Energy-Efficient Multiobjective Thermal Control for Liquid-Cooled 3-D Stacked Architectures

Mohamed M. Sabry, *Student Member, IEEE,* Ayse K. Coskun, *Member, IEEE,* David Atienza, *Member, IEEE,* Tajana Šimunić Rosing, *Member, IEEE,* and Thomas Brunschwiler, *Member, IEEE*

***Abstract*—3-D stacked systems reduce communication delay in multiprocessor system-on-chips (MPSoCs) and enable heterogeneous integration of cores, memories, sensors, and RF devices. However, vertical integration of layers exacerbates temperature-induced problems such as reliability degradation. Liquid cooling is a highly efficient solution to overcome the accelerated thermal problems in 3-D architectures; however, it brings new challenges in modeling and run-time management for such 3-D MPSoCs with multitier liquid cooling. This paper proposes a novel design-time/run-time thermal management strategy. The design-time phase involves a rigorous thermal impact analysis of various thermal control variables. We then utilize this analysis to design a run-time fuzzy controller for improving energy efficiency in 3-D MPSoCs through liquid cooling management and dynamic voltage and frequency scaling (DVFS). The fuzzy controller adjusts the liquid flow rate dynamically to match the cooling demand of the chip for preventing overcooling and for maintaining a stable thermal profile. The DVFS decisions increase chip-level energy savings and help balance the temperature across the system. Our controller is used in conjunction with temperature-aware load balancing and dynamic power management strategies. Experimental results on 2-tier and 4-tier 3-D MPSoCs show that our strategy prevents the system from exceeding the given threshold temperature. At the same time, we reduce cooling energy by up to 63% and system-level energy by up to 21% in comparison to statically setting a flow rate setting to handle worst-case temperatures.**

***Index Terms*—3-D integration, liquid cooling, multiprocessor SoC (MPSoC), thermal management.**

I. Introduction

3-D INTEGRATION is a recently proposed design method to overcome the limitations with respect to the delay, bandwidth, and power consumption of the interconnects in large multicore chips, while reducing the chip footprint and improving the fabrication yield. However, high temperatures resulting from higher thermal resistivity [17], [28] are among the main challenges for designing 3-D multiprocessor system-on-chips (MPSoCs). In addition, it is more difficult to remove the heat from 3-D MPSoCs as cores may be located at different tiers and have significantly different heating/cooling rates in comparison to conventional 2-D chips [9]. 3-D MPSoCs are also prone to larger thermal variations, which have adverse effects on system reliability, performance, and cooling costs [11].

A number of thermal management techniques have been proposed for controlling temperature on 2-D (single-tier) MPSoCs. Dynamic voltage and frequency scaling (DVFS) and thread migration/scheduling based on thermal feedback are examples of such techniques [14]. Recent research has extended 2-D management techniques for workload scheduling and DVFS-based thermal management in 3-D MPSoCs [9], [44], [45]. However, as power densities, number of cores, and number of tiers increase, extremely high temperature values appear in 3-D stacks [44], resulting in severe restrictions in high-performance 3-D MPSoC design.

Interlayer liquid cooling is an attractive solution to address the high temperatures in 3-D chips, due to the higher heat removal capability of liquids in comparison to air [7], [10]. However, we need to integrate liquid cooling management with task scheduling and DVFS to maximize the energy efficiency and reliability of high-performance 3-D MPSoCs. In addition, previous work has shown that as workload dynamics change at run-time, choosing the flow rate setting dynamically to meet the cooling demand saves significant energy [10].

Combining various control knobs in a single low-overhead optimum controller is a highly challenging task, as the control parameters differ in their time constants, performance/energy overheads, and benefits. For example, DVFS has an overhead in the order of tens to hundreds of microseconds, while flow rate changes may take hundreds of milliseconds. Hence, simultaneously utilizing such distinct control knobs at run-time requires a thorough understanding of their impact and interactions.

In this paper, we propose a combined design-time/run-time thermal management strategy for 3-D MPSoCs to interpret current system state (temperature, power, and workload) with a degree of uncertainty and flexibility. In particular, we advance

Manuscript received May 19, 2011; revised July 19, 2011; accepted July 21, 2011. Date of current version November 18, 2011. This work was supported in part by the PRO3D EU FP7-ICT-248776 Project, and in part by the Nano-Tera.ch RTD Project CMOSAIC (ref. 123618), which is financed by the Swiss Confederation and scientifically evaluated by Swiss National Science Foundation. This paper was recommended by Associate Editor Y. Xie.

M. M. Sabry and D. Atienza are with the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: mohamed.sabry@epfl.ch; david.atienza@epfl.ch).

A. K. Coskun is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA (e-mail: acoskun@bu.edu).

T. S. Rosing is with the Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA 92093 USA (e-mail: tajana@ucsd.edu).

T. Brunschwiler is with IBM Research GmbH, Zurich Research Laboratory, Zurich 8803, Switzerland (e-mail: tbr@zurich.ibm.com).

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

Digital Object Identifier 10.1109/TCAD.2011.2164540

the state-of-the-art on dynamic thermal management (DTM) for 3-D MPSoCs in the following directions.

1) We perform a thorough design-time thermal impact analysis of various DTM methods (i.e., flow rate control, DVFS, and task scheduling/migration) in 3-D MPSoCs with interlayer liquid cooling to identify a set of optimal decisions in achieving energy-efficient thermal management with minimal performance degradation.
2) We utilize this study to design a fuzzy controller that extends previous work [30] by including a complete stability and implementation complexity analysis. The controller is able to trigger appropriate flow rate and DVFS adjustments based on the system's temperature, workload requirements, and spatial location of the computational units and caches.
3) We analyze the benefits of the state-of-the-art temperature-aware job scheduling methods [9], [44] in our DTM scheme for 3-D MPSoC with liquid cooling. We propose a novel job scheduler which takes the physical locations of the units in the 3-D stack into consideration to stabilize the temperature on the die and to improve cooling efficiency without affecting performance.
4) We provide extensive experimental evaluation on 2-tier and 4-tier 3-D MPSoCs by comparing our thermal management approach with respect to state-of-the-art DTM techniques [10], [30], [44] for a large number of metrics, such as peak temperatures, thermal gradients, energy efficiency, and performance degradation. These experiments show that our strategy completely removes thermal hot spots in the 3-D stacks, while saving cooling energy by up to 63% and 45%, and system-level energy by up to 21% and 18%, in comparison to using a static worst-case flow rate setting and in comparison to using a look-up table-based control [10], respectively.
5) We analyze how core and cache temperatures in different locations of the 3-D MPSoC are affected by the liquid flow to reduce the pumping power and the thermal gradients in the 3-D stack.

The rest of this paper starts with an overview of the prior work in Section II. Then, Section III describes our developed thermal model for liquid-cooled 3-D MPSoCs. Next, we describe the proposed design-time/run-time thermal management strategy in Section IV. In Section V, we explore the thermal impact analysis of various thermal control knobs on the 3-D MPSoCs temperature. Section VI describes our new fuzzy controller for DTM in 3-D MPSoCs with liquid cooling, and we present the experimental results in Section VII. Finally, Section VIII summarizes the main conclusions of this paper.

II. Related Work

A. Accurate and Compact Thermal Modeling of 2-D/3-D ICs

Accurate thermal modeling is critical in system design and evaluation. HotSpot [33] is a R-C network-based thermal model that calculates transient temperature response for a given power trace. To reduce the potentially long thermal simulation time in HotSpot for large MPSoCs, recent work proposes a thermal emulation framework using field-programmable gate arrays [2]. Latest versions of HotSpot include 3-D modeling, and methods to extend HotSpot for liquid-cooled 3-D MPSoCs are available [8]. 3-D interlayer cooling emulator (3D-ICE) [34] is a new thermal modeling tool specifically designed for transient thermal analysis of 3-D stacks, including interlayer liquid cooling modeling. The authors showed that their modeling and simulation framework can be extended to account for different cavity structures, such as pin-fins [35]. Feng *et al.* [15] introduced a thermal simulation framework of 3-D stacks where graphical processing units are used for accelerating temperature calculation. All these methods to model and speedup 3-D MPSoC thermal simulations are complementary to our paper.

B. Interlayer Liquid Cooling

The use of convection in microchannels to cool down high power density chips has been an active area of research since the initial work by Tuckerman and Pease [41]. Their liquid cooling system can remove 1000 W/cm^2; however, the volumetric flow rate and the pressure drop are too large for practical applications in 3-D integrated circuits (ICs). Recent work shows that back-side liquid cold plates, i.e., staggered microchannel and distributed return jet plates, can handle up to 400 W/cm^2 in single-chip applications [6]. The heat removal capability of interlayer heat-transfer with pin-fin inline structures for 3-D chips has been also investigated [7], [18]. At a chip size of 1 cm^2 and maximal difference between junction and liquid cooling temperatures of 60 K, the heat-removal performance is more than 200 W/cm^2 at interconnect pitches larger than 50 μm. However, research in liquid cooling has typically developed thermal packaging solutions rather than utilizing active cooling in DTM, which is the focus of our paper.

C. DTM of 3-D MPSoCs

Prior work on thermal management for 3-D MPSoCs mainly addresses design-time optimization, such as thermally aware floorplanning [16], integrating thermal via planning in the 3-D floorplanning process [22], and joint optimization that targets temperature, power interconnect, and signal wires [20]. A tradeoff study in recent work compares thermal behavior and interconnect congestion for two 3-D MPSoC cooling technologies: inter-tier liquid cooling and thermal through-silicon-vias (TSVs) [20]. This paper shows that inter-tier liquid cooling has superior cooling abilities, but induces limitations for TSVs and increases cost with respect to using thermal TSVs only.

Recent work considers DTM for 3-D MPSoCs. Zhu *et al.* [45] evaluated several policies for task migration and DVFS by exploring thermal profiles of adjacent processing cores on the same vertical column (interlayer adjacent) or within the same layer (intralayer). Zhou *et al.* [44] integrated a thermally aware task scheduler with DVFS on a 2-tier system with eight cores. A recent paper proposed a temperature-aware scheduling method specifically designed for air-cooled (AC) 3-D systems [9]. Their method considered thermal heterogeneity among the 3-D MPSoC layers; however, it does not study interlayer cooling. Prior work on DTM in AC 3-D systems

demonstrated very high temperatures (85–120 °C), motivating the search for alternative energy-efficient cooling techniques beyond conventional methods.

