Minimizing Thermal Gradient and Pumping Power in 3D IC Liquid Cooling Network Design*

Gengjie Chen , Jian Kuang , Zhiliang Zeng , Hang Zhang , Evangeline F. Y. Young , and Bei Yu

CSE Department, The Chinese University of Hong Kong

{gjchen,jkuang,zlzeng,hzhang,fyyoung,byu}@cse.cuhk.edu.hk

ABSTRACT

Liquid cooling shows great potential in resolving the huge thermal obstacle in 3D ICs. However, it brings new challenges including large thermal gradient and high pumping requirement. In this paper, liquid cooling networks with flexible topology are investigated to achieve more desirable trade-offs between energy efficiency and thermal profile. Specifically, a fast thermal model for the cooling network is proposed and analyzed, followed by our optimization methodologies to construct cooling networks targeting at pumping power saving and thermal gradient reduction, respectively. Experimental results show that, under the same constraints, the cooling network can save as much as 84.03% pumping power or reduce 37.65% thermal gradient compared to straight microchannels.

1. INTRODUCTION

With the failure of Dennard's scaling a decade ago, power becomes the number one problem in modern chip design [1, 2]. Meanwhile, with not only significant saving in delay, power and area but also possibility of yield increase and heterogeneous integration, through-silicon-via (TSV) based three-dimensional integrated circuits (3D ICs) are envisioned as one of the most promising solutions to continue the performance increase of computer systems [3]. However, 3D integration increases both heat dissipation density and thermal resistance from junction to ambient, aggravating the existing thermal problem.

To resolve the huge thermal challenge in chip design and especially in 3D ICs, microchannel-based single-phase liquid cooling has been proposed with immense potential for high-performance servers [4]. Single-phase fluid, such as water, is injected into micro-scale channels (a.k.a. microchannels) etched between two consecutive vertical tiers to carry the heat out from the 3D stack, as shown in Fig. 1(a). It is much more effective than both conventional air cooling and back-side liquid cold plate [5]. Moreover, with this aggressive cooling mechanism, some high-performance technologies limited by thermal constraints will become possible and leakage power consumption can also be reduced [6]. Prototypes of 3D ICs with microchannel-based liquid cooling system have been built by various research groups showing promising results [7–9].

However, liquid cooling brings new challenges including large thermal gradient [10] and high pumping requirement [11]. In liquid-cooled chips, coolant absorbs heat along the microchannels as it flows from inlets to outlets, making temperatures in downstream regions tend to be much higher than those in upstream regions. The deduced large thermal gradient may lead to reliability issues and timing errors. Also, due to the limited diameter of the microchannels, the energy required to inject the coolant can be a significant overhead to the whole system.

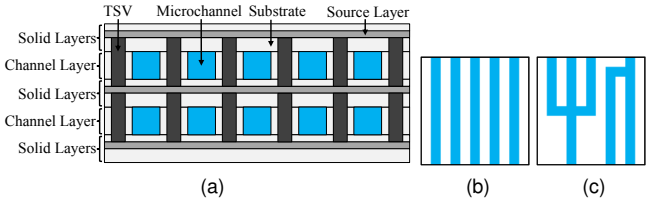

Figure 1: (a) 3D IC using microchannel-based liquid cooling. (b) Straight microchannels. (c) Cooling network with bends and branches.

Many design-time and run-time approaches are proposed to handle the challenges of liquid-cooled 3D ICs [12–15]. They primarily target at power reduction, improving thermal gradient by limited extent by chance. In [10], Sabry *et al*. use channel width modulation to optimize the cooling energy under thermal gradient and peak temperature constraints. However, the optimization is based on an one dimensional model which ignores heat transfer between regions cooled by different channels and is thus inaccurate on the full-chip scale. In addition, they all consider straight channels only and do not utilize the flexibility of CMOS process to design liquid cooling networks, as Figs. 1(b) and 1(c) show. In [16], Van Oevelen *et al*. begin to adopt the topological design in order to minimize heat transfer, but the assumption of constant temperature heat source is far from realistic chip design.

