to run transient simulations.

3) PACT can be easily extended to support various emerging integration and cooling technologies. This is in contrast to the existing compact thermal simulators that only support a specific cooling technology (or no cooling technology). Owing to the easy extensibility of PACT, users can explore the vast co-design space of the computing and cooling systems. In addition, PACT provides various steady-state and transient solvers to enable tradeoffs between simulation speed and simulation accuracy (e.g., for modeling the ultrathin layers in a monolithic 3-D stack).

4) To demonstrate the applicability of PACT, we select large and complex chips (realistic 2-D and monolithic 3-D ICs) and run standard-cell to architecture-level thermal simulations to compare PACT to a well-known compact thermal simulator, HotSpot [3]. PACT shows up to 232× speedup compared to HotSpot in these experiments. To demonstrate the extensibility of PACT, we also integrate an emerging cooling technology model, i.e., liquid cooling via microchannels, and validate it against 3D-ICE [4]. Compared to 3D-ICE, PACT shows a

---

[1]SPICE stands for Simulation Program with IC Emphasis.

[2]Standard-cell level thermal simulation refers to a high grid resolution simulation (i.e., a grid node can occupy one or more standard cells) and architecture-level thermal simulation refers to a relatively low grid resolution simulation (i.e., a hardware block is often occupied by several grid nodes).

[3]PACT is opensourced at https://github.com/peaclab/PACT.

maximum temperature difference of 0.41 °C and 1.12 °C with a speedup of 1.6× and 2.05× for steady-state and transient simulations, respectively.

5) We validate PACT's accuracy by comparing it to HotSpot and COMSOL, using full standard-cell level industrial designs provided by OpenROAD. Compared to COMSOL, PACT has a maximum temperature error of 2.77% for steady-state and 3.28% for transient simulation. We also compare the simulation time to HotSpot using full industrial designs with a high grid resolution ($\geq 256 \times 256$). When compared to HotSpot, PACT achieves speedups of up to 1.83× and 186× for steady-state and transient simulation, respectively.

The remainder of this article starts with a discussion on existing thermal simulators. Section III elaborates on the simulation flow, thermal netlist generation, and compact modeling of various emerging technologies in PACT. We demonstrate the impact of PACT by simulating realistic 2-D ICs, monolithic 3-D ICs, die-stacked 3-D ICs with liquid cooling, and chips with PNoC in Section IV. Section IV also shows the validation and speed analysis of PACT using full industrial designs from OpenROAD. Finally, we conclude the article and discuss the limitations and future work in Section V.

II. RELATED WORK

To maintain safe chip temperatures, researchers have proposed various solutions, including design-time thermal management techniques [1], [20] and runtime policies, such as dynamic voltage frequency scaling [21], [22], task scheduling [23], [24], and thread migration [25], [26]. Several emerging cooling technologies, such as liquid cooling via microchannels [4], [27], [28], TECs [6], [29], two-phase cooling [2], [7], and hybrid cooling (such as a hybrid design of liquid cooling via microchannels and TECs [6], [30]) have also been proposed by the researchers to mitigate the high chip temperatures. These solutions often rely on fast and accurate thermal analysis to enable design exploration and optimization of their design parameters and runtime knobs.

However, when modeling large and complex chips or conducting standard-cell level analysis, existing FEM-based thermal simulators experience high computational complexity and memory usage. For example, simulating the transient behavior of a realistic chip with a high grid resolution can take from several hours to days and easily requires beyond tens of GBs of memory [6].

Compact thermal modeling methodology is a popular solution that can be used to solve the long simulation time problem. In this method, the heat flow (W) passing through a thermal resistor (°C/W) can be represented as an electric current (A) flowing through an electrical resistor (Ω). The corresponding temperature difference (°C) is equivalent to the voltage drop (V). In addition, there is also a thermal capacitance (J/°C) that determines how much heat can be absorbed, which is represented as the electric capacitor (F). A node's temperature can then be modeled as the node voltage of an electric RC circuit as shown in Fig. 2(a). To model a chip with multiple heat sources, heat conduction from each neighbor node is modeled as thermal resistance. Node n_k represents the temperature of the circuit block and the current source i_k represents the power consumption of the corresponding node. C_{k0} represents the thermal capacitance of the node. A thermal
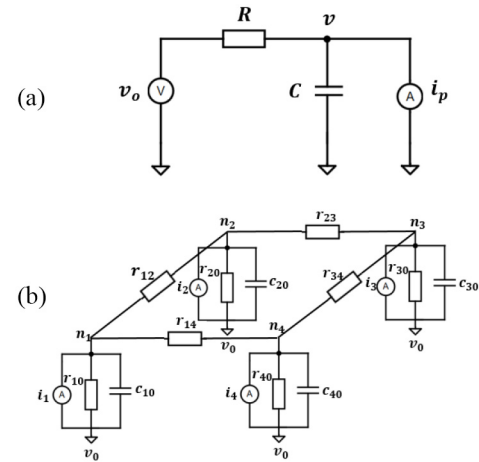

Fig. 2. (a) Thermal RC circuit. R is the thermal resistor, C is the thermal capacitor, v_0 is the ambient temperature, and v is the temperature of the node. (b) Four-node thermal RC network to model temperature distribution.

RC network can be built based on the above parameters as shown in Fig. 2(b).

Several compact thermal simulators have been designed to model the full-chip temperature behavior and emerging cooling solutions [1], [3], [4], [7]. Skadron *et al.* [3] introduced HotSpot, an architectural thermal simulator that utilizes the CTM method to conduct the thermal analysis for processors. The latest version of HotSpot utilizes a sparse matrix direct solver (SuperLU [3]) to obtain steady-state temperature profiles and an adaptive RK4 method to compute the transient thermal behavior [31]. However, the forward Euler methods such as explicit adaptive RK4 can suffer from numerical instability issues [11]. That is, as the number of grids increases or layer thickness decreases to the nanometer level, adaptive RK4 continuously decreases the minimum simulation step size, which slows down the simulation speed significantly. For instance, transient simulation of thin layers (such as in a monolithic 3-D system) with a high grid resolution takes more than a day in HotSpot. There exist other compact thermal simulators that focus on modeling specific types of emerging cooling technologies [4], [7]. However, a common issue in these compact thermal simulators [1], [3]–[5], [7], [12] is that these simulators can only perform sequential thermal simulations and are hard to modify to support parallel thermal simulations. As the problem size increases, the simulation time also increases significantly, especially for standard-cell design transient thermal analysis.

To speed up standard-cell level thermal simulations, Green's function is a promising solution to conduct efficient simulation for high grid resolution thermal simulations [32]. However, if the geometry of the chip or boundary condition changes, Green's function needs to be recomputed or resimulated [33]. Other works have either introduced fast thermal simulation algorithms [34], [35] or used hardware platforms (CPU-GPU platforms) [36] to accelerate the thermal simulations. However, these works focus solely on architecture-level thermal simulations and their methods have not been demonstrated to be applicable for emerging integration and cooling technologies.

Another potential solution is to use the SPICE simulator to build the thermal network and carry out thermal simulations [37], [38]. However, these works model the thermal effects and reliability of interconnects and do not focus on
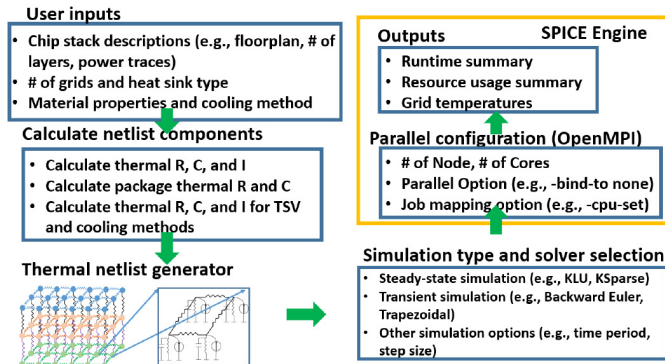
Fig. 3.   PACT simulation flow.

using the SPICE simulator for full system thermal analysis. Moreover, these works are not opensourced and cannot be extended to support emerging integration and cooling technologies.

PACT provides a single tool to conduct an efficient thermal evaluation from the standard-cell level to the architecture-level, for a variety of chip integration and cooling technologies. A key distinguishing feature of PACT is its inherent parallelism, which speeds up the simulation time for standard-cell level thermal simulations while maintaining high accuracy. As PACT is a SPICE-based simulator, it can be easily extended to support and evaluate chip designs with emerging cooling technologies. Moreover, PACT provides flexibility for the users to decide whether they want a faster convergence speed or a more accurate thermal profile by supporting various steady-state and transient solvers.

III. PROPOSED SPICE-BASED THERMAL SIMULATOR

PACT is a SPICE-based standard-cell level to architecture-level parallel compact thermal simulator. To explain how PACT works, we first go over the simulation flow of PACT and then discuss the core of PACT, which is a thermal netlist. A thermal simulator itself should support the modeling of various emerging integration and cooling technologies, and should be compatible with architecture-level performance and power simulators. Because of the simple structure of PACT's thermal netlist and the available SPICE component library, it is easy to extend PACT to support various emerging integration and technologies. We illustrate the extensibility of PACT by modifying the thermal netlist to support the modeling of conventional heat sinks, 3-D ICs (die-stacked 3-D and monolithic 3-D), and liquid cooling via microchannels. We show the compatibility of PACT with popular architecture-level performance and power simulators by creating a 2.5-D PNoC simulation framework. Since PACT acquires full industrial designs from OpenROAD, we also elaborate on the interface between PACT and OpenROAD. The SPICE engine also provides PACT with various steady-state and transient solvers, which can benefit PACT in terms of simulation speed. We discuss the available solvers in PACT and also demonstrate why the selection of the solver is important for evaluating the thermal behavior of processors.