Prior liquid cooling work [8] evaluated existing thermal management policies on a 3-D MPSoC with a fixed-flow rate value, and the benefits of using a policy to increment/decrement the flow rate based on temperature measurements. Our recent paper [10] considered the energy efficiency of 3-D MPSoCs with variable flow rate adjustment and thermally aware load balancing, without utilizing DVFS for increased energy savings. Recently, Qian *et al.* [29] explored the use of a cyber-physical approach to manage the temperature of 3-D MPSoCs with inter-tier liquid cooling. They construct their control mechanism with software-based thermal estimation and prediction. They use a nonuniform liquid flow in different microchannels, to meet the cooling demands of different modules.

This paper brings several major contributions over prior work. First, we study the thermal impact of various thermal management methods (flow rate control, DVFS, and task scheduling/migration) in 3-D MPSoCs to identify a set of optimal decisions in achieving energy-efficient thermal management with minimal performance degradation. Second, we utilize this study in designing a fuzzy controller that generates the appropriate control decisions, as well as discussing the controller's use in conjunction with a novel job scheduler to further improve the cooling efficiency without affecting performance. Finally, we demonstrate how design-time optimization can help improve energy efficiency in 3-D systems with liquid cooling.

III. Thermal Modeling Framework for Liquid-Cooled 3-D MPSoCs

Modeling the temperature dynamics of liquid-cooled 3-D MPSoC architectures consists of forming the grid-level thermal R-C network model of the whole stack, modeling the TSVs and the microchannels [34], [45], and modeling the impact of the pump and coolant flow rate.

Fig. 1 shows the 3-D MPSoCs designs we target in this paper. These 3-D MPSoCs consist of two or more stacked tiers (i.e., tiers *A* and *B* include cores, L2 caches, crossbar, and other units for memory control and buffering), and microchannels are built in between the vertical layers for liquid flow (tier *C*). In this paper, we assume face-to-back bonding, and we assume forced convective interlayer cooling with single-phase fluids, in particular water [7]. TSVs are located and etched within the microchannel walls, and uniformly distributed in TSV bundles as in layout *C* in Fig. 1. Thus, the channel wall dimensions take the TSV features and spacing requirements into account. We use uniformly distributed microchannels, and equivalent fluid flow rate through each channel in the same layer. Although variation of the fluid flow due to nonuniform heat flux [18] can exist, we examine the impact of this flow through simulations and show that variations do not exceed 2% for single-phase flows (see Section V). Finally, the liquid flow rate provided by the pump can be dynamically altered at run-time [6], [7].

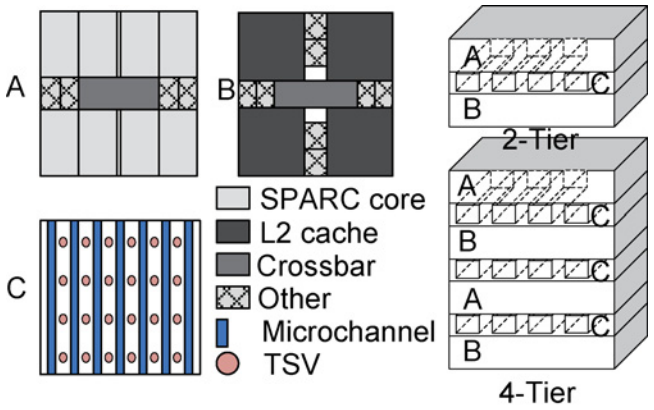

Fig. 1. Layouts of the modeled 3-D MPSoCs using Sun T1 SPARC cores.

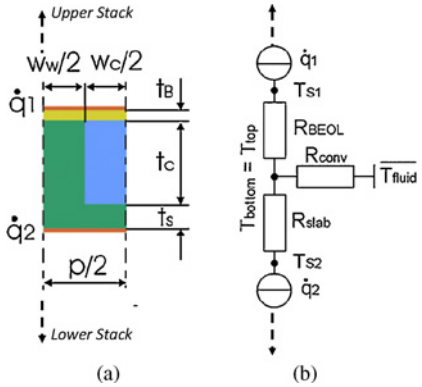

Fig. 2. Cross-section of the (a) 3-D layers and the (b) resistive network [10].

A. Grid-Level Thermal Model for Liquid-Cooled 3-D MPSoCs

Similar to thermal modeling in 2-D chips [1], [33], 3-D thermal modeling is performed using a discrete thermal model that forms the R-C circuit for given grid dimensions. In this paper, we utilize 3D-ICE [34] to perform the transient thermal simulation of 3-D MPSoCs with interlayer liquid cooling.

To model the heterogeneous characteristics of the interlayer material including run-time variable flow rate in microchannels, we introduce the ability to change the resistivity value of the microchannels cells, based on the liquid flow rate, at run-time. However, the thermal resistance of the channel walls (with the TSVs) are fixed to a specific value.

We compute the local junction temperature of 3-D MPSoCs with liquid cooling using the resistive network of Fig. 2 for each layer, where heat is dissipated to both opposing vertical directions (up and down) from the heat sources. The top and bottom layers' temperatures are denoted by T_{S1} and T_{S2}, respectively. Then, the thermal resistance of the back-end-of-line (BEOL) layers (R_{BEOL}), the silicon slab thermal resistance (R_{slab}), and the convective thermal resistance (R_{conv}) are combined to model the 3-D stack, while the heat flux values ($\dot{q}$) represent the heat sources. We compute the junction temperature ($T_j = T_{S1}$ in Fig. 2) by assuming isothermal channel walls (i.e., the top and bottom parts of the microchannel have the same temperature).

Table I lists all the parameters used in our thermal model, which are based on experimentally validated liquid cooling technology [7]. Note that the provided flow rate ($\dot{V}$) range is for the interlayer cavity, and it is equally divided between all the microchannels.

TABLE I
PARAMETERS FOR COMPUTING (1)

Parameter	Definition	Value
$R_{\text{th}-\text{BEOL}}$	Thermal resistance of wiring levels	(3)
t_B	BEOL thickness (Fig. 2)	12μm
k_{BEOL}	Conductivity of wiring levels	2.25 W/(m·K)
$R_{\text{th}-\text{heat}}$	Effective thermal resistance	(5)
A_{heater}	Heater area	Area of grid cell
c_p	Coolant heat capacity	4183 J/(kg·K)
ρ	Coolant density	998 kg/m^3
$\dot{V}$	Volumetric flow rate per cavity	0.01–0.0323 l/min
h	Heat-transfer coefficient	371323 W/(m^2·K)
w_c	Channel width (Fig. 2)	50 μm
w_w	Wall width (Fig. 2)	100 μm
t_c	Channel thickness (Fig. 2)	100 μm
t_s	Silicon slab thickness (Fig. 2)	50 μm
p	Channel pitch (Fig. 2)	150 μm

The junction temperature change (shown in Fig. 2) is a sum of the following three components: 1) the thermal gradient due to conduction (ΔT_{cond}); 2) the change in coolant temperature, which increases linearly with position along the channel due to heat absorption (ΔT_{heat}); and 3) the convective (ΔT_{conv}) portion, which is independent of the flow rate with fully developed hydrodynamic and thermal boundary layers that have been reached [7]. Hence, the total temperature rise on the junction (ΔT_j) can be computed as follows:

$$\Delta T_j = \Delta T_{\text{cond}} + \Delta T_{\text{heat}} + \Delta T_{\text{conv}}. \tag{1}$$

The thermal gradient due to heat conduction through BEOL layer (ΔT_{cond}) can be computed using the definitions and values of t_B and k_{BEOL} in Table I as follows:

$$\Delta T_{\text{cond}} = R_{\text{th}-\text{BEOL}} \cdot \dot{q}_1 \tag{2}$$

$$R_{\text{th}-\text{BEOL}} = \frac{t_B}{k_{\text{BEOL}}}. \tag{3}$$

The temperature change between two adjacent cells due to heat absorption (ΔT_{heat}) is computed as follows:

$$\Delta T_{\text{heat}} = (\dot{q}_1 + \dot{q}_2) \cdot R_{\text{th}-\text{heat}} \tag{4}$$

$$R_{\text{th}-\text{heat}} = \frac{A_{\text{heater}}}{c_p \cdot \rho \cdot \dot{V}}. \tag{5}$$

Finally, (6) and (7) show how to calculate the temperature change due to convection (ΔT_{conv}). The heat-transfer coefficient (h) is dependent on the hydraulic diameter, Nusselt number, and conductivity of the fluid [7]. As the effective heat-transfer coefficient (h_{eff}) is not affected by flow rate changes with developed boundary layers, we compute this parameter prior to simulation and use it during the experiments (Section VII). Fig. 2 demonstrates w_c, t_c, and p, and Table I shows their definitions and values.

$$\Delta T_{\text{conv}} = (\dot{q}_1 + \dot{q}_2) \cdot h_{eff} \tag{6}$$

$$h_{eff} = h \frac{2 \cdot (w_c + t_c)}{p}. \tag{7}$$

In our target 3-D MPSoCs, we use copper TSVs with 50 μm diameter and 150 μm pitch. These particular dimensions have been used in 3-D ICs with interlayer cooling in previous work [3], [19]. Considering the dimensions and pitch requirements of microchannels and TSVs, there are 66 microchannels in each cavity (tier C), and each cavity is located between every two silicon tiers (A or B). Thus, there are 66 and 198 microchannels in the 2-tier and 4-tier 3-D MPSoCs, respectively. Based on previous work [3], [8], we assign a uniform TSV density for the interlayer material.

TABLE II
PARAMETERS DEFINITION USED IN RELATING FLOW RATE TO THE APPLIED PRESSURE DIFFERENCE

Parameter	Definition
ε	Cavity porosity (0.33)
κ	Cavity permeability (7.17E-11 m^2)
μ	Dynamic viscosity (1E-3 Pascal· s at 300 K)
ΔP	Pressure difference between the inlet and outlet ports (1 bar)
L	Channel length

B. Modeling the Pump and Liquid Flow Rate

All the microchannels are connected to a pump to receive the coolant. The pump injects the fluid at a certain pressure difference required by the system. In accordance with prior work on liquid cooling technology validation for 3-D stacks [7], we fix the maximum applied pressure on the stack to 1 bar. The flow velocity within the microchannels is governed by:

$$v_{\text{bulk}} = \frac{v_{\text{darcy}}}{\varepsilon} \tag{8}$$

$$v_{\text{darcy}} = \frac{\kappa}{\mu} \cdot \nabla P \tag{9}$$

$$\nabla P = \frac{\Delta P}{L} \tag{10}$$

where v_{bulk} is the actual velocity in the channel, and the other symbols are defined in Table II. These equations show that the fluid velocity is dependent on the applied pressure difference and on the fluid temperature via the dynamic viscosity. In particular, if the fluid temperature increases the dynamic viscosity decreases, and the fluid moves at a higher speed. However, in our experiments, we have observed that the change in the fluid temperature with nonuniform heat fluxes does not exceed 10°, which in turn negligibly affects the dynamic viscosity. Therefore, we assume that the fluid velocity remains constant at a constant pressure gradient.