In this paper, we hence propose novel thermal modeling and design optimization methodologies for liquid cooling networks in realistic 3D ICs, in order to achieve better trade-offs among energy efficiency, thermal gradient and peak temperature. Our major contributions are as follows. **(1)** We develop a fast and accurate thermal modeling method for cooling networks. **(2)** We propose some design guidelines and also a hierarchical tree-like cooling network structure based on an extensive experimental exploration. **(3)** We develop a novel search scheme to obtain a desirable configuration for the tree-like structure, under two problem formulations which minimizes pumping power and thermal gradient respectively. The result of our method outperforms the first place in the ICCAD 2015 Contest [17].

2. THERMAL MODELING

Thermal modeling for liquid-cooled 3D ICs has recently received much attention [18–20], which however all assume unidirectional straight channels. The latest work 3D-ICE [20] is quite accurate, which has been validated by commercial computational fluid dynamics simulator and a real liquid-cooled 3D IC. The ICCAD 2015 Contest [17] extends it for flexible topology. Nevertheless, the extension only considers a 4-register model (4RM) and is slow. Therefore, we construct a fast thermal simulator for cooling network based on a 2-register model (2RM), which enables simulation in the inner loops of the design flow.

Before discussing details of our 2RM method, some preliminaries and 4RM method are briefly introduced.

2.1 Preliminaries

For a liquid *cooling system*, there are two variables: **(1)** *cooling network* $\boldsymbol{N}$, including its topology and positions of inlets and outlets; **(2)** *system pressure drop* P_{sys} across inlets and outlets.

*This work was partially supported by the Research Grants Council of Hong Kong SAR, China (Project No. CUHK14209214).

DAC '17, June 18-22, 2017, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/3061639.3062285

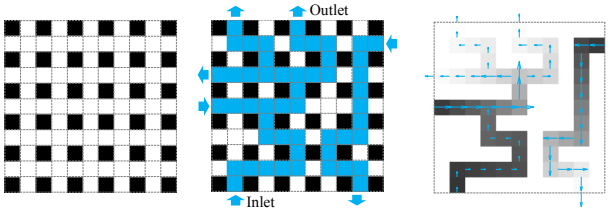

Figure 2: (a) Discretized channel layer where a basic cell either is TSV (black) or may be used for microchannels (white). (b) A cooling network. (c) Pressure and flow rate distribution where longer arrows represent larger flow rates and darker liquid cells have higher pressures.

To represent $\boldsymbol{N}$, we divide the channel layer into discrete *basic cells* with a 2D rectangular grid, assign solid/liquid properties to each basic cell, and designate the boundary liquid cells as inlets/outlets. An *inlet* or *outlet* is defined as the surface where the coolant flows into or out of the corresponding liquid cell. Besides, some basic cells are reserved for TSVs and thus not allowed to be liquid, as Figs. 2(a) and 2(b) show.

To calculate the heat transfer caused by the flowing coolant, local flow rates should be known. The following calculation is among liquid cells (with a number of n), inlets and outlets.

For fully developed laminar flow, the volumetric flow rate $Q_{i,j}$ from liquid cell i to its neighbor j is [21]:

$$Q_{i,j} = g_{fluid,i,j} \cdot (P_i - P_j), \tag{1}$$

where P_i and P_j are pressures at i and j respectively, and $g_{fluid,i,j}$ is the fluid conductance computed as $g_{fluid,i,j} = (D_h^2 A_c)/(32 l_{i,j} \mu)$. Here, $l_{i,j}$ is the distance between centers of i and j, μ is the coolant dynamic viscosity, A_c is the cross-sectional area, and D_h is the hydraulic diameter. Besides, the flow rate at an inlet/outlet of cell i is calculated similarly with a smaller fluid conductance $g_{fluid,i,edge}$.

By assuming constant water density, there is volume conservation for cell i:

$$\sum_{j \in N_i} Q_{i,j} = 0, \tag{2}$$

where N_i is the set of neighboring cells and possible neighboring inlet/outlet of i.

For convenience, the pressure at the outlet P_{out} is put as zero, so pressure value at the inlet P_{in} is P_{sys}. Then, substituting (1) into (2) creates the following system of linear equations:

$$\boldsymbol{G} \cdot \boldsymbol{P} = \boldsymbol{Q}_{in}, \tag{3}$$

where $\boldsymbol{P} \in \mathbb{R}^n$ is the vector of all liquid cell pressures, $\boldsymbol{Q}_{in} \in \mathbb{R}^n$ is the vector about flow rates at inlets, and $\boldsymbol{G} \in \mathbb{R}^{n \times n}$ is the conductance matrix. Since $\boldsymbol{G}$ and $\boldsymbol{Q}_{in}$ are known, the pressure vector $\boldsymbol{P}$ can be solved. Local flow rates are then attained by (1). An example is in Fig. 2(c).