A. PACT Simulation Flow

Fig. 3 shows the simulation flow of PACT. The simulation steps are as follows:

1) Users pass information of the chip stack (such as the number of layers, floorplans, or power traces), material properties (including thermal resistivity and specific heat), problem size (number of grids), heat sink type, and cooling method to PACT.
2) PACT calculates the lateral and vertical thermal resistance, as well as thermal capacitance for each grid. For the layers that consume power, PACT also computes the power consumption of each grid. For emerging cooling layers, PACT determines the corresponding cooling parameters based on the cooling design as well as the input. In the meantime, PACT builds the heat sink requested by users.
3) PACT calculates and assigns R, C, and power values to the corresponding resistors, capacitors, and independent current sources and uses these circuit components to build a thermal netlist.
4) PACT allows the users to specify the type of simulation (steady-state or transient) as well as the solvers.
5) Users can also enable parallel thermal simulations by specifying the number of cores and nodes via OpenMPI [39]. PACT utilizes hypergraph partitioning via the Zoltan library [40] and subdivides and distributes the thermal netlist to the available processors. The Zoltan library provides an effective load balancer and seeks to minimize the message passing overhead among processors [40].
6) PACT solves the RC thermal netlist using the SPICE engine of PACT and outputs the grid temperatures along with the simulation time and resource usage summary.

B. Thermal Netlist and SPICE Circuit Components

Similar to other compact simulators, PACT also calculates the thermal resistor, capacitor, and heat flow values using (1)–(4) shown as follows:

$$R_x = \frac{R_\lambda \cdot w}{l \cdot t}, \tag{1}$$

$$R_y = \frac{R_\lambda \cdot l}{w \cdot t}, \tag{2}$$

$$R_z = \frac{R_\lambda \cdot t}{w \cdot l}, \tag{3}$$

$$C = c_p \cdot w \cdot l \cdot t. \tag{4}$$

R_x, R_y, and R_z are the thermal resistance along the x, y, and z directions, respectively. C is the thermal capacitance of the grid node. R_λ and c_p are the thermal resistivity (mK/W) and specific heat capacity (J/m^{3}K) of the material, respectively. w, l, and t are the width, height, and thickness of the grid node, respectively. To calculate the heat flow values, PACT uniformly divides the power profile of the chip into grids based on the predefined grid resolution. Then, it creates a power matrix (W) to assign power to each grid to represent the heat flow. Since PACT is a SPICE-based simulator, PACT can directly use the circuit components available in the SPICE library to construct the thermal netlist. To extend PACT to support emerging integration and cooling technologies, users need to add additional libraries or utility functions and modify the thermal netlist. It is straightforward to build and modify the thermal netlist by adding and deleting the circuit components or changing the connection of the thermal grids in PACT. Fig. 4 shows the component symbol, component name

Symbol	Component name	Equivalent terminology in PACT
⏚ Resistor symbol	Resistor	Thermal Resistor
Capacitor symbol	Capacitor	Thermal Capacitor
Current source symbol	Current source	Heat flow (power)
Voltage-controlled source symbol	Voltage-controlled current source	Liquid convection in microchannel grid
Voltage source symbol	Voltage source	Assign initial temperature and ambient temperature
PWL source symbol	PWL current source	Enable transient thermal simulation with step response or real power traces

Fig. 4.   SPICE circuit component usage in PACT.

Fig. 5.   High-level simulation flow with the medium-cost heat sink.

in SPICE, and equivalent terminology in PACT. For steady-state simulation, PACT only uses resistors, voltage sources, and current sources to build the thermal netlist and conducts operating point analysis (.OP in SPICE) to solve the thermal netlist. For transient simulation, PACT also calculates the thermal capacitance of the corresponding grid node. To construct the thermal netlist for emerging cooling technologies, users need to add the circuit components from the SPICE library to model the unique cooling behavior of that cooling method. For instance, to model the heat conduction along the microchannel of the liquid cooling via microchannels method, additional voltage-controlled current sources need to be added to the thermal netlist. For transient thermal simulations with real power traces, PACT uses the piecewise linear (PWL) function component and stores the power traces for each grid node in the corresponding PWL component to conduct transient analysis (.TRAN in SPICE).

C. Extensibility of PACT

As we discussed in Section III-B, building the thermal netlist in PACT using SPICE simplifies the construction and modification of the netlist, which enhances the extensibility of PACT. In this section, we give several examples to demonstrate how we can extend PACT to support new technologies, such as different kinds of heat sinks, 3-D ICs, and liquid cooling via microchannels.

1) Heat Sink: There are many different kinds of heat sinks that can be modeled using PACT. In the current version of PACT, we support a medium-cost heat sink that is adopted from a recent work [3] and a fixed air convection heat transfer coefficient (HTC) heat sink.

The medium-cost heat sink represents a combination of the heat spreader, heat sink, and fan and is used to mimic the realistic heat sinks in processors and servers [3]. By modifying the size, material, and air convection HTC of this medium-cost heat sink, it can also be used to model heat sinks for mobile chips. To build this type of heat sink, we add two additional layers on top of the chip to represent the heat spreader and heat sink. In addition to the normal heat spreader and heat sink grid nodes that connect to the chip nodes, we only need to add 12 additional heat sink and heat spreader nodes on the top of the original thermal netlist and populate the resistance and capacitance as the thermal resistors and capacitors attached to these nodes [3]. Similar to HotSpot, four of the additional nodes are assigned to the periphery of the heat spreader, while the remaining of the eight nodes (four inner nodes and four
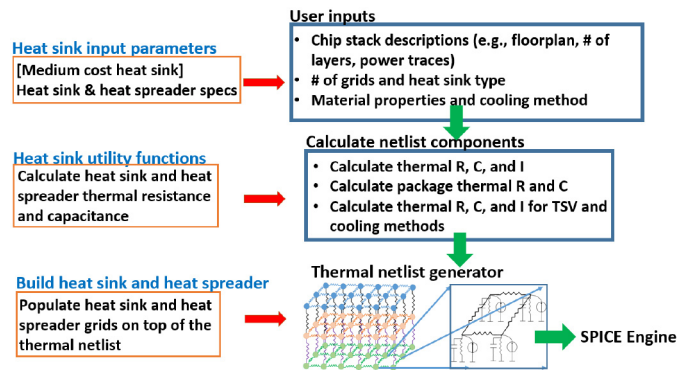
outer nodes) are assigned to the periphery of the heat sink. The thermal resistance and capacitance of the additional nodes of the heat spreader and heat sink are calculated based on the size, thickness, air convection resistivity, thermal conductivity, and specific heat of the heat sink and heat spreader. We show the high-level simulation flow for enabling this medium-cost heat sink in Fig. 5. The heat spreader and heat sink specifications have to be specified through PACT front-end. The medium-cost heat sink utility functions are added to the PACT's backend, to calculate the additional thermal resistance and thermal capacitance introduced by this medium-cost heat sink.

Since simulations of some emerging cooling technologies (e.g., liquid cooling via microchannels and two-phase cooling) require a fixed air convection HTC heat sink or even no heat sink on top of the chip, it is not realistic to use the medium-cost heat sink [2], [4], [6], [7], [12]. Due to this reason, PACT also provides a fixed air convection HTC heat sink, where vertical thermal resistance of the heat sink is the air convection HTC. PACT replaces the heat spreader and heat sink with a dummy layer and connects it to the ground with a vertical thermal resistance calculated using the fixed air convection HTC [6].

2) Modeling Layers With Heterogeneous Materials: Unlike the typical 2-D chips, 3-D ICs need additional TSVs or monolithic interlayer vias (MIVs) to enable interlayer communication and power delivery to the tiers. Therefore, thermal simulators should have the ability to model heterogeneous materials within one layer. Similar to the 3-D extension in HotSpot, PACT is also capable of modeling layers with heterogeneous materials [3], [41]. For a layer with the homogeneous material, PACT assigns the same vertical and horizontal thermal resistance as well as thermal capacitance to each resistor and capacitor component inside of this layer, respectively. For heterogeneous material nodes in a layer, PACT directly modifies the thermal resistance and thermal capacitance of the corresponding heterogeneous nodes and creates thermal resistance and capacitance matrices to generate the thermal netlist.