In a high-performance computing cluster, several chips are included in a set of racks and a central pump must be used for several 3-D stacks for reducing the cost [27]. Hence, a centrifugal pump EMB MHIE [31] is responsible for the fluid injection to a cluster of nodes via a pumping network. This pump has the capability of producing large discharge rates at small pressure heads. To enable different flow rates for each stack, the cooling infrastructure includes valves in the network. We assume normally closed valves (NCVs) provided by the Festo group [32]. NCVs use external power to reduce the pressure drop and to increase the flow rate. Fig. 3 shows the pump and valve power consumption for three flow rate settings of a single stack deployed in a cluster with 60 similar 3-D computing stacks, as proposed in the Aquasar data-center design [27]. The maximum energy required to inject the fluid to all stacks is approximately 180 W, which is a significant overhead to the whole system as this value is similar to the energy consumed by a 4-tier 3-D chip. Thus, the energy consumed in the liquid cooling subsystem should be minimized.

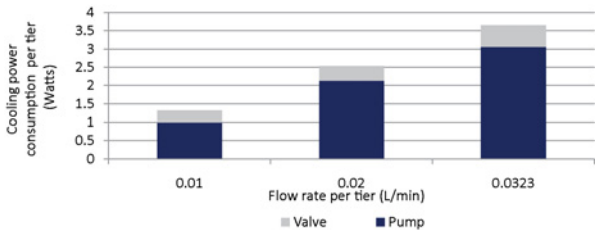

Fig. 3. Power consumption and flow rates of the cooling infrastructure per one stack in a 60 3-D chips cluster [27].

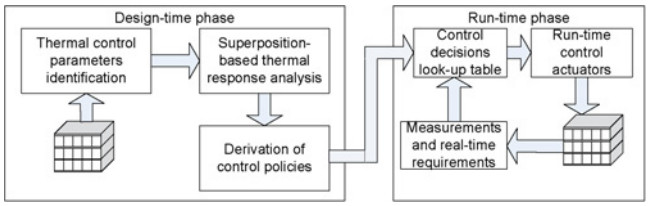

Fig. 4. Schematic diagram of the combined design-time/run-time thermal management scheme.

IV. Combined Design-Time/Run-Time Thermal Management Strategy

Designing an efficient multiobjective run-time thermal management technique for liquid-cooled 3-D systems requires the study of the target system at design-time. Fig. 4 shows a schematic diagram of our combined design-time/run-time thermal management scheme. It starts at design-time (see Section V) with the identification of the different control knobs, along with the definition of the ranges of values these knobs can have. Hence, our choices on the controlled parameters are highly flexible, unlike various assumptions taken in previous works [10], [43]. After the identification of control knobs, we perform a superposition-based thermal impact analysis of each knob to quantify its individual thermal impact on the whole system and to develop a suitable multiobjective run-time thermal scheme (Section VI). In particular, we identify the following thermal control parameters to be explored for DTM in 3-D MPSoCs.

A. Dynamic Voltage and Frequency Scaling

This technique has been used for 2-D/3-D MPSoCs DTM. DVFS has a fast response time (μs range, 100–200 μs) with respect to the other DTM techniques, but it can also imply a significant performance overhead [11].

B. Task Scheduling and/or Migration

Job scheduling is an effective tool for reducing and balancing MPSoC temperature [9], [44], [45]. This technique has a lower control decision frequency (fewer control actions per unit time) with respect to DVFS, as it relies on higher-level OS-based decisions, which are taken on the order of tens of milliseconds.

C. Variable Fluid Flow Rate

Interlayer liquid cooling plays a major role in DTM of 3-D stacked MPSoCs [10], [23]. Applying variable flow rates changes the interlayer thermal resistance and creates the effect of having intermediate heat sinks in between tiers in 3-D architectures [6], [7] without performance degradation. However, interlayer liquid cooling has a large response time as it relies on mechanical changes of the pumping network.

After performing the design-time thermal analysis, we derive a set of rules that are used in run-time thermal management. We use fuzzy logic to derive the run-time control actions. Although the same approach can be adapted to use different cyber-physical controllers, we opt to use rule-based fuzzy controller instead of other linear multi-input multi-output (MIMO) controllers [13] as with this technique, we are able to achieve effective control with a straightforward, low-complexity, and flexible implementation. Various low-complexity techniques can be used for deriving the fuzzy rules, such as offline analytical analysis and online learning mechanisms [37], and fuzzy control can be implemented at the software-level with low overhead, as we show in Section VII. Moreover, fuzzy control operates efficiently at run-time with inputs that have a degree of uncertainty in describing the system state [26], which is the case in 3-D MPSoCs where various inputs can be affected by a number of conditions (e.g., ambient temperature changes, unexpected workloads, temperature sensors inaccuracy, and stack degradation).

Overall, using fuzzy control for 3-D MPSoCs with interlayer liquid cooling with time-varying liquid flow rate is a low-cost yet effective approach to find an optimal solution [42], in comparison to other MIMO control techniques for linear time-varying systems.

V. Thermal Response Analysis in Liquid-Cooled 3-D MPSoCs

In this section, we analyze the thermal response and impact of each of identified thermal control knobs (Section IV) on 3-D MPSoC designs with respect to the worst case and typical operating conditions. In the analysis of DVFS and varying flow rate, we model an infinite thread input with full utilization. The threads are executed on a variable number of active cores in our 3-D test-bed, which is shown in Fig. 5(c). Our 5-tier 3-D test chip prototype (presented in [7] and [34]) is shown in Figs. 5(a) and (b). The 3-D MPSoC we use for the analysis consists of two tiers, where each tier contains four main hot spot sources modeling high performance processors. The remaining area contains background heaters playing the role of caches, interconnects, and other blocks. We have experimentally validated the transient thermal response of our test-bed using 3-D test chip stacks [Figs. 5(a) and (b)] with a die area of 1 cm^2. In this test-bed, the hot spot sources occupy an area of 5 mm$\times$2 mm and dissipate 250 W/cm^2, while the residual area dissipates 50 W/cm^2.

A. Variant Liquid Flow Rate

We first examine the effect of changing the liquid flow rate when the maximum temperature of the 3-D chip exceeds a certain threshold. This experiment involves the dynamic variation of the pumping flow rate (i.e., between 0.1–0.2 l/min) [7] to observe the variation of the core temperature with respect to the change in flow rate. We use five different flow rate values: {0.1, 0.125, 0.15, 0.175, 0.2} l/min. Fig. 6 shows

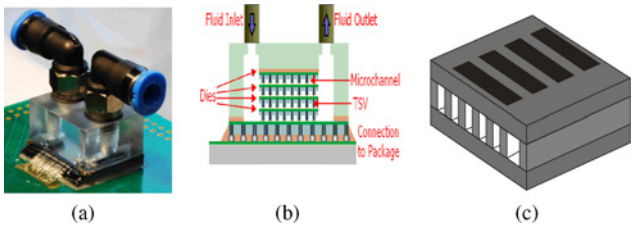

Fig. 5. Manufactured (a) prototype, (b) cross-section, and (c) layout of the test-bed we use in this exploration. The 3-D test-bed (c) has four hot spots sources (black), liquid microchannels (white), and background heaters (gray).

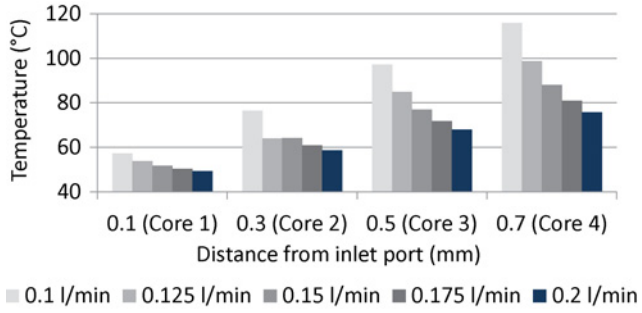

Fig. 6. Thermal change with respect to the distance from the inlet port of the cores at different flow rates. All cores are active simultaneously.

the thermal response of cores located as a function of the distance from the fluid inlet port and the flow rate. This figure illustrates that changing the liquid flow rate has a larger effect on reducing the temperature for the cores that are further from the inlet port. In fact, this thermal reduction reaches values up to 40 °C when the flow rate is changed from the minimum to maximum values because the flow thermal development depends on the flow rate. When the flow rate is increased, the fluid gains velocity and spends less time in the microchannel, thus causing the fluid to have a lower temperature at the outlet port in comparison to using a smaller flow rate value. As a result, the fluid absorbs more heat flux from the cores closer to the outlet port, helping to reduce their temperatures.

Based on these results, it is highly beneficial to increase the flow rate when the cores located away from fluid inlet have high temperatures, enabling temperature reduction regardless of the activity of the other cores.

In addition, regarding thermal hot spots, Fig. 6 indicates that cores closer to the inlet ports have the lowest temperature. In fact, since the liquid has a lower temperature when it is injected in the microchannel, the thermal gradient between the chip and the liquid is larger and more heat can be absorbed by the liquid. However, this gradient is reduced when the liquid flows in the microchannel toward the outlet port due to the thermal development of the fluid. Thus, interlayer liquid cooling reduces the need for applying other thermal management techniques on cores closer to the inlet port.

B. Task Scheduling and Migration

In our second set of experiments, we explore the influence of task migration and task scheduling on 3-D MPSoCs temperature. Thermally-aware task migration involves moving a task (or set of tasks) from hotter units to the cooler processing elements [9], [25].