2.2 4RM-Based Thermal Modeling

To model liquid cooling, heat transfer inside microchannels is incorporated into a lumped thermal resistance network. 4-register-model (4RM) based modeling [20] follows the microchannel geometry, where *thermal cells* are formed according to both the 2D grid defining basic cells and the stack layer division. Each thermal cell is then represented by its center as a node. There are totally three kinds of heat transfer: between solid and solid, between solid and liquid, and between liquid and liquid (shown as Fig. 3).

The thermal conductance between two neighboring solid nodes i and j is:

$$g_{ss} = \frac{q_{i,j}}{T_i - T_j} = \frac{k_{solid} \cdot A_{i,j}}{l_{i,j}}, \tag{4}$$

where $q_{i,j}$ is the heat transfer from i to j, T_i and T_j are their temperatures, k_{solid} is the thermal conductivity of the solid material, and $A_{i,j}$ is the cross-sectional area.

There are two parts for the thermal conductance g_{sl} between a solid node i and its liquid neighbor j, the conductance g_{ss}^* from i to the channel wall, and the conductance g_{sl}^* from the channel wall to j:

$$g_{sl} = \frac{q_{i,j}}{T_i - T_j} = g_{sl}^* \parallel g_{ss}^* = \frac{g_{sl}^* \cdot g_{ss}^*}{g_{sl}^* + g_{ss}^*}, \tag{5}$$

where g_{ss}^* is still calculated from (4), while g_{sl}^* is derived by $g_{sl}^* = (k_{liquid} A_{i,j} Nu)/D_h$ with k_{liquid} and Nu being the coolant thermal conductivity and Nusselt number [22].

For liquid-liquid heat transfer, the net energy received by cell i is $q_{ll} = C_v \cdot \sum_{j \in N_i} (Q_{j,i} \cdot T_{j,i}^*)$, where C_v is the volumetric specific heat of the coolant, and $T_{j,i}^*$ represents the temperature at the corresponding boundary. The boundary is either inlet/outlet or the interface between two liquid cells. The temperature at inlet $T_{in,i}^* (= T_{in})$ is constant; that at outlet $T_{out,i}^*$ can be approximated by T_i; that at the interface between two cells $T_{j,i}^* = (T_j + T_i)/2$ under the central differencing scheme. Then together with (2), there is

$$q_{ll} = \frac{C_v}{2} \cdot \sum_{j \in N_i} (Q_{j,i} \cdot T_j). \tag{6}$$

Combining energy conservation for each cell, (4), (5) and (6), a system of linear equations similar to (3) can be created. Temperatures of all thermal cells are then solved from it.

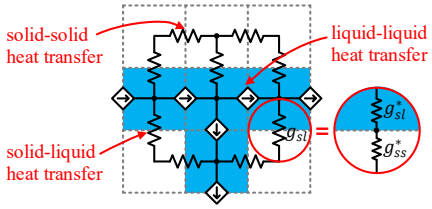

Figure 3: 4RM model with three kinds of heat transfer.

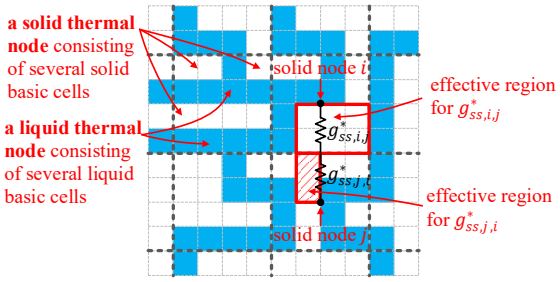

Figure 4: 2RM model with discretization of 4×4 basic cells.

2.3 Faster 2RM-Based Thermal Modeling

As Section 6 will show, 4RM simulation is quite slow. For a three-die stack, it takes as much as 16 seconds to finish a simulation. This may be acceptable for final evaluation, but is forbidding for simulation inside the design flow, where the simulator usually needs to be invoked repeatedly. Therefore, we propose a simulation method in this section, which can be tremendously faster with limited accuracy loss.