3) Liquid Cooling via Microchannels in PACT: PACT offers standardized interfaces for easy integration of various compact models of emerging cooling techniques. These models are imported as python modules in PACT. A sample liquid cooling via microchannels chip stack is shown in Fig. 6. In this chip stack, both the bottom and top layers are silicon dies, and the liquid microchannel layer is placed in the middle to mitigate the strong vertical thermal coupling issue for 3-D stacking architectures. We adopt the liquid cooling via microchannels compact modeling methods from recent work [4], [6]. Unlike
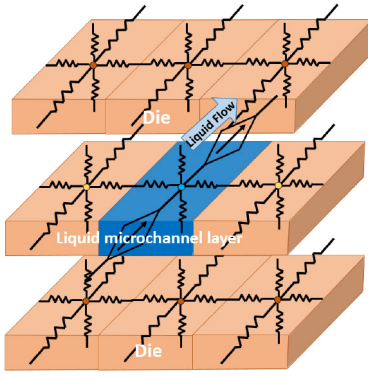
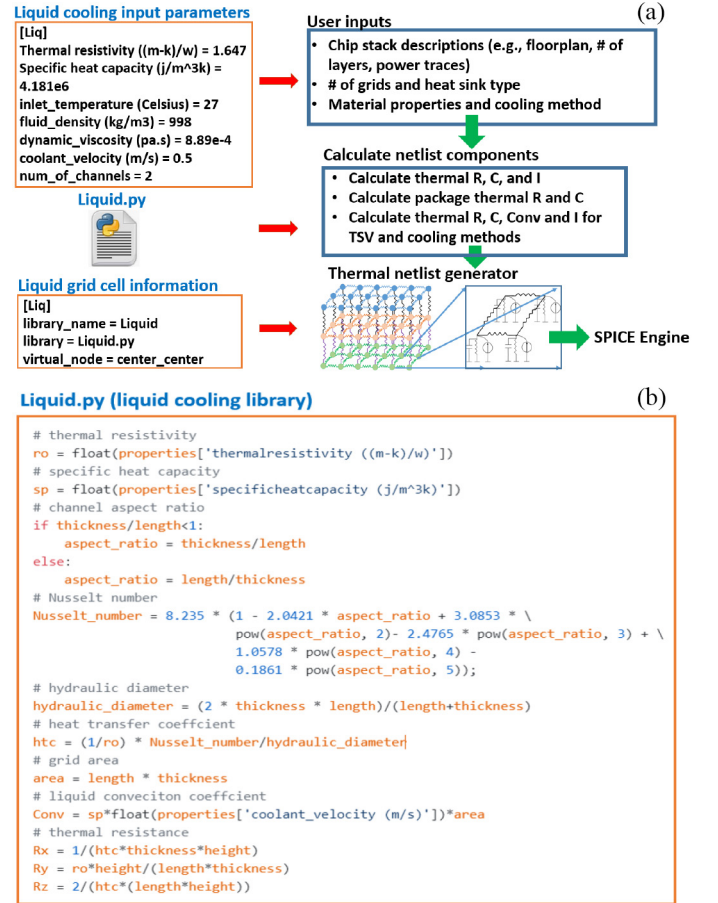Fig. 6.   Small section of a liquid-cooled chip stack.

Fig. 7.   (a) High-level simulation flow with liquid cooling via microchannels. (b) Additional liquid cooling library file for implementing a CTM for liquid cooling via microchannels.

a typical compact thermal grid that consists of 6 thermal resistors for each node to represent the heat conduction from north, south, east, west, top, and bottom directions, a liquid microchannel grid node has only four thermal resistors, which represent the heat conduction between the coolant and the microchannel walls. In PACT, the thermal resistance of a liquid microchannel grid node is calculated based on the vertical and side wall HTCs (i.e., $h_{f,vertical}$ and $h_{f,side}$, respectively) shown as follows [4], [6]:

$$h_{f,vertical} = h_{f,side} = \frac{k_{coolant} \cdot Nu}{d_h}. \tag{5}$$

Nu, $k_{coolant}$, and d_h are the Nusselt number, the thermal conductivity of the coolant, and the hydraulic diameter of the channel, respectively. The additional voltage-controlled current source models the liquid convection effect inside the microchannel. The relationship between the current J_{conv} and liquid convection coefficient c_{conv} is shown as follows:

$$J_{conv} = c_{conv}(T_{in} - T_{out}). \tag{6}$$

PACT uses c_{conv} as the transconductance of the voltage-controlled current source and $\{T_{in}, T_{out}\}$ as the voltage controlling nodes. T_{in} is the average voltage of the previous microchannel node and current microchannel node, and T_{out} is the average voltage of the current microchannel node and the next microchannel node. We show how to implement liquid cooling via microchannels grid nodes in Fig. 7. All the liquid cooling input parameters (e.g., liquid flow velocity, thermal resistivity, specific heat capacity, etc.) have to be specified as user inputs. Users have to create a python module (Liquid.py) to define the vertical and side walls' thermal resistance, as well as the liquid convection coefficient. The thermal resistance and liquid convection coefficient are then used to create the thermal netlist, where vertical and side walls' thermal resistance are modeled as electric resistors and the liquid convection coefficient is used to model the voltage-controlled current source. In addition, users also need to define the liquid grid type (e.g., virtual temperature node is placed at the center of the grid node and not at the bottom of the grid node). PACT calls the correct liquid cooling library (Liquid.py) to obtain the thermal resistance and liquid convection coefficient. In this way, the modeling methodology of liquid cooling via the microchannel grid node in PACT can be applied to model the grid nodes of microchannel-based two-phase cooling and TEC units by creating their respective compact libraries (i.e., python modules).

As we see in Figs. 5 and 7, to support emerging integration and cooling technologies in PACT, users only need to add their additional cooling method libraries and the existing circuit components from the SPICE simulator library to create a new thermal netlist based on the existing design. To model a new cooling technology in PACT, users need to first create the CTM of the cooling method and then map the CTM components to circuit components. The thermal netlist code is well structured and requires minimal changes to support emerging technologies. It is also possible for users to extend the SPICE library with a self-defined circuit component to support other emerging cooling technologies. Depending on the SPICE engine integrated with PACT, users can either modify the .lib file or create a new component written in Verilog-A [15].

D. Compatibility of PACT

To show the compatibility with architecture-level performance/power simulators, we integrate PACT with Sniper [18] and McPAT [19] and create a PNoC cross-layer simulation framework to model the system performance and PNoC power under different activated laser wavelengths and microring resonators (MRRs) lock status. The PNoC simulation framework is adopted from recent work [17] and shown in Fig. 8. The original simulation framework uses HotSpot as the thermal engine; we replace HotSpot with PACT to evaluate the temperature of the PNoC. POPSTAR
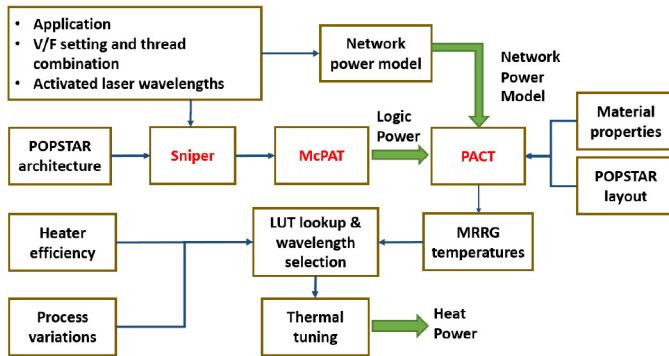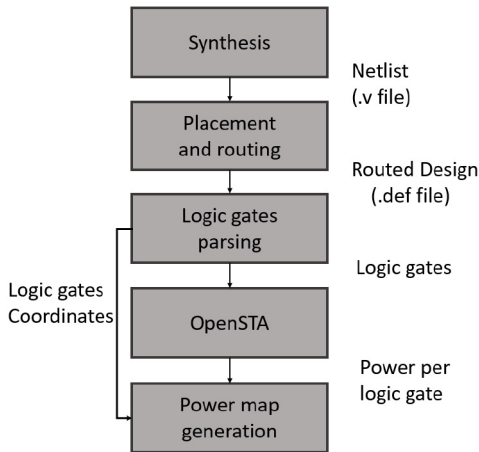
Fig. 8. PNoC simulation framework.

Fig. 9. Flow diagram of OpenROAD.

Fig. 10. Transient simulation time of a two-layer chip stack.

TABLE II
INFORMATION ABOUT AVAILABLE SOLVERS IN PACT

Solver [17]	Type	Mode	Simulation type
KLU	direct	serial and parallel	steady-state
KSparse	direct	serial and parallel	steady-state
SuperLU	direct	serial and parallel	steady-state
AztecOO	iterative	parallel	steady-state
Belos	iterative	parallel	steady-state
Backward-Euler	implicit	serial and parallel	transient
Trap	trapezoidal	serial and parallel	transient
Gear	linear Multistep	serial and parallel	transient

is a 2.5-D manycore system with a PNoC architecture and it has been modeled in Sniper. McPAT is used to compute the core and cache power consumptions, while PACT is used to determine the temperatures of all the MRR groups (MRRGs). We show the temperature validation results against the original PNoC simulation framework in Section IV-B.

E. OpenROAD Interface

OpenROAD is a top-level RTL-to-GDS flow, which generates post-routing design exchange format (DEF) files of a given circuit [8]. We use OpenROAD to get spatial power information at the standard-cell level. Fig. 9 shows the flow diagram of using OpenROAD [8] to generate an industrial input for PACT. Using the DEF files, we generate the power values for every single instance in the design using OpenSTA[4] [8], which is a static timing analysis tool from parallax software that recently went opensource and supports gate-level simulation. OpenSTA is included in the OpenROAD project and the power reporting mechanism is similar to Synopsys PrimeTime [8]. The accuracy of OpenSTA was verified against industrial tools by its developer. Using the DEF files, every single instance in the circuit is passed to OpenSTA [8], while providing the standard-cell library files (lib and lef) and the operating frequency. Finally, based on the die dimensions and the number of grid nodes the user desires, we compute the power per grid node by identifying the gates that belong to each single grid node based on their
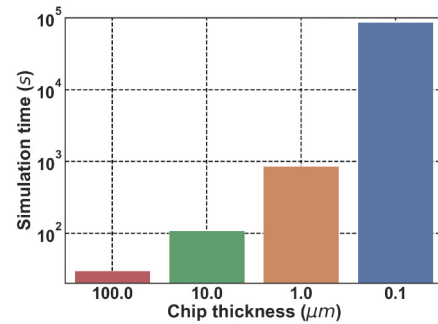
coordinates, and then compute the grid node power by summing the power values of all the gates that belong to it. Since OpenROAD is an opensource project, users can directly utilize this interface to create standard-cell level power maps and perform thermal simulations. For other commercial EDA design flows (e.g., Cadence and Synopsis), PACT can also be used as the backend thermal simulator with the same interface.