In this exploration, we experiment with a scenario where four new threads are assigned to the cores of the upper layer

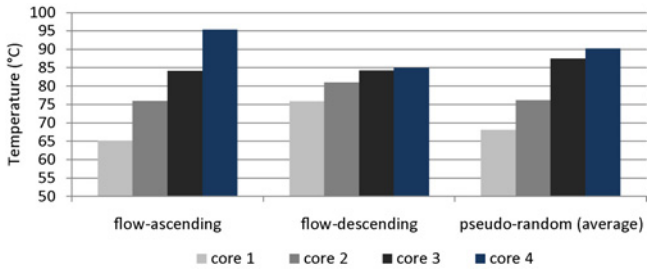

Fig. 7. Thermal response of processing cores using three task allocation configurations: flow-ascending, flow-descending, and pseudo-random. Core 1 is the nearest and core 4 is the furthest from the inlet port.

of our 3-D test-bed, while the lower layer cores are working at full utilization. Each core in the upper layer can be assigned one of the following four different utilization rates, i.e., 90%, 70%, 40%, or 10%. This configuration generates 24 different thread assignment configurations. We classify these configurations into three sets, i.e., flow-ascending, flow-descending, and pseudo-random. The flow-ascending set includes all the assignment patterns where the cores located nearer to the fluid inlet are assigned the lower utilization tasks, whereas the flow-descending set has the opposite configuration. The pseudo-random set contains the remaining task assignment patterns. Fig. 7 shows the maximum temperature variations obtained at each core among the different configurations and indicates that the minimum temperature is achieved when the threads with higher utilization rate are assigned to the cores nearer to the fluid inlet port.

In addition, in Fig. 7 we observe that the thermal gradient across the chip from the inlet to the outlet gets affected by the thread assignment. In fact, flow-descending achieves the minimum gradient among cores. Thus, we conclude that in presence of interlayer liquid cooling, the cores closer to the fluid inlet can be assigned a significantly larger task set without creating thermal issues in the 3-D stack.

Based on the previous analysis, we propose the flow-aware 3-D load balancing (FALB) technique, a novel scheduling technique for 3-D MPSoCs with liquid cooling, where the tasks requiring higher throughput are allocated on the processing elements based on their distance to the fluid inlet port. This technique extends the previously proposed temperature-aware load balancing (TALB) [10] technique for 3-D MPSoCs with interlayer liquid cooling. In particular, since we know in advance that the current temperature at a given time is closely related to the relative distance to the fluid inlet port, FALB uses this distance (X) to calculate the weight ($\omega^i_{\text{Thermal}}(X)$) in the weighted queue length (l^i_{weighted}) that is derived from the actual task-queue length (l^i_{queue}) as follows:

$$l^i_{\text{weighted}} = l^i_{\text{queue}} \cdot \omega^i_{\text{Thermal}}(X). \tag{11}$$

This new weight requires lower design-time effort (i.e., constant time) in comparison to the TALB [10], as the calculation mechanism is only dependent on the layout of the 3-D MPSoC and the porous media structure.

C. Dynamic Voltage and Frequency Scaling

In the third set of experiments, we explore applying DVFS in 3-D stacks with liquid cooling. Although DVFS reduces

the speed of the cores and can imply a degradation of the throughput for real-life workloads (see Section VII), in this analysis, we consider the execution of a thread of an infinite duration. The performance degradation is not taken into account as the main goal is to explore the thermal impact in this experiment.

To examine the thermal effects of DVFS, we use a simple two-threshold policy, where the frequency of a certain core is decreased when the temperature exceeds a high threshold value T_1, and increased when the temperature falls below another value T_2, where $T_1 > T_2$. We assume that there are three voltage/frequency (VF) settings for each processing element, as proposed in [11]. The DVFS settings are (V, F), (0.91 V, 0.83 F), and (0.83 V, 0.67 F). We compute the VF switching frequency for three different (T_1, T_2) pairs: [(77, 73), (80, 78), (85, 82)] °C.

Our experiments indicate that core location plays an important role in the use of DVFS for temperature control, as shown in Fig. 8. Cores located near the fluid inlet port do not experience a significant number of VF changes due to their very low temperature. However, as cores are located further from the inlet port, more frequent scaling occurs to maintain the temperature of these elements within the defined thresholds. In fact, when the thermal control thresholds are at the lowest range (77, 73 °C), middle-distance elements (i.e., 0.5 mm from the input port) change their VF values 50% of the possible switching moments, while the elements located furthest from the inlet port (i.e., 0.7 mm) cannot maintain their temperature within the defined thresholds, and they scale-down their VF to the lowest value. In addition, cores near the outlet port have higher temperatures than the threshold, thus VF changes occur less frequently.

When the temperature control threshold is higher (85, 82 °C), the temperatures of the cores near the outlet port get close to these thresholds, thus they perform DVFS more often with respect to the cores closer to the inlet port. Also, as shown in Fig. 8, the cores located at 0.7 mm from the inlet port have fewer VF changes in comparison to the cores at 0.5 mm. This behavior is related to the impact of VF scaling on these cores: when the VF is scaled down to reduce the temperature, DVFS has a higher impact on the cores closer to the inlet port than those further from the inlet port. As a result, the time needed to cool down the cores near the inlet port (0.5 mm) to a temperature below the 82 °C threshold is shorter in comparison to the cores at a further distance, which triggers VF scaling to higher levels sooner in closer-to-inlet cores.

VI. Integrated Flow Rate and DVFS Fuzzy Controller

This section provides the details of our fuzzy controller, which combines DVFS and dynamic flow rate management to achieve low operating temperatures while saving energy and maintaining the desired performance. In our system, we take the control decisions based on two dynamically changing inputs, temperature (T) and workload utilization (U), and on the relative distance from the inlet port as a third static input (D).

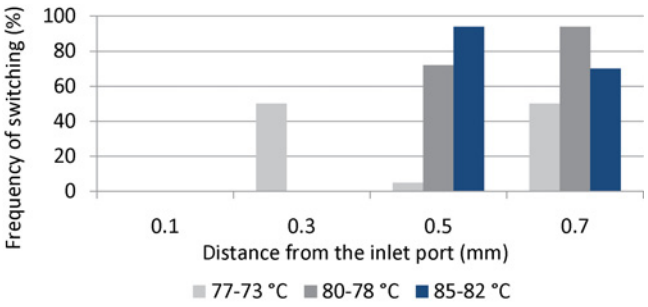

Fig. 8. VF switching frequency of processing cores with respect to the distance from the fluid inlet, using three different temperature threshold sets.

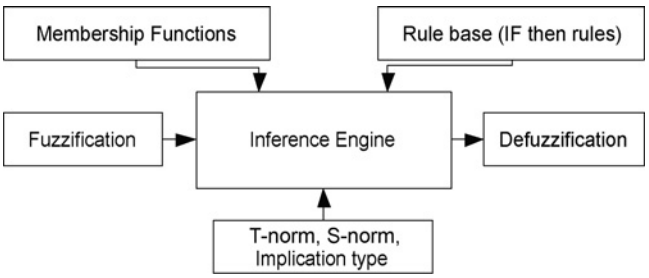

Fig. 9. Schematic diagram of the fuzzy-based thermal controller.

A. Controller Architecture

Our fuzzy controller is a Takagi-Sugeno fuzzy model [38], which defines the rules as a function of the inputs. In this fuzzy model, the rules are given using the following form:

$$\text{IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ ... AND } x_k \text{ is } A_{ik}$$
$$\text{THEN } Y_i = f_i(x_1, x_2, \ldots, x_k), \ i = 1, 2, \ldots, n$$

where $x_j | j = 1, 2, \ldots, k$ are the inputs, A_{ij} values refer to the predefined ranges, and Y_i is the output function. A schematic diagram of the building blocks of the fuzzy controller is shown in Fig. 9. We next discuss the components of the fuzzy controller.

1) *Fuzzifier:* Fuzzifier is the interface between the fuzzy controller and the outside world, as it transforms any numerical input to its corresponding fuzzy value. For any variable ($x = x_o, x_o \in R$) where R is the range of x, a fuzzy function $\mu_o(x)$ is generated to be used with the rule-base in order to infer the proper output. In our controller, we deploy a singleton fuzzifier [26], which is suitable in our context due to its low implementation complexity. For run-time thermal management, the fuzzy function model is:

$$\mu_o(x) = 1 \text{ when } x = x_o, \ 0 \text{ otherwise.}$$

2) *Fuzzy Membership Functions:* Membership functions translate a variable numerical range R to a linguistic one, such that $\{\forall x \in R, \mu(x) \in [0, 1]\}$. In the membership function selection, a key aspect is full coverage of the input variable range R with N fuzzy functions [26], such that $\{\forall x \in R, \bigcup_{i=1}^{N} \mu_i(x) > 0\}$. In our fuzzy-based thermal controller, each variable has full range coverage through three membership functions. We use triangular and trapezoidal-based memberships to minimize the controller's execution complexity [26] via these piecewise linear functions, as it is

TABLE III
FUZZY-DERIVED RULE-BASE

IF			THEN	
D Is	AND T Is	AND U Is	VF Is	AND Flow Rate Is
L	X	X	H	L
M	L	X	H	L
M	M	L	L	L
M	M	M	M	M
M	M	H	M	M
M	H	L	L	L
M	H	M	M	M
M	H	H	M	H
H	L	X	H	L
H	M	L	L	L
H	M	M	M	L
H	M	H	H	M
H	H	L	L	M
H	H	M	L	H
H	H	H	M	H

X is a "don't care."

illustrated in the upper graph in Fig. 10. This figure shows the low (L), medium (M), and high (H) membership functions used to describe the core utilization rates and temperature values. These functions, as shown in the figure, cover the full range of the measured input variables.

3) *Rule-Base:* This module is the basic building block in the controller, which contains all the IF-THEN rules relating the outputs to the inputs. We present these rules in Section VI-B.

4) *Inference Engine:* This block is used in the evaluation of the output functions implied by the rule-base, the membership functions, and the fuzzified inputs [26].

5) *Defuzzifier:* This block transforms the outputs from the inference engine into numerical values. Since we use a Takagi-Sugeno type controller, the defuzzifier applies a weighted sum from each implied IF clause to compute the output [26]. Thus, the output is defined as follows:

$$y = \frac{\sum_{i=0}^{n} A_i f_i}{\sum_{i=0}^{n} A_i} \tag{12}$$

where f_i is the fuzzy output in rule i, and A_i is the rule evaluation (the IF clause evaluated value of rule i) [38].

B. Rule-Base Derivation

Knowledge acquisition is an important block in the design of any fuzzy controller, since it is used to derive the most suitable rule-base for the fuzzy inference engine [26]. This acquisition can be achieved by utilizing expert knowledge or by other techniques (e.g., genetic algorithms [37]). In our derivation, we rely on the offline thermal response analysis presented in Section V to observe how each processing element is affected by each thermal control knob, and our controller output variables are: the flow rate and the VF settings.

Following the analysis of Section V for 2-tier and 4-tier 3-D MPSoCs using the 3-D ICE thermal simulator [34], we derive the complete rule-base shown in Table III. In our case, the rules functions, i.e., $f_i(x_1, x_2, \ldots, x_k)$, are constant values, where they are expressed as follows: H corresponds to the maximum range of values applicable to a certain variable, M is the mean range, and L is the minimum value of the range for the variable.