Actually, it is not difficult to understand why 4RM simulation is slow, since it requires thermal cells to conform to the microchannel geometry. Freed from this constraint, thermal cells can be larger and thus fewer, accelerating the simulation. In [9, 20], the porous-medium approach (a.k.a. 2-register-model (2RM) based modeling) has applied the idea to straight microchannels. It is also applicable to general cooling network.

In 2RM, the horizontal 2D discretization is therefore coarser than basic cells. Fig. 4 shows an example with a grid size of $m \times m$ ($m = 4$) basic cells. In the channel layer, basic cells in each grid are treated as two thermal nodes, a solid one and a liquid one, because of their diverse thermal properties and temperatures. In solid layers, a thermal node represents exactly $m \times m$ basic cells.

For the solid-solid heat transfer, the core calculation is still (4), but the corresponding geometry is no longer a simple cuboid. Take solid thermal nodes i and j in Fig. 4 as an example. If node j represents a pure solid region, its effective region for calculating the thermal conductance with node i will be the upper half of the 4×4 grid. Now among the

distributed solid basic cells, only complete conducting paths are taken into account. In this way, $g^*_{ss,j,i}$, the thermal conductance between node j and the interface, is obtained. Similarly, there is $g^*_{ss,i,j}$. The total conductance between nodes i and j is then computed as:

$$g_{ss,i,j} = g^*_{ss,i,j} \parallel g^*_{ss,j,i} = \frac{g^*_{ss,i,j} \cdot g^*_{ss,j,i}}{g^*_{ss,i,j} + g^*_{ss,j,i}}. \tag{7}$$

The above example is for a neighboring solid node pair in the same layer, but the approach is also applicable to a cross-layer pair.

For the solid-liquid heat transfer, both the vertical heat transfer (from top/bottom walls to the coolant) and the horizontal one (from side walls to the coolant) are considered only in the vertical direction [20]. That is, the thermal conductance between a liquid node and its side wall $g^*_{sl,side} = 0$. The side wall area A_{side} is added into the calculation of vertical heat transfer. The thermal conductance between a liquid node and its top/bottom wall is thus:

$$g^*_{sl,top/bottom} = h_{conv} \cdot (A_{top/bottom} + A_{side}/2), \tag{8}$$

where $A_{top/bottom}$ is its top/bottom wall area. The total solid-liquid conductance is then derived by (5).

The liquid-liquid heat transfer depends on flow rates between liquid thermal nodes. With possible multiple microchannel connections, total heat transfer is determined by the net flow rate and (6).

Similar to 4RM, we then obtain temperatures from a system of linear equations. In general, an $m \times m$ discretization reduces the problem size to $\frac{1}{m^2}$ of the 4RM one, and thus accelerates more than m^2 times (note that the exact value depends on the linear algebra (LA) solver used). Note that though only steady thermal analysis is discussed above, it can be easily extended to transient one.

3. PROBLEM FORMULATIONS

By Bernoulli's equation, *pumping power* $W_{pump} = P_{sys}Q_{sys}/\eta$ with P_{sys} being the system pressure drop and Q_{sys} being the system flow rate. There is an efficiency term η because energy loss across components such as tubes, heat exchangers and pumps. However, η depends on the parts outside the cooling network and has no impact on the optimization procedure, so it will be removed from the upcoming calculation of W_{pump}. *Thermal gradient* is defined as $\Delta T = \max_i(\Delta T_i)$ with ΔT_i being the range of node temperatures in the i-th source layer [17]. *Peak temperature* T_{max} is the maximum of the thermal node temperatures. Note that T_{max} can only occur in the source layer due to energy conservation.

Besides, the following design rules are used to make the problem concrete and realistic: **(1)** TSV positions are assumed to be at alternating basic cells in both dimensions, like Fig. 2(b); **(2)** inlets and outlets can only occur at the edges of the channel layer; **(3)** to reduce the complexity of packaging, there can be at most one "continuous" inlet and outlet on each side. In fact, without the last rule, straight channels with alternating directions can compensate temperature rise from inlets to outlets of each other very well. However, it is unpractical for packaging.