F. PACT Solver

The steady-state and transient solvers in the existing compact thermal simulators, such as HotSpot, are not comprehensive enough to model and simulate different chip architectures. For instance, we model and simulate the transient behavior of a two-layer chip stack with a grid resolution equals 50×50. The sampling interval is set to 3.33 μs and the end time is set to 666 μs (total 200 steps). We sweep the layer thickness from 100 μm to 100 nm and show the simulation time results in Fig. 10. The simulation time increases by more than 2880$\times$ when the chip thickness decreases from 100 μm to 100 nm. As we discussed in Section II, the reason behind this simulation time burst is the numerical instability issue of RK4. The forward Euler methods can provide high accuracy and simulation speed for nonstiff equations, but for stiff equations (such as modeling thin layers in HotSpot), the simulation time can be extremely long [11].

Unlike other compact thermal simulators, PACT supports various steady-state solvers (e.g., KLU, SuperLU, and AztecOO) and transient solvers (such as Trapezoidal, backward Euler, and Gear) [15]. We list the information of available solvers in PACT in Table II. KLU, KSparse, and SuperLU are serial solvers. However, if the users use parallel settings with these serial solvers, the thermal netlists are evaluated and assembled in parallel, which is significantly more efficient compared to only using a single processor to evaluate and assemble the netlist [15]. These solvers make PACT comprehensive so that it can be applied to solve thermal netlists from various chip architecture designs at different simulation granularities.

---

[4]OpenSTA: https://github.com/The-OpenROAD-Project/OpenSTA.

There are accuracy and speed tradeoffs among different solvers and simulation modes (parallel or serial) in PACT [15], [42]. The simulation mode, the number of cores, problem size, and the solver type determine the overall accuracy and running time of the thermal simulation. For example, TRAP is a hybrid solver of the backward Euler and the Trapezoidal method, and for the chip stack used in Fig. 10 with 100 nm thickness, the simulation time of PACT using TRAP solver takes less than 29 s. As another example, KLU is a direct solver that is used for single-core steady-state simulation, while AztecOO is an iterative steady-state solver and it outperforms KLU for multicore simulations. For standard-cell level thermal simulations, AzetcOO is preferred since it enables parallel thermal simulations. For architecture-level thermal simulations, KLU outperforms AztecOO mainly because the problem size is small and the additional communication cost of multicore processing takes longer time than single-core simulations. Another example is that for certain thermal netlists, using an iterative solver (e.g., AztecOO) to conduct steady-state simulations may result in a convergence error in PACT [15]. In this case, PACT notifies the users of the convergence error and suggests the users use a direct solver (e.g., KLU) instead.

Since the SPICE engine is designed from the ground up to be distributed-memory parallel, all of these solvers can support parallel simulation via OpenMPI [15]. However, for the existing compact thermal simulators, such as HotSpot, 3D-ICE, and ThermalScope, the designers have not considered the standard-cell level simulation problem and how to utilize the benefits of multicore and multiprocessor simulations with a server cluster to tackle this problem. Therefore, PACT can be parallelized to achieve notable speedup when compared to running thermal simulations via existing compact thermal simulators.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the advantages of running parallel thermal simulations with PACT. We first run steady-state and transient simulations with large and complex realistic 2-D and monolithic 3-D multiprocessor system on chips (MPSoCs) and compare the simulation speed to HotSpot. Then, we show thermal evaluation results against a PNoC simulation framework with HotSpot to show the compatibility of PACT with respect to popular architectural performance and power simulators. In addition, we validate the accuracy of the liquid cooling via microchannels CTM integrated with PACT and compare the simulation time to 3D-ICE. Finally, to validate the accuracy of PACT, we compare the standard-cell level steady-state and transient thermal profiles to those obtained using HotSpot and a FEM-based simulator, COMSOL. Since PACT is a parallel thermal simulator, we also compare the simulation speed of PACT to HotSpot using parallel simulation mode. In addition, we also compare the accuracy and running time of PACT to the Manchester thermal analyzer (MTA) [43].

PACT is written in Python and we use Xyce 6.12 with OpenMPI 3.1.4 as our SPICE engine for all the experiments [15], [39]. We perform our simulations on the Massachusetts Green High Performance Computing Center (MGHPCC). MGHPCC consists of hundreds of compute nodes and each node has at least 128 GB of memory and two sockets. We run on nodes that contain two Intel Xeon E5-2680

TABLE III
EXPERIMENTAL SETUP OF MONOLITHIC 3-D CHIP AND THE SCC-BASED CHIP SIMULATIONS

Chip	Simulator	# of Grids (row)	Step Size (μs)	# of Steps	# of Cores	Solver
Mono3D	HotSpot 6.0	50,100,200	3.33	5	N/A	SuperLU
	PACT	50,100,200	3.33	5	8	AztecOO
SCC	HotSpot 6.0	256,512,1024	3.33	100	N/A	RK4
	PACT	256,512,1024	3.33	100	8	Trap

TABLE IV
SIMULATION RESULTS OF THE MONOLITHIC 3-D CHIP AND THE SCC-BASED CHIP

Simulations	Chip	# of Grids	HotSpot running time	PACT running time
Steady-state	Mono3D	50×50	$1min5s$	$59s$
		100×100	$13min11s$	$5min54s$
		200×200	$3hrs2min$	$15min53s$
	SCC	256×256	$24.7s$	$23s$
		512×512	$3min19s$	$2min15s$
		1024×1024	$26min32s$	$13min55s$
Transient	Mono3D	50×50	$>3\ day$	$2min23s$
		100×100	$>3\ day$	$6min21s$
		200×200	$>3\ day$	$18min48s$
	SCC	256×256	$21min45s$	$1min1s$
		512×512	$5hr38s$	$5min20s$
		1024×1024	$>3\ day$	$18min33s$

v4 CPUs, each with 14 2-way hyper-threaded cores. We use at most four nodes (112 cores) in each of our experiments.

A. Speed Analysis With Complex 2-D and Monolithic 3-D ICs

We use PACT and HotSpot to simulate two large and complex chips to demonstrate the applicability and advantages of PACT. We simulate a 256-core processor (2-D IC) inspired by the Intel SCC scaled to 22 nm [20] and a 33-layer monolithic 3-D IC adopted from recent work [44]. For the 256-core SCC-based chip, the core architecture is based on the IA-32 core [45]. We obtain power profiles of a simulated SCC-based chip from recent work [20]. For our simulations, we select the power profile that results in the highest thermal gradient and chip temperature of the SCC-based chip, to extract the most interesting thermal profile of the chip. The selected power profile has a hot spot power density of 216.6 W/cm^2. We summarize the experimental setup in Table III. We use the same medium-cost heat sink in both HotSpot and PACT and report the simulation speed results in Table IV. We observe in these results that PACT is favorable for solving standard-cell level problems due to its ability to conduct parallel thermal simulations. For the monolithic 3-D chip, when the number of grids = 200×200, PACT takes less than 19 min to finish both steady-state and transient simulations. On the other hand, it takes HotSpot 3 h to finish the steady-state simulation and more than three days for transient. Another advantage of using PACT is that users are allowed to select different types of solvers. We observe that the HotSpot numerical instability problem in transient simulations is exaggerated for the thin layers in monolithic 3-D ICs (thickness < 1 μm), which makes HotSpot and forward Euler solver unsuitable for simulating thin layer chips. For standard-cell level thermal simulations such as Intel SCC-based chip, when compared to HotSpot, PACT achieves a maximum speedup of 1.9$\times$ and 232$\times$ for steady-state and transient simulations, respectively. The reason behind this speedup is that as the problem size increases at a finer granularity, the direct steady-state solver (SuperLU) in HotSpot significantly slows down due to its large memory usage. However, for finer grid resolutions,

TABLE V
EXPERIMENTAL SETUP OF PNoC SIMULATIONS

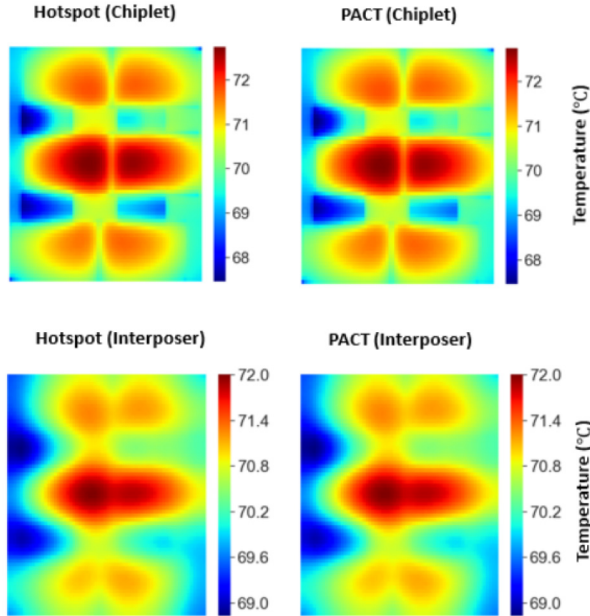| Applications	bt, ft, hpccg, is,lu,mg,shock,sp
VF Settings	$V = 0.85\ V$, f = 533 MHz
Average Core Power	0.83 W
# of threads	48,96
# of grids	64×64
Performance Threshold	10 %
# of cores in PACT	1
Solver in PACT	KLU
Heat Sink	Medium cost heat sink
# of instructions	10 billion

Fig. 11. Thermal maps for running application *bt* with 96 threads and 10% performance constraint using the original PNoC simulation framework and PNoC simulation framework using PACT. MRRG is on the interposer layer. The number of grids used in the simulation is set to 64 × 64.