As this table shows, in 2-tier and 4-tier 3-D MPSoCs, the cores closer to the inlet port (D is L) have the lowest temperature, and overall they do not require VF changes (i.e., VF is always H) or the liquid flow rate (flow rate is always L). However, when the processing elements are located at a further location (D is M) from the input port, we need to monitor their current state and adapt accordingly. For instance, if the temperature of a core is low (T is L), no change is required in either VF or flow rate settings (VF is High, flow rate is Low). On the contrary, if the temperature reaches the medium range (T is M), the utilization rate plays a role in the controller decision. If the utilization is low (U is L), VF and flow rate should be reduced to the minimum setting to minimize energy and thermal variations (VF is L AND flow rate is Low). Moreover, increasing the flow rate could reduce the temperature of any core at any state, but using such method implies a larger energy overhead, which may not always be the optimal control action for an energy efficient controller. Thus, we increase the flow rate as a last resort, only when we cannot mitigate the hot spots with other techniques at acceptable performance levels. In addition, Table III outlines that our fuzzy logic controller selects the appropriate VF value at every state to enable fine-grained thermal control along with minimal performance degradation.

C. System Stability with Fuzzy Controller

From the description of the proposed fuzzy controller, the output $y(k+1)$ can be expressed as follows:

$$y(k+1) = \frac{\sum_{i=0}^{n} mp_i(k) f_i}{\sum_{i=0}^{n} mp_i(k)} \tag{13}$$

where $mp_i(k)$ stands for the outcome of the IF clause in each rule. We express this outcome with $T(\mu(x_1), \mu(x_2), \ldots, \mu(x_k))$, where the T operator resembles the T-norm [26] performed. We prove that the controller output is bounded between the lowest and the highest values of such output as follows.

To estimate the output range obtained, we show the input membership functions, which are similar to the functions in Fig. 10. We define these functions by three major points (a, b, c), thus dividing the domain area into four distinct regions. To simplify the analysis, we explore the stability using two variables only, but the procedure is valid for any number of variables. Fig. 10 shows the extended 2-D domain of temperature and utilization membership functions. For brevity, we derive the stability analysis on a subset of the rules derived in Table III where the distance (D) is medium (M), but this analysis is applicable to the whole set of rules.

By observing the four corner regions in Fig. 10, it is clear that the resulted T-norm (of the IF part) equals 1 for either low (L) or high (H) membership function [T(1, 1)=1] [26]. Thus, the output variables get the exact values assigned within the associated rules.

Then, if we consider a more complex case in the regions that have more than a single active membership function, this case leads to having more than a single fired rule (i.e., having a nonzero evaluated IF clause). Thus, the output is evaluated from the different fired rules using the defuzzifier. We consider the case where $b_T < T_o < c_T$ and $a_U < U_o < b_U$ (see Fig. 10).

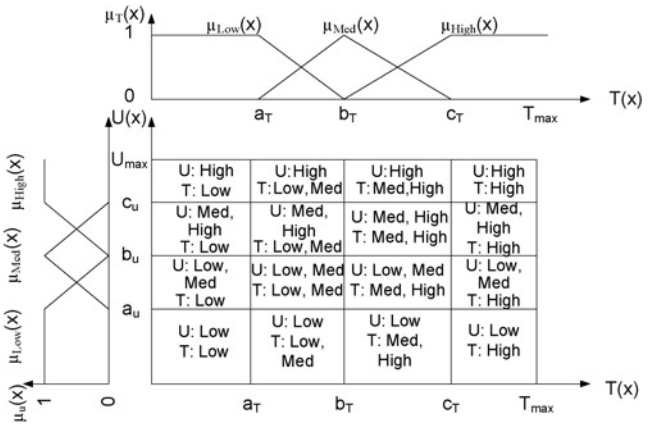

Fig. 10. Domains of two input variables (U and T) in our fuzzy controller.

Based on these ranges, the membership functions have the values as follows:

$$\mu_{\text{Low}}(T_o) = 0 \tag{14}$$

$$\mu_{\text{Medium}}(T_o) = \frac{T_o - c_T}{b_T - c_T} \tag{15}$$

$$\mu_{\text{High}}(T_o) = \frac{T_o - b_T}{c_T - b_T} \tag{16}$$

$$\mu_{\text{Low}}(U_o) = \frac{T_o - b_T}{a_T - b_T} \tag{17}$$

$$\mu_{\text{Medium}}(U_o) = \frac{T_o - a_T}{b_T - a_T} \tag{18}$$

$$\mu_{\text{High}}(U_o) = 0. \tag{19}$$

From these previous values, we observe that no rule using the membership functions $\mu_{\text{Low}}(T)$ or $\mu_{\text{High}}(U)$ in the *IF* clause is fired (activated). Hence, we can calculate the control output VF settings as follows:

$$VF(k+1) = \frac{\sum T(\mu_i(T_o), \mu_j(U_o))VF_k}{\sum T(\mu_i(T_o), \mu_j(U_o))} \tag{20}$$

$$i \in [\text{Medium, High}] \tag{21}$$

$$j \in [\text{Low, Medium}] \tag{22}$$

$$k \in [L, M]. \tag{23}$$

Due to the structure of membership functions, where there is always a nonzero value of some function at any instance, it is guaranteed that at least a single nonzero T-norm exists in the aforementioned output formula. Hence, the output is bounded between VF_L and VF_M.

Similarly, by performing the full analysis on all variable ranges, we find that the outputs are always bounded, such that, $VF_L < VF < VF_H$ and $FL_L < FL < FL_H$. Therefore, the proposed controller output is bounded between the minimum and the maximum values. Consequently, we can apply the stability criteria of [4] and [39] to demonstrate the stability of the system.

VII. Experimental Results

A. Experimental Setup

The 3-D MPSoCs we use in our experiments are based on the 90 nm UltraSPARC T1 (Niagara-1) processor [21]. The power consumption, area, and the floorplan of UltraSPARC T1 are available in [21]. UltraSPARC T1 has eight multithreaded cores, and a shared L2-cache for every two cores. Our simulations are carried out with 2-tier, and 4-tier stack architectures.

TABLE IV
Workload Characteristics

	Benchmark	Average Util (%)	L2 I-Miss	L2 D-Miss	FP instr
1	Web-med	53.12	12.9	167.7	31.2
2	Web-high	92.87	67.6	288.7	31.2
3	Database	17.75	6.5	102.3	5.9
4	Web and DB	75.12	21.5	115.3	24.1
5	MPlayer	6.5	9.6	136	1
6	MPlayer and Web	26.62	9.1	66.8	29.9

First, we gather workload characteristics of real applications on an actual UltraSPARC T1. We sample the utilization percentage for each hardware thread at every second using `mpstat`, and record half an hour long traces for each benchmark. Also, the length of user and kernel threads are recorded using `DTrace` [24]. We use various real-life benchmarks including web server, database management, and multimedia processing. A detailed summary of the workloads is given in Table IV. The utilization ratios are averaged over all cores throughout the execution. We also record the cache misses and floating point (FP) instructions per 100 K instructions using `cpustat`. The workload statistics collected on the UltraSPARC T1 are replicated for the 4-tier 16-core system.

The peak power consumption of SPARC is close to its average value [21]. Thus, we assume that the instantaneous dynamic power consumption is equal to the average power at each state (active, idle, sleep). The active state power is taken as 3 W [21]. The cache power consumption is 1.28 W per each L2, as computed by CACTI [40] and verified by the values in [21]. We model the crossbar power consumption by scaling the average power value according to the number of active cores and the memory accesses (i.e., the more the active cores at a specific time instance, the higher the crossbar power is at that time instance). To account for 3-D integration, we scale down the crossbar power to account for the reduction in wire length due to vertical stacking.

The leakage power of processing cores is dynamically calculated according to the structural areas of the components and their actual run-time temperatures. We assume a base leakage power density of 0.25 W/mm^2 at 383 K for 90 nm technology [5]. To account for temperature effects on leakage power, we use the second-order polynomial model proposed in [36]. We empirically determine the coefficients in the model to match the normalized leakage values shown in [36].

For implementing DVFS, three voltage and frequency scaling values are used in our simulations: [(1.2, 1.2), (1.1, 1.0), (1.0, 0.8)]. The values are shown in pairs (V, F), with V in *Volts* and F in GHz. When there is a change in F at run-time, the thread utilization U is changed to U' using a scaling assumption $U' = U\frac{f_{\text{old}}}{f_{\text{new}}}$. Thus, performance degradation is taken into account in our simulations.

Current MPSoCs typically have power management capabilities to reduce the energy consumption [5]. Thus, we implement dynamic power management (DPM) to investigate the effect on thermal variations. We utilize a fixed timeout

TABLE V
THERMAL AND FLOORPLAN PARAMETERS DEPLOYED IN THE MODEL

Parameter	Value
Silicon conductivity	130 W/(m · K)
Silicon capacitance	1635660 J/(m^3· K)
Wiring layer conductivity	2.25 W/(m· K)
Wiring layer capacitance	2174502 J/(m^3· K)
Water conductivity	0.6 W/(m· K)
Water capacitance	4183 J/(kg·K)
Heat sink conductivity	10 W/K
Heat sink capacitance	140 J/K
Die thickness (one stack)	0.15 mm
Area per core	10 mm^2
Area per L2 cache	19 mm^2
Total area of each layer	115 mm^2
Interlayer conductivity (channel walls)	160 W/(m · K)
Interlayer capacitance (channel walls)	1641101 J/(m^3· K)

policy, which puts a core to sleep state if it has been idle longer than the timeout period (i.e., 200 ms in our experiments). We set a sleep state power of 0.02 W, which is estimated based on sleep power of similar cores [10].

In the thermal modeling tool (3D-ICE), we use a sampling interval of 100 ms, and all simulations are initialized with steady-state temperature values. The model parameters are provided in Table V. This table contains the thermal conductance and capacitance of the materials in the stack. The modeling methodology for the interlayer material to include TSVs and microchannels is described in Section III. In our experiments, we assume TSVs are homogeneously distributed across the die, which is in line with the actual locations of the TSVs as we discuss in the earlier sections. Moreover, we compare AC and liquid-cooled 2-tier and 4-tier 3-D MPSoCs.