Based on the above design rules, we present two problem formulations for trade-off among W_{pump}, ΔT and T_{max}. The first problem formulation is from ICCAD 2015 Contest [17], where W_{pump} should be minimized:

Problem 1 (Pumping Power Minimization). *Given the heat dissipation of a 3D IC and some design rules, decide the cooling network and the system pressure drop of the cooling system, such that the pumping power is minimized, while the constraints on peak temperature and thermal gradient are satisfied.*

Liquid cooling causes large ΔT, bringing reliability issues and timing errors. Therefore, a second formulation treating it as the objective will also be discussed:

Problem 2 (Thermal Gradient Minimization). *Given the heat dissipation of a 3D IC and some design rules, decide the cooling network and the system pressure drop of the cooling system, such that the thermal gradient is minimized, while the constraints on peak temperature and pumping power are satisfied.*

Algorithm 1 Optimization Flow for Pumping Power Minimization

Input: $\boldsymbol{N}_{init}$, ΔT^*, T^*_{max}, stack description and floorplan files.
Output: $\boldsymbol{N}$, P_{sys}.
1: $\boldsymbol{N} \leftarrow \boldsymbol{N}_{init}$
2: **while** # iteration is within the limit **do**
3: Obtain neighboring network solution $\boldsymbol{N}'$;
4: Get the score W'_{pump} of $\boldsymbol{N}'$; ▷ Algorithm 2
5: $\boldsymbol{N} \leftarrow \boldsymbol{N}'$ or not according to SA mechanism;
6: **if** W'_{pump} converges **then** return $\boldsymbol{N}$ and P_{sys};

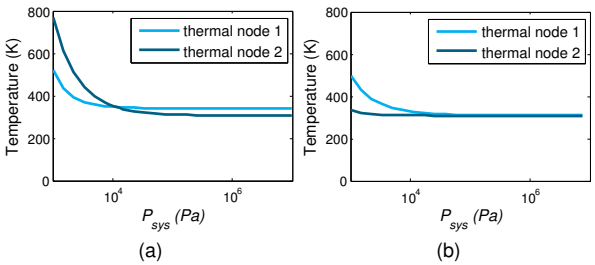

Figure 5: Relation between temperatures and P_{sys} in a network.

Among the three targets (i.e., W_{pump}, ΔT and T_{max}), we should take more care of ΔT. It is due to the following two reasons. First, for a specific cooling network $\boldsymbol{N}$, increasing W_{pump} will lower T_{max}, and vice versa, which is a simple trade-off. However, increasing W_{pump} will not necessarily lower ΔT. Second, with liquid cooling, T_{max} is decreased already, while ΔT is higher.

For ΔT, there are three inducing factors: **(1)** with a practical flow rate, temperature rise of the coolant will create uneven heat-sinking from inlet to outlet; **(2)** the power source distribution in active layers is probably non-uniform, making temperatures in different regions tend to differ; **(3)** for non-uniform channel distribution, some regions have less contact area with the coolant or are even far from the channel, which also creates uneven heat-sinking. Among them, the first two are unavoidable, but factor 3 can be used to compensate for them. For example, in a region with higher power source (factor 2), more channels can be assigned to achieve stronger heat-sinking (factor 3).

4. MINIMIZING PUMPING POWER

The ideas and technical details for solving pumping power minimization are introduced in this section. The extension to thermal gradient minimization will be explained in the next section.

First of all, the mathematical formulation for Problem 1 can be written as follows:

$$\begin{aligned} \min \quad & W_{pump}, \\ \text{s.t.} \quad & P_{sys} \in \mathbb{R}^+,\ \boldsymbol{N} \in \mathcal{N},\ T_{max} \le T^*_{max},\ \Delta T \le \Delta T^*, \end{aligned} \tag{9}$$

where $\mathcal{N}$ is the set of all legal cooling networks, ΔT^* and T^*_{max} are the corresponding constraints.

The overall two-level optimization framework for (9) is shown in Algorithm 1. In the inner level, P_{sys} is varied to minimize W_{pump} for a specific $\boldsymbol{N}$, which evaluates $\boldsymbol{N}$ by its *lowest feasible pumping power* W'_{pump} (line 4). In the outer level, simulated annealing (SA) searches for a good $\boldsymbol{N}$ solution according to W'_{pump}.