PACT automatically uses AztecOO, which is an iterative solver with parallel mode to speed up the thermal simulations. For standard-cell level thermal simulations with large and complex chips, PACT outperforms HotSpot in terms of steady-state and transient simulation times. Most importantly, since the majority of the runtime thermal management policies are based on the transient behavior of the chip thermal profile, having a fast transient thermal simulation is particularly important.

B. Full System Simulation of 2.5-D Systems With PNoC

We obtain the power profiles from running the original PNoC simulation framework (using HotSpot as the thermal simulator) with multithreaded applications from HPCCG [46], UHPC [47], and NAS-PB [48] with a different number of thread combinations. We compare PACT's simulation results to the results generated using the original PNoC simulation framework. For the transient power traces, we collect the average power value every 100 million instructions. We summarize the experimental setup in Table V. The detailed model, architecture, policy, and experimental setup can be found in the previous work [17], [49]. Since MRRG temperatures directly determine the heat power, we only compare the temperature results of PACT to HotSpot. Fig. 11 shows the thermal maps of application *bt* with 96 threads simulated using both the original
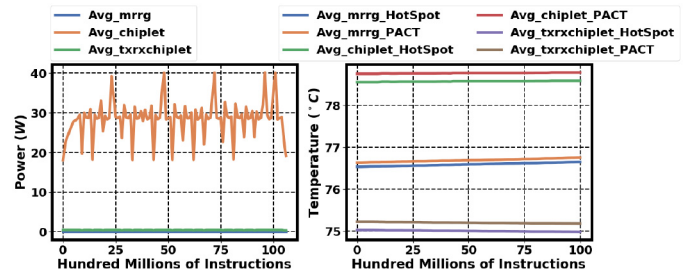

Fig. 12. Transient temperature results for running application *hpccg* with 96 threads and 10% performance constraint using the original PNoC simulation framework and PNoC simulation framework using PACT. The number of grids used in the simulation is set to 64 × 64. The left image shows the average power traces and the right image shows the average temperature traces.

TABLE VI
PNoC SIMULATION RESULTS

# of threads	Apps	Max diff (°C)	Avg diff (°C)
48	bt	0.08	<0.05
	ft	0.08	<0.05
	hpccg	0.47	0.15
	is	0.11	<0.05
	lu	0.34	0.09
	mg	0.02	<0.05
	shock	0.12	<0.05
	sp	0.41	0.09
96	bt	0.31	0.08
	ft	0.37	0.16
	hpccg	0.67	0.19
	is	0.35	0.05
	lu	0.19	<0.05
	mg	0.38	0.16
	shock	0.55	0.21
	sp	0.61	0.27

PNoC simulation framework and the PNoC simulation framework with PACT. Note that MRRG is placed on the interposer layer. PACT thermal maps are almost identical to the thermal maps generated using HotSpot. We also show the transient simulation results compared to HotSpot in Fig. 12. Table VI shows the maximum and average temperature difference for these two PNoC simulation frameworks across all the experiments. As we see in the table, in comparison to the original PNoC simulation framework, the PNoC simulation framework with PACT has less than 1% maximum temperature difference, which demonstrates that PACT is also compatible with popular architecture-level performance and power simulators.

C. Liquid Cooling via Microchannels Simulation Results

To investigate the accuracy of the liquid cooling via microchannels model in PACT, we directly compare the steady-state and transient simulation results against 3D-ICE, which has already been validated against real prototypes [4]. We select a liquid cooling chip stack as shown in Fig. 13(a) and model it in both PACT and 3D-ICE. We summarize the validation setup in Table VII. Note that we set the grid resolution to 1000×5 for these experiments and use the same setup in PACT and 3D-ICE. We summarize the simulation results of PACT and 3D-ICE in Fig. 14. ΔT is the temperature difference between the temperature of the current step and the coolant inlet temperature. PACT shows a maximum temperature difference of 0.41 °C and 1.12 °C for steady-state and transient simulations, respectively. Compared to 3D-ICE, PACT also shows up to 1.6× and 2.05× speedup
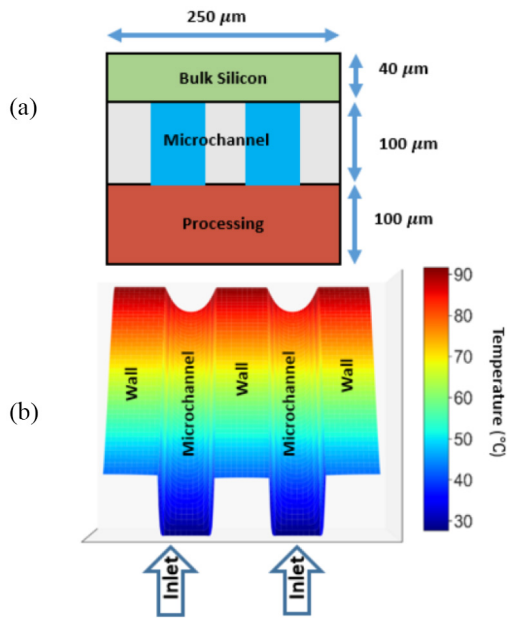
Fig. 13. (a) Front view of the chip stack. (b) Microchannel layer thermal map (power density = 100 W/cm^2 and coolant velocity = 0.5 m/s).
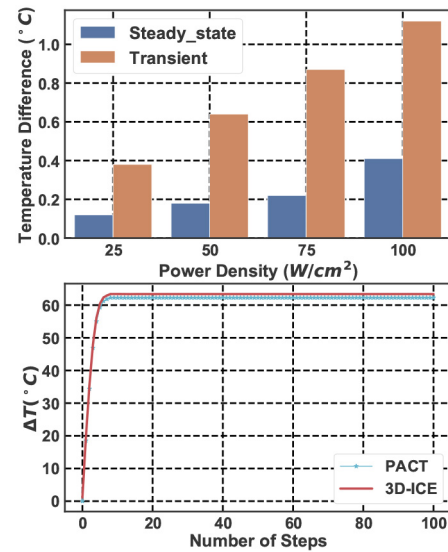

Fig. 14. Liquid cooling via microchannels simulation results. The top image shows the maximum temperature difference for each power profile when coolant flow velocity = 0.5, 1, 1.5, and 2 m/s. The bottom image shows the transient temperature curve of PACT and 3D-ICE when power density = 100 W/cm^2 and liquid flow velocity = 0.5 m/s. This case shows the maximum temperature difference between PACT and 3D-ICE.

TABLE VII
VALIDATION SETUP OF LIQUID COOLING VIA
MICROCHANNELS SIMULATIONS

Chip Length	5 *mm*
Chip Width	250 *μm*
Channel Length	5 *mm*
Channel Width	50 *μm*
Wall Width	50 *μm*
Heat Sink	Fixed air convection HTC heat sink
Air Convection HTC	0.01 *W/m²K*
# of Grids	1000 × 5
Uniform power densities	12.5,25,50,100 *W/cm²*
Flow Velocities	0.5,1.0,1.5,2.0 *m/s*
Step Size	3.33 *ms*
# of Steps	100
# of Cores in PACT	8
Steady-state Solver in 3D-ICE 2.2.6	SuperLU
Transient Solver in 3D-ICE 2.2.6	Backward-Euler
Steady-state Solver in PACT	AztecOO
Transient Solver in PACT	Trap

TABLE VIII
VALIDATION SETUP OF HOTSPOT, COMSOL, AND PACT

Simulator	COMSOL	HotSpot 6.0	PACT
# of grids	256×256		
Solver	FEM-based solver	SuperLU, RK4	KLU, AztecOO, Trap
Heat Sink	Fixed air convection HTC heat sink		
Air Convection HTC	1e5 *W/m²K*		
# of Cores in PACT	1		
Step Size	3.33 *ms*		
Total Step	30		

TABLE IX
STATISTICS OF THE REALISTIC MPSOCS FROM THE
OPENROAD BENCHMARK SET

MPSoCs	Avg PD(*W/cm²*)	Freq(*GHz*)	Util(%)	# of standard cells	Dimension (*μm²*)
PicoSoC	368	3	85	254815	1567×1577
PicoSoC	387	3	90	254815	1522×1534
PicoSoC	409	3	95	254815	1483×1493
Sparc	351	3	85	192871	1225×1244
Sparc	374	3	90	192871	1194×1198
Sparc	391	3	95	192871	1162×1176
Black_Parrot	319	3	85	71285	769×779
Black_Parrot	343	3	90	71285	748×752
Black_Parrot	362	3	95	71285	728×732
Swerv	311	3	85	63423	620×622
Swerv	326	3	90	63423	602×610
Swerv	338	3	95	63423	595×600

for steady-state and transient simulations, respectively. PACT potentially can achieves a higher speedup compared to 3D-ICE when the initial matrix factorization time in 3D-ICE is considered. The main reason for this speedup is that PACT supports parallel thermal simulation. Fig. 13(b) shows the microchannel layer thermal map in PACT (power density = 100 W/cm^2 and coolant flow velocity = 0.5 m/s). The temperature of the coolant increases as the coolant flows across the chip and that results in a higher temperature at the outlet. This trend is expected since the coolant keeps absorbing heat as it flows along the microchannel. Accuracy comparison of PACT's liquid cooling model against another validated recent model [6] also shows very similar results of only up to 0.09 °C maximum temperature difference.