We assume that each core has a temperature sensor, which is able to provide temperature readings at regular intervals (e.g., 100 ms). Modern operating systems have a multiqueue structure, where each CPU core is associated with a dispatch queue, and the job scheduler allocates the jobs to the cores according to the current policy. In our 3-D MPSoC thermal simulator, we implement a similar infrastructure, where the queues maintain the threads allocated to cores and execute them.

B. Fuzzy Controller Implementation

Our fuzzy controller operates as an interrupt-driven software routine. As shown in Fig. 11, this routine is triggered at every temperature sampling interval (100 ms), where temperature values are acquired from the thermal sensors, and core utilization values are computed based on the workload allocated onto each core's queue. After these inputs are acquired, the fuzzy controller is enabled and it selects the appropriate VF value to each processing element, as well as the desired flow rate.

We measure the computational and delay overhead induced by this control routine using the same methodology mentioned before (using `mpstat` and `time` commands), and we find that this routine introduces 120 K additional clock cycles per single routine run. By mapping this application to the targeted 3-D MPSoC, this routine introduces an additional 0.1 ms processing activity. The controller can also be implemented using a dedicated hardware module to further improve performance.

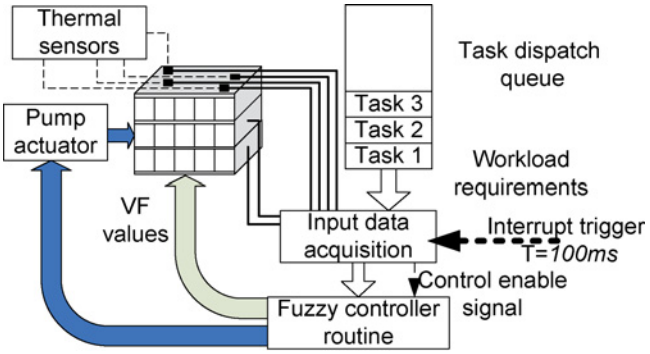

Fig. 11. Schematic diagram showing the fuzzy controller action as an interrupt-triggered software routine.

C. DTM Policies for 3-D MPSoCs

We implement various thermal management techniques to evaluate the thermal and energy efficiency of the proposed fuzzy thermal management technique(LC_FUZZY). We experiment with both air-cooled (AC) and liquid-cooled (LC) 2-tier and 4-tier 3-D MPSoCs for comparison purposes. The techniques we implement are as follows.

1) *Dynamic load balancing (AC_LB)* [14], [25]: Some form of load balancing exists in most OSes today. LB balances the workload by moving threads from a core's queue to another if the difference in queue lengths is over a threshold.
2) *Temperature-triggered task migration (AC_TTMIG)* [12]: It migrates tasks from a core if that core exceeds the threshold temperature (85 °C in our case). In this paper, we assume a 1 ms overhead when a thread is migrated to a new core.
3) *Temperature-triggered DVFS (AC_TDVFS)* [9]: It scales down the VF settings of a core when the core's temperature exceed the 85 °C threshold value. In our implementation, as long as the temperature is above the threshold and there is a lower setting, we reduce the VF level at every DVFS interval. When the temperature falls below another threshold value (82 °C), we increase the voltage frequency setting by one step.
4) *TTMIG and TDVFS (AC_MIG_VF)*: It combines TDVFS and TTMIG into a joint policy.
5) *Liquid cooling with LB (LC_LB)*: It applies the maximum flow rate (0.0323 l/min per cavity), while the jobs are scheduled with LB.
6) *Liquid cooling with FALB (LC_FALB)*: It has same flow rate as LC_LB, while the jobs are scheduled with flow-aware 3-D load balancing introduced in Section V-B.
7) *LUT-based flow rate control with LB (LC_VAR)* [10]: It changes the flow rate (between 0.01–0.0323 l/min per cavity) based on the predicted maximum temperature, while the jobs are scheduled with LB.
8) *LUT-based flow rate control with FALB (LC_VAR_FALB)* [10]: It has same flow rate adjustment as LC_VAR_LB, but the jobs are scheduled with flow-aware 3-D load balancing as discussed in Section V-B.

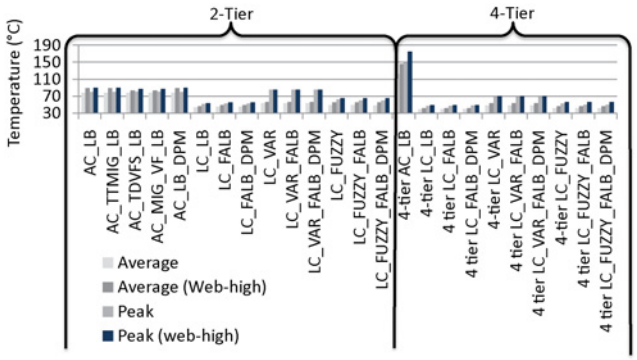

Fig. 12. Peak and average temperatures for all the policies, both for the average case across all workloads and for Web-high, the hottest benchmark, for 2-tier and 4-tier 3-D MPSoCs.

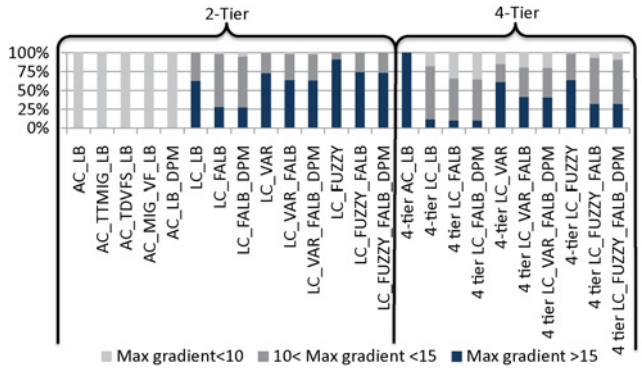

Fig. 13. Maximum thermal gradient of the whole stack, using the average case of all workloads.

D. Transient Temperature Evaluation

The thermal impact of all the policies on the 2-tier and 4-tier 3-D MPSoC is shown in Fig. 12. This figure compares peak and average temperatures for the average case across all the workloads and also for the benchmark with hottest temperatures (Web-high). TTMig and TDVFS help reduce the hot spots in the 2-tier AC systems, while the integration of liquid-cooling removes all the hot spots in 2-tier and 4-tier systems. In the 2-tier AC 3-D MPSoC, the peak temperature with LB is 87 °C, while the peak temperatures of TTMig and TDVFS are 85–86 °C. However, in the 4-tier AC 3-D MPSoC, we observe that the maximum temperature of the system is reaching up to 178 °C. The increase in temperature of 4-tier AC system is a consequence of the increased stacking of tiers with limited cooling capabilities, hence leaving little opportunity for any thermal management technique to successfully control the hot spots without severely degrading the performance. Thus, in the following experiments we focus on the 4-tier 3-D MPSoCs with liquid cooling to compare the DTM policies.

Interlayer liquid cooling has the ability to absorb the heat flux between different tiers surrounding the cooling layer. In the 2-tier system (with 1-cooling layer), LC_LB reduces the peak temperature to 56 °C, whereas LC_FUZZY pushes the system into a higher peak of 68 °C, but still avoids hot spots. However, there is a small hot spot threshold violation when LC_VAR is used, where the temperature reaches 85 °C. This violation is due to the limited flow rate values we are using in this simulation. When the minimal flow rate value is set during a high utilization phase, the temperature may exceed the thermal-threshold. Although LC_VAR predicts this violation, the time needed to adjust the flow rate is not small enough to prevent the maximal temperature to reach 85 °C.

In the 4-tier 3-D MPSoC with three liquid cooling layers, the maximum temperature is maintained at an even lower value in comparison to the 2-tier system in all three techniques with LC. This is due to the increased number of channel layers and better cooling capability.

E. Thermal Gradients

It is important to reduce the large thermal gradients on a system to maximize cooling efficiency and system reliability. We calculate the maximum thermal gradient in the whole stack as well as the average intralayer thermal gradient of the different layers in the stack, which are shown in Figs. 13 and 14, respectively. From these figures, we observe that the maximum thermal gradient in AC systems increases with respect to the number of stacked elements. In fact, the maximum thermal gradient for 4-tier, AC 3-D MPSoC exceeds 15 °C with values reaching up to 60 °C for the Web-high benchmark. This increase in thermal gradient is correlated with the cooling technique, which includes single heat sink on the top layer, hence creating a huge thermal gap between tiers of different relative distances with respect to the heat sink. The average intralayer gradient of the 4-tier, AC 3-D MPSoC is maintained below 10 °C, which implies a much more uniform thermal distribution within a single layer.

Although interlayer liquid cooling mitigates the thermal hot spots, it increases both the intralayer and the maximum thermal gradient of the stack. This is due to the fact that the fluid develops thermally from the inlet to the outlet such that the elements near the inlet have more heat removed than the ones at the outlet. This is clearly shown in the 2-tier 3-D MPSoC where a single cooling layer is present and the thermal gradient (both maximum and average) exceeds 15 °C (up to 22 °C) in Figs. 13 and 14.

We examine several techniques to avoid this issue. We find that increasing the number of interlayer cooling layers aids in reducing the thermal gradient. Using two cooling layers, the 2-tier stack has a thermal gradient value less than 5 °C. Also, the additional cooling layer in the 4-tier system reduces the average gradient (Figs. 13 and 14) to smaller values in comparison to the gradients in the 2-tier system with one cooling layer.

We also evaluate the new FALB introduced in Section V-B for its effect in reducing thermal gradients. By applying FALB, the thermal gradients of the systems are significantly reduced, implying a better thermal balance (Figs. 13 and 14). *FALB* reduces the large intralayer thermal gradients (i.e., larger than 15 °C) of the 2-tier 3-D MPSoC to 50% of the time in comparison to the 80% frequency when using LC_FUZZY only.

F. Energy Consumption and Performance Overhead

Fig. 15 shows the total energy consumed and the performance overhead when running the policies on the 2-tier and 4-tier stacks for the average case. Energy consumption values are normalized with respect to the load balancing policy on a system with air cooling.

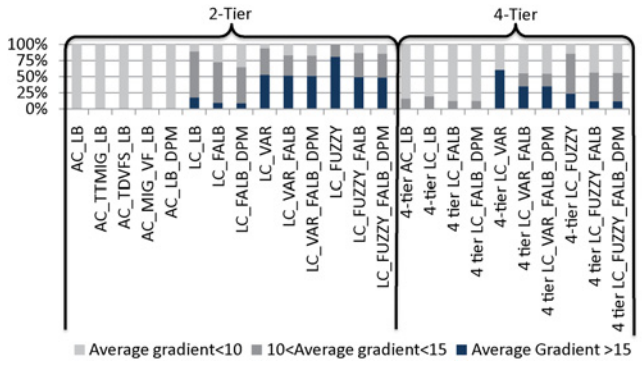

Fig. 14. Average intra-layer thermal gradient of the whole stack, using the average case of all workloads.