Before introducing the inner level (Section 4.2) and the outer level (Section 4.4), the relationship between P_{sys} and thermal profile is introduce first.

4.1 Relationship Between Pressure and Temperature

In general, as P_{sys} increases, temperatures of all thermal cells will decrease. When P_{sys} becomes sufficiently large, coolant temperature will be very close to T_{in} and approximately constant, making the temperatures of its neighboring solid cells also almost constant. Prior to this sufficiently large P_{sys}, temperature decrease is gradually smaller. We call it a *turning point*, as Figs. 5(a) and 5(b) show. For different cells, turning points are different (e.g., upstream regions reach turning points earlier).

Algorithm 2 Network Evaluation for Pumping Power Minimization

Input: $\boldsymbol{N}$, ΔT^*, T^*_{max}.
Output: W'_{pump}.
1: Solve (11); ▷ Algorithm 3
2: **if** $\Delta T > \Delta T^*$ **then** return $+\infty$;
3: **else if** $T_{max} > T^*_{max}$ **then**
4: Minimize P_{sys}, s.t. $T_{max} \leq T^*_{max}$;
5: **if** $\Delta T > \Delta T^*$ or $T_{max} > T^*_{max}$ **then** return $+\infty$;
6: **return** W'_{pump} corresponding to P_{sys} ;

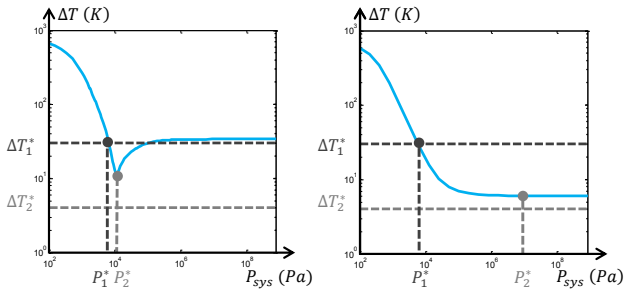

Figure 6: Cooling network with (a) uni-modal f and (b) monotonically decreasing f.

Suppose $T_{max} = h(P_{sys})$ and $\Delta T = f(P_{sys})$. Since T_{max} is the maximum among node temperatures, h also decreases monotonically and finally becomes approximately constant, as P_{sys} increases. For ΔT, if thermal cells with later turning points become cooler than those with earlier turning points (e.g., Fig. 5(a)), ΔT will begin rising at certain P_{sys}, as Fig. 6(a) shows. Otherwise (e.g., Fig. 5(b)), ΔT will keep dropping, as Fig. 6(b) shows. In short, f is either uni-modal (with minimum) or monotonically decreasing.

4.2 Network Evaluation

A network $\boldsymbol{N}$ is evaluated by a lowest feasible pumping power W'_{pump}. For a specific $\boldsymbol{N}$, the relationship between W_{pump} and P_{sys} is monotonic:

$$W_{pump} = P_{sys} \cdot Q_{sys} = P^2_{sys}/R_{sys}, \tag{10}$$

where, R_{sys} is the system fluid resistance determined by $\boldsymbol{N}$. In this way, optimizing W_{pump} is equivalent to optimizing P_{sys}.

Based on the knowledge in Section 4.1, P_{sys} is minimized for a specific $\boldsymbol{N}$ in two steps (Algorithm 2). In the first step, the problem without constraint ΔT^* on ΔT is solved. By replacing W_{pump} with P_{sys}, ignoring T^*_{max} temporarily and substituting $\Delta T = f(P_{sys})$ in (9), the mathematical formulation is single-variable:

$$\begin{aligned} \min \quad & P_{sys}, \\ \text{s.t.} \quad & P_{sys} \in \mathbb{R}^+,\ f(P_{sys}) \leq \Delta T^*. \end{aligned} \tag{11}$$

Solving (11) is still difficult because: **(1)** f comes from numerical simulation so analytical method is not suitable; **(2)** f may not be monotonic; **(3)** probing f once is time-consuming. Thus, Algorithm 3 is carefully designed to achieve accuracy and speed, based on the analysis about f in Section 4.1. If a feasible P_{sys} exists (e.g., $\Delta T^* = \Delta T^*_1$ in Fig. 6), it returns the optimal P_{sys} (i.e., P^*_1); otherwise (e.g., $\Delta T^* = \Delta T^*_2$), it returns the P_{sys} for minimum f (i.e., P^*_2), which in fact shows the nonexistence of a feasible P_{sys}.