D. Standard-Cell Level Validation of PACT Against COMSOL and HotSpot

To validate the accuracy of PACT, we compare the steady-state and transient simulation results to COMSOL and HotSpot

using different numbers of grids. We summarize the validation setup in Table VIII. The detailed statistics of the MPSoCs from OpenROAD are shown in Table IX. To ensure standard-cell level thermal simulation, the grid resolution should depend on the number of standard cells, standard cell size, and design complexity. Based on the MPSoCs we used in the experiments, a grid resolution of equal or higher than 256×256 should be used to simulate the standard cell designs. The utilization is defined as the ratio of the area of standard cells, macros, and the pad cells to the area of the chip minus the area of the sub floorplan. Higher utilization indicates more logic is packed into a smaller area, which, in turn, results in higher power density. To show the scalability of PACT, the MPSoCs
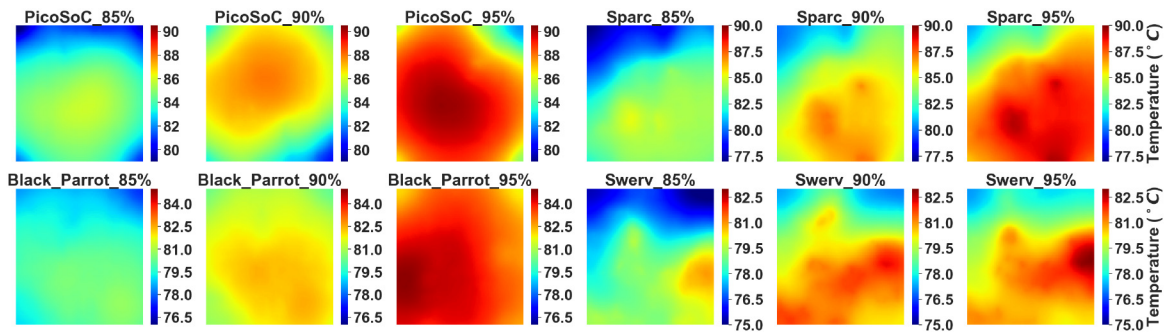
Fig. 15. PACT's thermal maps for the MPSoCs from OpenROAD. The number of grids used in the simulation is set to 256×256. Different utilization levels (shown next to chip names) affect floorplan, chip size, and power density.
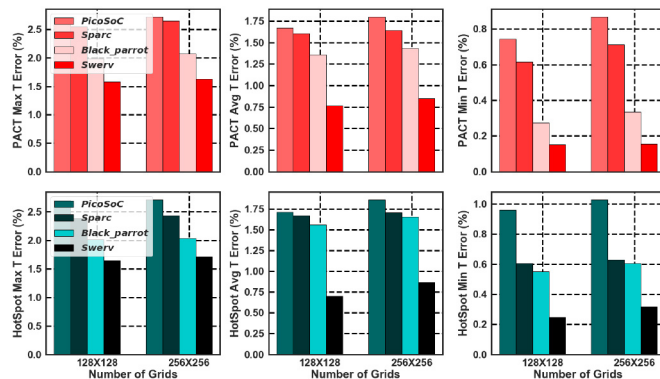

Fig. 16. Steady-state grid temperature validation results (utilization = 95%). MPSoCs with 95% utilization result in the highest maximum, average, and minimum grid temperature error. The error is calculated with respect to COMSOL.
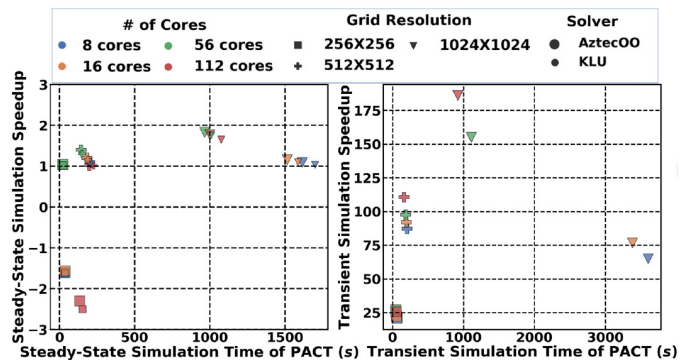

Fig. 17. Steady-state and transient simulation times of PACT. The speedup of PACT against HotSpot is shown on the y-axis. The speedup is computed as the ratio of the simulation times of HotSpot and PACT. Negative values mean HotSpot is faster than PACT for those cases.

in our test set have different power values and chip sizes. The steady-state thermal maps (256×256) of the MPSoCs from OpenROAD are shown in Fig. 15. These thermal maps indicate that the maximum chip temperature across all cases are close to 90 °C and the maximum thermal gradient is around 9 °C. The steady-state grid temperature validation results are shown in Fig. 16. We observe that in comparison to COMSOL, PACT has maximum, average, and minimum grid temperature errors of 2.77%, 1.76%, and 0.89%, respectively, which demonstrates the accuracy of PACT's steady-state simulation. The error is calculated with respect to COMSOL by dividing the grid temperature difference (°C) by the maximum on-chip temperature reported by COMSOL. Fig. 16 also shows the accuracy results for HotSpot with respect to COMSOL. As we see in the figure, when compared to COMSOL, PACT and HotSpot have similar maximum, average, and minimum errors.

Next, we compare the steady-state simulation time of HotSpot and PACT using the setup as shown in Table VIII with various numbers of cores (8, 16, 56, and 112). We also include finer grid resolutions, such as 512×512 and 1024×1024. We show the speedup of PACT's simulation time against HotSpot in Fig. 17. For parallel steady-state thermal simulations with multiple cores, we select KLU and AztecOO as PACT's solvers. As we see in Fig. 17, for steady-state simulations using 256×256 grids with a relatively small number of cores (8 and 16), HotSpot is faster than PACT by as much as 2.3×. But note that the simulation time is rather short in these cases (22–134 s). The reason is that since PACT is written in

Python (and HotSpot is written in C), the front-end processing time of PACT is longer than HotSpot. Another possible reason is that Xyce 6.12 (PACT's SPICE engine) uses a one-step DC analysis to perform operation point analysis, which slows down the steady-state simulation. When the problem size is relatively small (e.g., 256×256), using a large number of cores (e.g., 112) results in a high communication cost between cores and nodes. This communication cost is a potential timing bottleneck [15] and may result in longer simulation times. For standard-cell level problems (e.g., 512×512 and 1024×1024), PACT results in shorter simulation times than HotSpot. The maximum steady-state simulation speedup of PACT compared to HotSpot is 1.83× (1024×1024 with 56 cores). Note that using 112 cores for problem sizes of 512×512 and 1024×1024 also has the high communication cost issue and results in longer simulation times compared to using 56 cores.

We also run steady-state simulations using PACT with KLU. For parallel simulation using a serial solver such as KLU, the thermal netlist is evaluated and assembled using multiple processors, but only one processor is used to solve the netlist [15]. However, AztecOO is a parallel iterative solver that uses multiple processors to evaluate, assemble, and solve the thermal netlist. In Fig. 17, where the thermal netlist is evaluated and assembled with the KLU solver using multiple processors, PACT still achieves speedups compared to HotSpot, with a maximum speedup of 1.75× (1024×1024 with 56 cores).

For transient validations, we create a step response for each MPSoC and compare the grid temperature results against
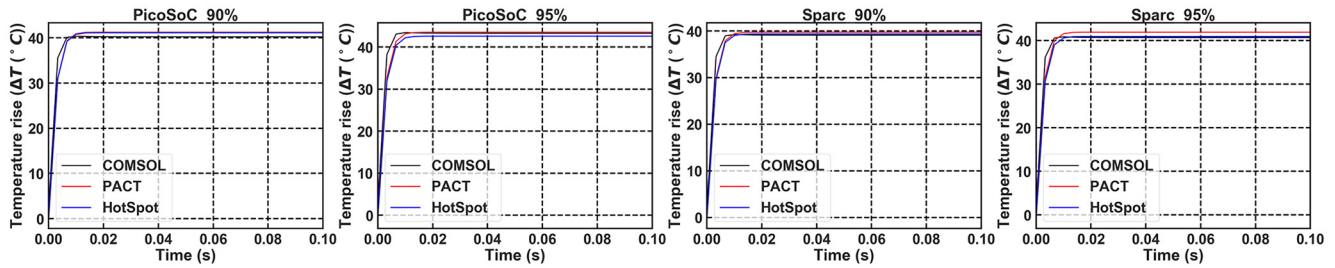
Fig. 18. Transient validation results. The number of grids used in the simulation is set to 256×256. Due to the space limitation, we only show the results that have the highest transient temperature difference.
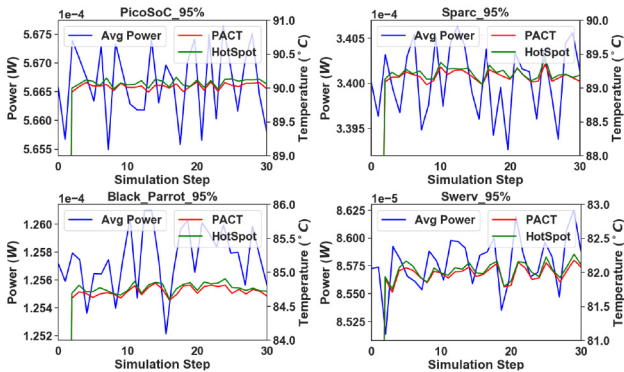

Fig. 19. Synthetic power traces for PACT and HotSpot simulations. Due to the space limitation, we only show the results that have the highest temperature difference.