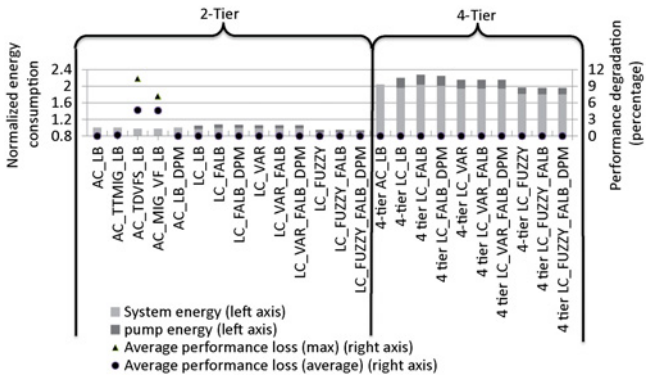

Fig. 15. Left axis shows the energy consumption in the whole system (chip and cooling network) for average workload, per stack. The right axis shows the % delay for each policy. Note that air cooling also includes fan power consumption, which is not included in the figure.

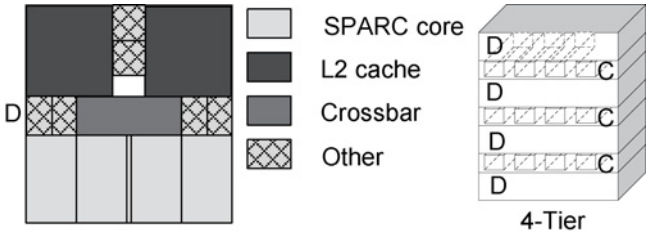

Fig. 16. Layout of the adjusted floorplan used in 4-tier 3-D MPSoCs.

Note that in LC_LB case, only a single flow rate is used. Thus, there is no need for control valves to alter the flow rate. Therefore, in our energy consumption computation, we calculate the cooling energy in LC_LB as the pumping power only, without any valve power.

The fuzzy controller achieves major reductions in both the coolant and the overall system energy consumption: LC_FUZZY reduces the system energy (2-tier 3-D MPSoC) by 10.3% and 9.5%, as well as the cooling energy by 35% and 26% at average workloads in comparison to LC_VAR and LC_LB, respectively. The reason LC_FUZZY outperforms all other techniques in energy savings is due to the joint control of flow rate and DVFS at run-time based on each core's thermal and utilization status. At lower utilization levels and lower temperatures, the integrated system leaks less energy than that of a higher thermal profile.

Our fuzzy controller reduces energy consumption while maintaining the thermal benefits of liquid cooling. In the 4-tier case, LC_FUZZY achieves 27% and 8% *average* coolant and overall system energy savings with respect to LC_VAR. Moreover, LC_FUZZY achieves 63% and 21% *peak* savings in comparison to LC_LB, as well as 45% and 18% *peak* savings in comparison with LC_VAR. The substantial increase in energy savings with respect to the 2-tier system is due to the higher number of cavities in the stack. With LC_LB and three existing cavities (as in Fig. 1), a flow rate of 3×0.0323 l/min is required for the whole 3-D MPSoC, which in turn increases the required cooling power. However, with LC_FUZZY and three cavities, the flow rate can be lowered to minimal values while allowing higher workload utilization than that of 2-tier 3-D MPSoC. This is due to the fact that the increased number of cavities enhances the interlayer cooling capabilities, as discussed earlier. Thus, more heat flux is absorbed from different directions leading to lower temperatures.

Finally, we explore the benefits of DPM to reduce the system energy. Fig. 15 shows that the benefits we gain from using DPM with LC_FUZZY and FALB (LC_FUZZY_FALB_DPM) are very minute with respect to LC_FUZZY_FALB. In fact, the average savings achieved by DPM is 1%, while the maximum savings reaches 5% with MPlayer benchmark.

For our multicore 3-D MPSoCs, we compute throughput as the performance metric. The performance degradation of the average workload under a set of policies is shown in Fig. 15. Liquid cooling-based systems do not suffer from any performance degradation since the temperature of such systems does not rise to a value where another thermal management technique should be applied. Although LC_FUZZY uses DVFS, we apply DVFS based on the core utilization, hence the performance degradation does not exceed 0.01%, which is negligible in comparison to the degradation observed in the air-cooling scenarios.

TABLE VI
MAXIMUM TEMPERATURE AND THERMAL GRADIENTS OF 4-TIER 3-D MPSOC WITH DIFFERENT FLOORPLANS

Flow Rate ml/min	Floorplan in Fig. 1			Floorplan in Fig. 16		
	Max Temp °C	Intertier Gradient °C/cm	Max Gradient °C/cm	Max Temp °C	Intertier Gradient °C/cm	Max Gradient °C/cm
32	55.65	16.06	20.78	49.59	3.3	4.85
24	62.92	21.49	27.02	53.72	4.26	6.1643
16	78.31	31.4	38.47	64.2	9.89	11.63
8	124.2	62	73.12	84.51	19.82	21.63

G. Design Considerations for 3-D MPSoCs

We have shown in our previous results that in the 4-tier 3-D MPSoC case, the maximum temperature is kept below the thermal threshold (85 °C). However, the maximum temperature is 49 °C, which shows that the system is thermally over-designed. Thus, in this section we show different design-time parameters we may use to further optimize the 3-D stacks. This design-space exploration uses LC_LB; however, same principles can also be applied in conjunction with the other DTM policies.

First, we reduce the maximum pumping flow rate to push the maximum temperature toward the thermal threshold. We reduce the flow rate and observe the peak temperature, as well as the intertier and maximum thermal gradients of the 4-tier

3-D MPSoC, when operating with the maximum workload requirements (Web-high), as shown in first column in Table VI. This table shows that a higher temperature is reached when the injected flow rate is reduced. However, this increase is also accompanied by an increase in the thermal gradient. These large thermal gradients are not preferable for the system operation, since large variations in thermal stress among different processing components lead to different degradation rates [9]. Therefore, solely reducing the maximum flow rate is an inadequate solution to push the operating temperature of the system near the thermal threshold as it may cause performance, reliability, or design challenges due to larger gradients.

Second, we investigate adjusting the floorplan to account for thermal properties of fluid flow. Using the results of Section V-A, we adjust the floorplan of the 3-D MPSoC by placing the elements with higher power elements (cores) near the fluid inlet, while allocating the lower power units (L2 caches) near the fluid outlet port. Thus, we change the floorplan from tiers *A* and *B* shown in Fig. 1 to tier *D* shown in Fig. 16.

We again reduce the flow rate for the new layout, and we observe a significant reduction in the thermal gradient with respect to the floorplan in Fig. 1, as shown in Table VI. The reason is the following: as the fluid is developing thermally across the channels, the thermal flux is decreasing closer to the fluid outlet port. This decrease in heat flux matches the cooling capabilities of the fluid at that point (near the outlet), thus the junction temperature of these elements is not increased. This behavior indicates that the fluid flow characteristics can be exploited in thermally aware floorplanning of 3-D MPSoCs, which opens up new research directions for designing 3-D MPSoCs with interlayer liquid cooling.

VIII. Conclusion

In this paper, we have presented a novel design-time/run-time thermal management strategy for 3-D MPSoCs including active cooling. We performed a design-time thermal response analysis of 3-D MPSoC varying the liquid flow rate, applying DVFS, and applying various workload utilization scenarios. We utilized this analysis in building a novel fuzzy controller that adjusted the liquid flow rate and the VF settings to balance temperature across the 3-D stack and to minimize system energy consumption while preventing thermal hot spots. We implemented a large set of thermal management strategies such as temperature-aware load balancing and dynamic power management in conjunction with our proposed controller. Our experimental results with 2-tier and 4-tier 3-D MPSoC case studies illustrated that our fuzzy controller maintained the temperature below the desired levels. At the same time, the controller reduced cooling energy by up to 63% and 45%, and system-level energy by up to 21% and 18% in comparison to setting the highest coolant flow rate to match the worst-case temperature and in comparison to using a state-of-the-art 3-D MPSoC thermal management approach, respectively.

References

[1] D. Atienza, P. G. Del Valle, G. Paci, F. Poletti, L. Benini, G. De Micheli, and J. M. Mendias, "A fast HW/SW FPGA-based thermal emulation framework for multi-processor system-on-chip," in *Proc. DAC*, 2006, pp. 618–623.

[2] D. Atienza, P. G. Del Valle, G. Paci, F. Poletti, L. Benini, G. De Micheli, J. M. Mendias, and R. Hermida, "HW/SW emulation framework for temperature-aware design in MPSoCs," *IEEE Trans. Design Automat. Electron. Syst.*, vol. 12, no. 3, pp. 1–26, Aug. 2007.

[3] M. S. Bakir, C. King, D. Sekar, H. Thacker, B. Dang, G. Huang, A. Naeemi, and J. D. Meindl, "3D heterogeneous integrated systems: Liquid cooling, power delivery, and implementation," in *Proc. CICC*, Sep. 2008, pp. 663–670.

[4] X. Ban, X. Z. Gao, X. Huang, and A. V. Vasilakos, "Stability analysis of the simplest Takagi-Sugeno fuzzy control system using circle criterion," *Inform. Sci.*, vol. 177, no. 20, pp. 4387–4409, 2007.

[5] P. Bose, "Power-efficient microarchitectural choices at the early design stage," in *Proc. PACS* (Keynote Address), 2003.

[6] T. Brunschwiler, H. Rothuizen, M. Fabbri, U. Kloter, B. Michel, R. J. Bezama, and G. Natarajan, "Direct liquid-jet impingement cooling with micron-sized nozzle array and distributed return architecture," in *Proc. ITHERM*, 2006, pp. 196–203.

[7] T. Brunschwiler, B. Michel, H. Rothuizen, U. Kloter, B. Wunderle, H. Oppermann, and H. Reichl, "Interlayer cooling potential in vertically integrated packages," *Microsyst. Technol.*, vol. 15, no. 1, pp. 57–74, 2009.

[8] A. K. Coskun, J. Ayala, D. Atienza, and T. S. Rosing, "Modeling and dynamic management of 3D multicore systems with liquid cooling," in *Proc. VLSI-SoC*, 2009, pp. 60–65.

[9] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *Proc. DATE*, 2009, pp. 1410–1415.