The general idea of Algorithm 3 is moving three probing points of P_{sys} to search for the smaller P_{sys} for $f(P_{sys}) = \Delta T^*$ (line 13) or minimum f (line 8 and line 11). The initialization step (lines 1–4) makes sure that $f(P_0) > \Delta T^*$ and $f(P_0) > f(P_1)$, where P_{init} is the initial pressure and r_{init} is the initial step ratio.

In the second step (Algorithm 2 line 4), if T^*_{max} is violated, another binary search is applied directly due to the monotonic h. Note that Algorithm 2 is optimal according to the properties of h and f stated in Section 4.1. The proof is easy and omitted.

4.3 Hierarchical Tree-like Cooling Network

Algorithm 3 Algorithm Solving (11)

Input: $\boldsymbol{N}$, ΔT^*.
Output: P_{sys}.
1: $P_0 \leftarrow P_{init}$;
2: **while** $f(P_0) < \Delta T^*$ **do** $P_0 \leftarrow P_0/2$;
3: $S \leftarrow P_0 \cdot r_{init}$, $P_1 \leftarrow P_0 + S$;
4: **if** $f(P_0) < f(P_1)$ **then** $P_0 \leftarrow P_0/2$ and **go to** 2;
5: **while** $f(P_1) > \Delta T^*$ **do**
6: $S \leftarrow 2 \cdot S$, $P_2 \leftarrow P_1 + S$;
7: **while** $f(P_1) < f(P_2)$ **do**
8: **if** $|1 - \frac{P_0}{P_1}|$ and $|1 - \frac{P_2}{P_1}|$ are small enough **then return** P_1;
9: $P_2 \leftarrow P_1$, $P_1 \leftarrow (P_0 + P_2)/2$, $S \leftarrow P_2 - P_1$;
10: $P_0 \leftarrow P_1$, $P_1 \leftarrow P_2$;
11: **if** keep moving right with small $|1 - \frac{f(P_0)}{f(P_1)}|$ **then return** P_1;
12: Use binary search to find $P_{sys} \in [P_0, P_1]$ so that $f(P_{sys}) = \Delta T^*$;
13: **return** P_{sys};

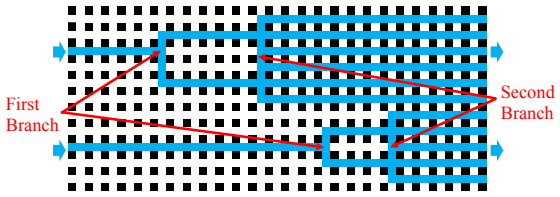

Figure 7: A tree-like cooling network on 23×51 basic cells.

In our early exploration, many cooling networks were designed manually with various styles. Among our observations, the most important one is that the thermal coupling between different regions in a chip is strong. For example, if the upstream region of a channel becomes slightly hotter, the temperature in the downstream region will be increased immediately. Therefore, global consideration is more significant than the subtle design in a local region.

Among the general structures attempted, the hierarchical tree-like structure is found to be simple (with a controllable number of parameters) and good (with respect to improving W_{pump} and ΔT under constraints). It includes several "trees" in which the coolant flows from roots to leafs, as Fig. 7 shows. This structure also conforms to the general considerations in Section 3 and can: **(1)** make cooling in upstream and downstream regions more even by having different surface areas of the microchannel walls (i.e., compensate for factor 1); **(2)** make cooling of different trees more even by differing fluid resistance and thus flow rates (i.e., compensate for factor 2).

4.4 Network Topology Optimization

In the proposed tree-like network structure, each "tree" has two parameters to be configured, the positions for the first and the second branches. We design an SA-based algorithm to search for a good configuration of those parameters.