Fig. 20. Steady-state and transient simulation time of PACT and MTA.

COMSOL and HotSpot. We run each transient thermal simulation with a step time of 3.33 ms and the total simulation time of 99.9 ms (total steps of 30). We show the average grid temperature simulation results of Sparc and PicoSoC in Fig. 18. Compared to HotSpot, PACT has a maximum and average temperature difference of 0.05% and 0.01% across all the experiments, respectively. In comparison to COMSOL, PACT has a maximum and average difference of 3.28% and 1.1%, respectively. ΔT is the temperature difference between the temperature of the current step and the ambient temperature. Since OpenSTA [8] lacks dynamic power traces, we utilize the steady-state power profiles from OpenROAD and randomly apply ±15% additional power values for each standard cell to create synthetic transient power traces. We simulate both PACT and HotSpot using the same setup as shown in Table VIII. The results are shown in Fig. 19. We see that PACT temperature traces overlap with HotSpot temperature traces. The steady-state and transient validation results indicate HotSpot and PACT are at the same accuracy level.

We then compare the transient simulation time of HotSpot and PACT with cores = 8, 16, 56, and 112. For parallel transient thermal simulations with multiple cores, we select TRAP as the solver of PACT. Fig. 17 demonstrates that PACT outperforms HotSpot in every test case. Since HotSpot uses an explicit adaptive RK4 method (fourth-order forward Euler), to ensure the accuracy of simulation results, adaptive RK4 needs to decrease the minimum simulation step to satisfy the numerical stability constraint [11]. On the other hand, PACT uses a TRAP solver (the second-order backward Euler method) that eliminates the numerical instability problem. PACT can achieve a speedup of up to 186× when compared to HotSpot
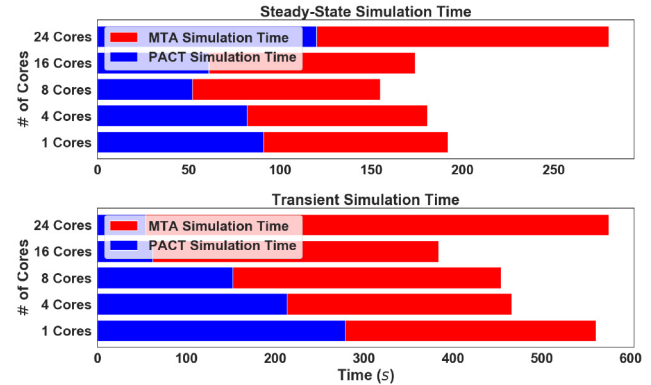
(1024 × 1024 with 112 cores). We also observe that different grid resolutions affect the thermal netlist generation, hypergraph partition, and solver running time, while the chip size affects the thermal netlist generation time only. Across all the standard-cell level simulations for the designs from OpenROAD, PACT's total running time is dominated by the hypergraph partition and solver running time. The thermal netlist generation time is negligible.

E. Standard-Cell Level Comparison of PACT Against MTA

MTA [43] is a thermal simulator that can perform standard-cell level thermal simulations. We compare PACT's temperature results and simulation speed for both steady-state and transient analysis to that of MTA 2.0 using full industrial designs from OpenROAD. The experimental setup is almost the same as Table VIII. We change the transient step size to 3.33 μs with a total number of steps to 100. We also use the same medium-cost heat sink in both PACT and MTA. We select the default mesh provided by MTA, which results in 639 920 degrees of freedom. To ensure a fair comparison, we set the grid resolution in PACT to 256 × 256. For steady-state simulations in MTA, we use {mode 0} and since MTA does not support adaptive mesh refinement for parallel thermal simulations, we use {mode 2} to perform transient simulations with the adaptive time step size. We carry out linear heat model parallel thermal simulations with MPICH. The steady-state and transient maximum temperature differences are 0.45 °C and 0.83 °C. We average the simulation time for each MPSoC selected from OpenROAD as shown in Table IX and present comparison in Fig. 20. Compared to MTA, PACT can achieve

a maximum speedup of $1.98\times$ and $9.64\times$ for steady-state and transient simulations, respectively. Since MTA is a FEM-based thermal simulator and PACT is based on the compact thermal modeling methodology, the complexity of solving the second-order heat equation is obviously higher than solving the first-order thermal RC network. Even with the adaptive time step size, PACT can still achieve better simulation time than MTA.

V. Final Remarks

A. Conclusion

In this article, we presented a SPICE-based Parallel Compact Thermal simulator (PACT) that enables fast and accurate standard-cell level to architecture-level steady-state and transient thermal simulations. PACT can be easily extended to support emerging integration and cooling technologies and is also compatible with popular architecture-level performance and power simulators. To demonstrate the extensibility of PACT, we integrated two types of heat sinks, a model for layers with heterogeneous materials, and a CTM for liquid cooling via microchannels in PACT. We also use PACT to build a PNoC simulation framework with Sniper and McPAT to show its compatibility. In addition, we also created an interface between PACT and OpenROAD that can be used to evaluate the thermal behavior of full industrial designs. When compared to COMSOL, PACT has a maximum temperature error of 2.77% for steady-state and 3.28% for transient simulation. Compared to HotSpot, PACT can achieve up to $1.83\times$ and $186\times$ speedup for steady-state and transient simulations, respectively.

B. Limitations and Future Work

The current version of PACT only supports the cuboid grid. Other grid shapes such as circular (which is useful for simulating round heat pipes) can only be approximated using several cuboid grids. However, this process can be done manually for one circular grid and can then be automated for all the grids across the design. Also, the current version of PACT does not support an adaptive grid (nonuniform grid) and we plan to add this feature in the later versions of PACT. Currently, PACT does not envision the quantum effects in the nanometer scale (40–300 nm [32]). To guarantee the simulation accuracy of PACT, the minimum grid size has to be larger than 300×300 nm^2. For sub-14-nm technology, users have to combine several standard cells into one grid node to conduct thermal simulations. Otherwise, the thermal dissipation will be dominated by the ballistic transportation of acoustical phonon and the overall simulation accuracy will be affected [32]. An open design problem for PACT is to consider the quantum effect in the nanometer scale and use the Boltzmann transport equation to model nanometer-scale phonon effects.

References

[1] M. Pedram and S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: Principles and methods," *Proc. IEEE*, vol. 94, no. 8, pp. 1487–1501, Aug. 2006.
[2] Z. Yuan, G. Vaartstra, P. Shukla, S. Reda, E. Wang, and A. K. Coskun, "Modeling and optimization of chip cooling with two-phase vapor chambers," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, 2019, pp. 1–6.
[3] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. IEEE Int. Symp. Comput. Archit. (ISCA)*, 2003, pp. 2–13.
[4] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs," *IEEE Trans. Comput.*, vol. 63, no. 10, pp. 2576–2589, Oct. 2014.
[5] N. Allec, Z. Hassan, L. Shang, R. P. Dick, and R. Yang, "ThermalScope: Multi-scale thermal analysis for nanometer-scale integrated circuits," in *IEEE/ACM Int. Conf. Comput. Aided Design Dig. Tech. Papers*, 2008, pp. 603–610.
[6] F. Kaplan, M. Said, S. Reda, and A. K. Coskun, "LoCool: Fighting hot spots locally for improving system energy efficiency," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 4, pp. 895–908, Apr. 2020.
[7] A. Sridhar, Y. Madhour, D. Atienza, T. Brunschwiler, and J. Thome, "STEAM: A fast compact thermal model for two-phase cooling of integrated circuits," in *Proc. IEEE Int. Conf. Comput.-Aided Design (ICCAD)*, 2013, pp. 256–263.
[8] T. Ajayi et al., "Toward an open-source digital flow: First learnings from the OpenROAD project," in *Proc. Design Autom. Conf.*, 2019, p. 76.
[9] V. Hanumaiah and S. Vrudhula, "Temperature-aware DVFS for hard real-time applications on multicore processors," *IEEE Trans. Comput.*, vol. 61, no. 10, pp. 1484–1494, Oct. 2012.
[10] S. Wong, A. El-Gamal, P. Griffin, Y. Nishi, F. Pease, and J. Plummer, "Monolithic 3D integrated circuits," in *Proc. Int. Symp. VLSI Technol. Syst. Appl. (VLSI-TSA)*, 2007, pp. 1–4.
[11] G. P. Distefano, "Causes of instabilities in numerial integration techniques," *Int. J. Comput. Math.*, vol. 2, nos. 1–4, pp. 123–142, 1968. [Online]. Available: https://doi.org/10.1080/00207166808803028
[12] Z. Yuan, G. Vaartstra, P. Shukla, E. Wang, S. Reda, and A. K. Coskun, "A learning-based thermal simulation framework for emerging two-phase cooling technologies," in *Proc. Design, Autom. Test Europe (DATE)*, 2020, pp. 400–405.
[13] Z. Yuan, G. Vaartstra, P. Shukla, S. Reda, E. Wang, and A. K. Coskun, "Two-phase vapor chambers with micropillar evaporators: A new approach to remove heat from future high-performance chips," in *Proc. Intersociety Conf. Thermal Thermomech. Phenomena Electron. Syst. (ITherm)*, 2019, pp. 456–464.
[14] G. Massobrio and P. Antognetti, *Semiconductor Device Modeling With SPICE*, vol. 21. New York, NY, USA: McGraw-Hill, 1993.
[15] S. Hutchinson et al., "The Xyce^{TM} parallel electronic simulator—An overview," in *Parallel Computing: Advances and Current Issues*. River Edge, NJ, USA: World Sci., 2002, pp. 165–172.
[16] W. Liu, A. Calimera, A. Nannarelli, E. Macii, and M. Poncino, "On-chip thermal modeling based on SPICE simulation", *International Workshop on Power and Timing Modeling, Optimization and Simulation*, Springer, 2009, pp. 66–75.
[17] A. Narayan, Y. Thonnart, P. Vivet, C. F. Tortolero, and A. K. Coskun, "WAVES: Wavelength selection for power-efficient 2.5D-integrated photonic NoCs," in *Proc. IEEE Design Autom. Test Europe Conf. Exhibition (DATE)*, 2019, pp. 516–521.
[18] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *Proc. ACM Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2011, p. 52.
[19] 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 *Proc. 42nd IEEE/ACM Annu. Int. Symp. Microarchit. (MICRO)*, 2009, pp. 469–480.
[20] F. Eris, A. Joshi, A. B. Kahng, Y. Ma, S. Mojumder, and T. Zhang, "Leveraging thermally-aware chiplet organization in 2.5D systems to reclaim dark silicon," in *Proc. IEEE Design Autom. Test Europe Conf. Exhibition (DATE)*, 2018, pp. 1441–1446.
[21] M. Bao, A. Andrei, P. Eles, and Z. Peng, "On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration," in *Proc. 46th ACM/IEEE Design Autom. Conf.*, 2009, pp. 490–495.
[22] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware DVFS," *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 127–133, Jan. 2010.
[23] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1458–1472, Nov. 2008.
[24] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao, "Thermal-aware task scheduling for 3D multicore processors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 1, pp. 60–71, Jan. 2010.