[10] A. K. Coskun, D. Atienza, T. S. Rosing, T. Brunschwiler, and B. Michel, "Energy-efficient variable-flow liquid cooling in 3D stacked architectures," in *Proc. DATE*, Mar. 2010, pp. 111–116.

[11] A. K. Coskun, T. S. Rosing, and K. Gross, "Utilizing predictors for efficient thermal management in multiprocessor SoCs," *IEEE Trans. Comput.-Aided Des.*, vol. 28, no. 10, pp. 1503–1516, Oct. 2009.

[12] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. DATE*, 2007, pp. 1659–1664.

[13] Y. Diao, J. L. Hellerstein, A. J. Storm, M. Surendra, S. Lightstone, S. Parekh, and C. Garcia-Arellano, "Using MIMO linear control for load balancing in computing systems," in *Proc. ACC*, 2004, pp. 2045–2050.

[14] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. ISCA*, 2006, pp. 78–88.

[15] Z. Feng and P. Li, "Fast thermal analysis on GPU for 3D-ICs with integrated microchannel cooling," in *Proc. ICCAD*, 2010, pp. 551–555.

[16] M. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, H. S. Lee, and G. H. Loh, "Multiobjective microarchitectural floorplanning for 2-D and 3-D ICs," *IEEE Trans. Comput.-Aided Design*, vol. 26, no. 1, pp. 38–52, Jan. 2007.

[17] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, amd M. J. Irwin, "Interconnect and thermal-aware floorplanning for 3D microprocessors," in *Proc. ISQED*, 2006, pp. 98–104.

[18] Y. J. Kim, Y. K. Joshi, A. G. Fedorov, Y.-J. Lee, and S.-K. Lim, "Thermal characterization of interlayer microfluidic cooling of three-dimensional integrated circuits with nonuniform heat flux," *J. Heat Transfer*, vol. 132, no. 4, pp. 1–9, 2010.

[19] C. R. King, D. Sekar, M. S. Bakir, B. Dang, J. Pikarsky, and J. D. Meindl, "3D stacking of chips with electrical and microfluidic I/O interconnects," in *Proc. ECTC*, 2008, pp. 1–7.

[20] Y. J. Lee, R. Goel, and S. K. Lim, "Multi-functional interconnect cooptimization for fast and reliable 3D stacked ICs," in *Proc. ICCAD*, 2009, pp. 645–651.

[21] A. Leon, J. L. Shin, K. W. Tam, W. Bryg, F. Schumacher, P. Kongetira, D. Weisner, and A. Strong, "A power-efficient high-throughput 32-thread SPARC processor," in *Proc. ISSCC*, Feb. 2006, pp. 295–304.

[22] Z. Li, X. Hong, Q. Zhou, S. Zeng, J. Bian, H. Yang, V. Pitchumani, and C.-K. Cheng, "Integrating dynamic thermal via planning with 3D floorplanning algorithm," in *Proc. ISPD*, 2006, pp. 178–185.

[23] K. Matsumoto, S. Ibaraki, M. Sato, K. Sakuma, Y. Orii, and F. Yamada, "Investigations of cooling solutions for three-dimensional (3D) chip stacks," in *Proc. SEMI-THERM*, 2010, pp. 25–32.

[24] R. McDougall, J. Mauro, and B. Gregg, *Solaris Performance and Tools*. Englewood Cliffs, NJ: Sun Microsystems Press/Prentice-Hall, 2006.

[25] F. Mulas, D. Atienza, A. Acquaviva, S. Carta, L. Benini, and G. De Micheli, "Thermal balancing policy for multiprocessor stream computing platforms," *IEEE Trans. Comput.-Aided Design*, vol. 28, no. 12, pp. 1870–1882, Dec. 2009.

[26] H. T. Nguyen and N. R. Prasad, *Fuzzy Modeling and Control, Selected Works of M. Sugeno*. Boca Raton, FL: CRC Press, 1999.
[27] L. D. Paulson, "IBM supercomputers heat will warm university structures," *Computer*, vol. 42, no. 9, pp. 18–21, Sep. 2009.
[28] K. Puttaswamy and G. H. Loh, "Thermal analysis of a 3D die-stacked high-performance microprocessor," in *Proc. GLSVLSI*, 2006, pp. 19–24.
[29] H. Qian, X. Huang, H. Yu, and C. Chang, "Cyber-physical thermal management of 3D multi-core cache-processor system with microfluidic cooling," *ASP J. Low Power Electron.*, vol. 7, no. 1, pp. 1–12, 2011.
[30] M. M. Sabry, A. K. Coskun, and D. Atienza, "Fuzzy control for enforcing energy efficiency in high-performance 3D systems," in *Proc. ICCAD*, 2010, pp. 642–648.
[31] *WILO MHIE Centrifugal Pump* [Online]. Available: http://www.wilo.com/cps/rde/xchg/en/layout.xsl/3707.htm
[32] *Festo Electric Automation Technology* [Online]. Available: http://www.festodidactic.com/ov3/media/customers/1100/0096636000107522-3683.pdf
[33] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. ISCA*, Jun. 2003, pp. 2–13.
[34] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3D-ICE: Fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling," in *Proc. ICCAD*, 2010, pp. 463–470.
[35] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, D. A. Alonso, "Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries," in *Proc. THERMINIC*, 2010, pp. 1–6.
[36] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full-chip leakage estimation considering power supply and temperature variations," in *Proc. ISLPED*, 2003, pp. 78–83.
[37] M. Su and H. Chang, "Application of neural networks incorporated with real-valued genetic algorithms in knowledge acquisition," *Fuzzy Sets Syst.*, vol. 112, no. 1, pp. 85–97, 2000.
[38] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, Feb. 1985.
[39] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, no. 2, pp. 135–156, 1992.
[40] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, *CACTI 4.0*, HP Laboratories, Palo Alto, CA, Tech. Rep. HPL-2006-86, 2006.
[41] D. B. Tuckerman and R. F. W. Pease, "High-performance heat sinking for VLSI," *IEEE Electron Device Lett.*, vol. 2, no. 5, pp. 126–129, May 1981.
[42] L.-X. Wang, "Stable and optimal fuzzy control of linear systems," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 137–143, Feb. 1998.
[43] F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd, "Online convex optimization-based algorithm for thermal management of MPSoCs," in *Proc. GLSVLSI*, 2010, pp. 203–208.
[44] X. Zhou, Y. Xu, Y. Du, Y. Zhang, and J. Yang, "Thermal management for 3D processors via task scheduling," in *Proc. ICPP*, Sep. 2008, pp. 115–122.
[45] C. Zhu, G. Zhenyu, L. Shang, R. P. Dick, and R. Joseph, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Trans. Comput.-Aided Design*, vol. 27, no. 8, pp. 1479–1492, Aug. 2008.

Mohamed M. Sabry (S'12) received the B.S. (summa cum laude) and M.S. degrees in electrical engineering from Ain Shams University, Cairo, Egypt in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical Engineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

He is a member of the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne. His current research interests include system design and resource management methodologies in embedded systems, and multiprocessor system-on-chips (MPSoCs), especially temperature and reliability management of 2-D and 3-D MPSoCs.

Mr. Sabry was the recipient of the Best Student Award when he was pursuing the B.S. degree.

Ayse K. Coskun (M'06) received the M.S. and Ph.D. degrees in computer science and engineering from the University of California at San Diego, San Diego.

She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Boston University (BU), Boston, MA. She was with Sun Microsystems (now Oracle), San Diego, for three years prior to her current position at BU. Her current research interests include energy-efficient computing, multicore systems, 3-D stack architectures, computer architecture, and embedded systems and software.

Dr. Coskun received the Best Paper Award at IFIP/IEEE VLSI-SoC Conference in 2009. She currently serves on the program committees of many design automation conferences including DATE, GLSVLSI, and VLSISoC. She is a member of the ACM.

David Atienza (M'05) received the M.S. degree from the Complutense University of Madrid, Madrid, Spain, and the Ph.D. degree from the Inter-University Microelectronics Center, Leuven, Belgium, in 2001 and 2005, respectively, both in computer science and engineering.

He is currently a Professor of electrical engineering and the Director of the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. His current research interests include system-level design methodologies for high-performance multiprocessor system-on-chip (MPSoC) and embedded systems, including new 2-D/3-D thermal-aware design for MPSoCs, wireless body sensor networks, HW/SW reconfigurable systems, dynamic memory optimizations, and network-on-chip design. He is a co-author of more than 150 publications in peer-reviewed international journals and conferences, several book chapters, and two U.S. patents in these fields.

Dr. Atienza has received a Best Paper Award at the VLSI-SoC 2009 conference, and three Best Paper Award Nominations at the WEHA-HPCS 2010, ICCAD 2006, and DAC 2004 conferences. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF CIRCUITS AND SYSTEMS, and *Integration* (Elsevier). Since 2008, he has been a member of the Executive Committee of the IEEE Council on EDA, and since 2010, a member of the Board of Governors of the IEEE Circuits and Systems Society.

Tajana Šimunić Rosing (M'01) received the M.S. degree in electrical engineering from the University of Arizona, Tucson, where her thesis topic was high-speed interconnect and driver-receiver circuit design, and the Ph.D. degree from Stanford University, Palo Alto, CA, in 2001, concurrently with completing the Masters degree in engineering management. Her Ph.D. topic was dynamic management of power consumption.

Prior to pursuing the Ph.D. degree, she was a Senior Design Engineer with Altera Corporation, San Jose, CA. She is currently an Associate Professor with the Computer Science Department, University of California at San Diego, San Diego. Prior to this, she was a full-time Researcher with HP Laboratories, Palo Alto, CA, while working part-time at Stanford University. With Stanford University, she has been involved with leading research of a number of graduate students and has taught graduate level classes. Her current research interests include energy efficient computing, and embedded and wireless systems.

Dr. Rosing has served at a number of technical paper committees, and is currently an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING.

Thomas Brunschwiler (M'10) received the Bachelors and Masters degrees from the NTB Buchs University of Applied Sciences and Technology, Buchs, Switzerland, in microsystem technology. He is currently completing the Ph.D. degree in electrical engineering from the Technical University of Berlin, Berlin, Germany.

Since 2001, he has been with the IBM Zurich Research Laboratory, Zurich, Switzerland, where he has developed integrated optical devices and explored organic light emitting diodes. He is currently a member of the Advanced Thermal Packaging Team, IBM Zurich Research Laboratory. His current research interests include 3-D integration with respect to volumetric heat removal and power delivery, supporting performance, and efficiency scaling of high-end servers.