[25] H. F. Sheikh, I. Ahmad, Z. Wang, and S. Ranka, "An overview and classification of thermal-aware scheduling techniques for multi-core processing systems," *Sustain. Comput. Informat. Syst.*, vol. 2, no. 3, pp. 151–169, 2012.

[26] M. V. Beigi and G. Memik, "Therma: Thermal-aware run-time thread migration for nanophotonic interconnects," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2016, pp. 230–235. [Online]. Available: https://doi.org/10.1145/2934583.2934592

[27] B. Dang, M. S. Bakir, D. C. Sekar, C. R. King, Jr., and J. D. Meindl, "Integrated microfluidic cooling and interconnects for 2D and 3D chips," *IEEE Trans. Adv. Packag.*, vol. 33, no. 1, pp. 79–87, Feb. 2010.

[28] A. K. Coskun, J. L. Ayala, D. Atienza, and T. S. Rosing, "Modeling and dynamic management of 3D multicore systems with liquid cooling," in *Proc. 17th IFIP Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, 2009, pp. 35–40.

[29] I. Chowdhury *et al.*, "On-chip cooling by superlattice-based thin-film thermoelectrics," *Nat. Nanotechnol.*, vol. 4, no. 4, p. 235, 2009.

[30] K. Yazawa *et al.*, "Cooling power optimization for hybrid solid-state and liquid cooling in integrated circuit chips with hotspots," in *Proc. 13th IEEE Intersoc. Conf. Thermal Thermomech. Phenomena Electron. Syst. (ITherm)*, 2012, pp. 99–106.

[31] X. S. Li, "An overview of SuperLU: Algorithms, implementation, and user interface," *ACM Trans. Math. Softw. (TOMS)*, vol. 31, no. 3, pp. 302–325, 2005.

[32] S. Varshney, H. Sultan, P. Jain, and S. R. Sarangi, "NanoTherm: An analytical fourier-Boltzmann framework for full chip thermal simulations," in *Prof. Int. Conf. Comput. Aided Design (ICCAD)*, 2019, pp. 1–8.

[33] A. Ziabari, J. H. Park, E. K. Ardestani, J. Renau, S. M. Kang, and A. Shakouri, "Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 11, pp. 2366–2379, 2014.

[34] S. X.-D. Tan, P. Liu, L. Jiang, W. Wu, and M. Tirumala, "A fast architecture-level thermal analysis method for runtime thermal regulation," *J. Low Power Electron.*, vol. 4, no. 2, pp. 139–148, 2008.

[35] P. Liu, H. Li, L. Jin, W. Wu, S. X.-D. Tan, and J. Yang, "Fast thermal simulation for runtime temperature tracking and management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2882–2893, Dec. 2006.

[36] X.-X. Liu, K. Zhai, Z. Liu, K. He, S. X.-D. Tan, and W. Yu, "Parallel thermal analysis of 3-D integrated circuits with liquid cooling on CPU-GPU platforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 3, pp. 575–579, Mar. 2015.

[37] T. Y. Chiang, K. Banerjee, and K. C. Saraswat, "Compact modeling and SPICE-based simulation for electrothermal analysis of multilevel ULSI interconnects," in *IEEE/ACM Int. Conf. Comput. Aided Design Dig. Tech. Papers*, 2001, pp. 165–172.

[38] T. Y. Wang and C. C. P. Chen, "SPICE-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on modeling order reduction," in *Proc. 5th Int. Symp. Qual. Electron. Design (ISQUED)*, 2004, pp. 357–362.

[39] *Open MPI: Open Source High Performance Computing*, OpenMPI, San Jose, CA, USA, 2014.

[40] E. G. Boman, U. V. Catalyurek, C. Chevalier, and K. D. Devine, "The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering, and coloring," *Sci. Program.*, vol. 20, no. 2, pp. 129–150, 2012.

[41] J. Meng, K. Kawakami, and A. K. Coskun, "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. Design Autom. Conf. (DAC)*, 2012, pp. 648–655.

[42] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, vol. 80. Philadelphia, PA, USA: SIAM, 2002.

[43] S. Ladenheim, Y.-C. Chen, M. Mihajlović, and V. Pavlidis, "IC thermal analyzer for versatile 3-D structures using multigrid preconditioned Krylov methods," in *Prof. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, 2016, pp. 1–8.

[44] P. Shukla, A. K. Coskun, V. F. Pavlidis, and E. Salman, "An overview of thermal challenges and opportunities for monolithic 3D ICs," in *Proc. Great Lakes Symp. VLSI (GLSVLSI)*, 2019, pp. 439–444.

[45] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, Jan. 2011.

[46] A. M. Heroux, "Hpccg solver package," Sandia National Laboratories, Albuquerque, NM, USA, Rep., 2007.

[47] D. Campbell *et al.*, "Ubiquitous high performance computing: Challenge problems specification," Georgia Tech Res. Inst., Atlanta, GA, USA, Rep. HR0011-10-C-0145, 2012.

[48] D. H. Bailey *et al.*, "The NAS parallel benchmarks," *Int. J. Supercomput. Appl.*, vol. 5, no. 3, pp. 63–73, 1991.

[49] A. Narayan, Y. Thonnart, P. Vivet, and A. K. Coskun, "PROWAVES: Proactive runtime wavelength selection for energy-efficient photonic NoCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Nov. 11, 2020, doi: 10.1109/TCAD.2020.3037327.

Zihao Yuan (Member, IEEE) received the B.S. degree in electrical engineering from Central Michigan University, Mount Pleasant, MI, USA, in 2015, and the M.S. degree in electrical engineering from the University of Southern California at Los Angeles, Los Angeles, CA, USA, in 2017. He is currently pursuing the Ph.D. degree in computer engineering with Boston University, Boston, MA, USA.

His current research interests include thermal modeling and optimization techniques for emerging cooling technologies, thermal management, and EDA tool development.

Prachi Shukla (Member, IEEE) received the B.S. degree in information systems from BITS-Pilani (Goa Campus), Sancoale, India, in 2012, and the M.S. degree in computer engineering from Columbia University, NY, USA, in 2015. She is currently pursuing the Ph.D. degree in computer engineering with Boston University, Boston, MA, USA.

Her current research interests include energy-efficient computing, computer architecture, monolithic 3-D systems, and EDA tool development.

Sofiane Chetoui received the M.S. degree in electronics from Ecole Nationale Polytechnique, El Harrach, Algeria, in 2017. He is currently pursuing the Ph.D. degree in computer engineering with Brown University, Providence, RI, USA.

His research areas include power and thermal modeling, as well as post-silicon power and thermal management for constrained devices (mobile devices and IoT devices) in order to maximize performance and energy efficiency.

Sean Nemtzow is currently pursuing the B.S. degree in computer engineering with Boston University, Boston, MA, USA.

He has been a Member of the Performance and Energy Aware Computing Lab, Boston University since spring of 2019. His research interests are in energy efficiency and thermal modeling of integrated circuits.

Sherief Reda (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of California at San Diego, San Diego, CA, USA.

In 2006, he joined the Computer Engineering Group, Brown University, Providence, RI, USA, where he is currently a Professor of Engineering and Computer Science. His current research interests include energy-efficient computing, thermal-power sensing and management, and molecular computing.

Prof. Reda was a recipient of the NSF CAREER Award. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.

Ayse K. Coskun (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science and engineering from the University of California at San Diego, San Diego, CA, USA.

She is a Professor with the Electrical and Computer Engineering Department, Boston University, Boston, MA, USA. Her research interests include energy and temperature awareness in computing systems, novel computer architectures, and management of cloud and HPC data centers.

Prof. Coskun was a recipient of the IEEE CEDA Early Career Award. She currently serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and IEEE TRANSACTIONS ON COMPUTERS.