Before searching, $\boldsymbol{N}$ is initialized with uniform tree parameters (i.e., same position for all first branches and same position for all second branches). There are totally four stages, each of which corresponds to a complete SA process described in Algorithm 1. **(1)** In each iteration, every tree parameter may be changed by a large step size or remains unchanged (with equal possibility). The acceptance of neighboring solutions are determined by SA. ΔT under a fixed P_{sys} is the cost function at this stage, which only needs simulating once. Note that a single simulation can also generate a result on W_{pump}, but under a fixed P_{sys}, W_{pump} reveals nothing about the die power (purely determined by $\boldsymbol{N}$) and thus is not eligible for the cost function. Besides, 2RM simulator is used for quick searching. **(2)** Stage 2 adopts the same move and simulator as stage 1 except that the neighboring solution is evaluated by the lowest feasible pumping power W'_{pump}. If there is no feasible P_{sys}, W'_{pump} is $+\infty$. This evaluating scheme needs invoking the simulator several times and takes longer runtime. **(3)** It is similar to stage 2, but a smaller step size is used. **(4)** It is similar to stage 3 except that the more accurate 4RM simulator is applied.

Settings for the four stages are summarized in Table 1. In general,

Table 1: Four-stage Optimization for Pumping Power Minimization

Stage	Step Size	Objective	Simulator	Runtime for an Iteration
1	10	ΔT	2RM	short
2	10	W'_{pump}	2RM	medium
3	2	W'_{pump}	2RM	medium
4	2	W'_{pump}	4RM	long

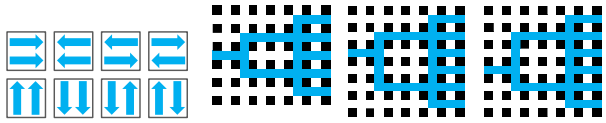

Figure 8: (a) Eight global flow directions. (b) Three types of branches.

earlier stages are rougher and much quicker. Therefore, with small runtime overhead, more rounds can be afforded to fully explore the solution space. In different rounds of a stage, all settings are the same except the random seed. After finishing a stage, the best solution in each round is re-evaluated by the metric in the next stage (if the metric is different). The re-evaluated best solution among all rounds is then selected as the output of the stage.

For the global flow directions (see Fig. 8(a)), all configurations are attempted and the best is chosen. Besides, there are three types of suitable branches (see Fig. 8(b)). They are assigned manually to fit the chip size.

5. MINIMIZING THERMAL GRADIENT

Our method for minimizing thermal gradient (Problem 2) is still under the flow in Algorithm 1, but adaption is necessary for validity and quality.

First, its mathematical formulation in general becomes:

$$\begin{aligned} \min \quad & \Delta T, \\ \text{s.t.} \quad & P_{sys} \in \mathbb{R}^+,\ \boldsymbol{N} \in \mathcal{N}, \\ & T_{max} \le T^*_{max},\ W_{pump} \le W^*_{pump}, \end{aligned} \tag{12}$$

where W^*_{pump} is the constraint on W_{pump}.

Second, compared to the network evaluation process in Section 4.2, two modifications are needed. **(1)** Its simplified form corresponding to (11) becomes:

$$\begin{aligned} \min \quad & f(P_{sys}), \\ \text{s.t.} \quad & P_{sys} \in \mathbb{R}^+,\ P_{sys} \le P^*_{sys}, \end{aligned} \tag{13}$$

where P^*_{sys} is the constraint on P_{sys} computed by (10) and W^*_{pump}. **(2)** Solving (13) is simpler. If P^*_{sys} locates on the falling side of f, it is the optimal solution directly; otherwise, golden section search is adopted to find the minimum f.

Third, the network optimization process is similar to that in Section 4.4 with the following changes. **(1)** Objective function is changed from W_{pump} to ΔT. **(2)** Several consecutive iterations are grouped together, the first of which is evaluated normally by the complete network evaluation process. The other iterations are then evaluated through one simulation under a fixed P_{sys} (the optimal P_{sys} obtained in the first iteration). In this way, runtime is reduced significantly. The later evaluations may be pessimistic but inaccuracy is small because the optimal P_{sys} of neighboring $\boldsymbol{N}$ is close. **(3)** The original stage 1 is no longer needed due to the speed-up obtained from above technique 2. **(4)** The speed-up also makes 4RM affordable in the original stage 3.

6. EXPERIMENTAL RESULTS

Our thermal modeling and design optimization methods were implemented in C++ and with LA library Eigen [23]. Experiments were performed on an 80-core 2.20 GHz Linux server and with ICCAD 2015 Contest benchmarks [17]. In the benchmarks, the die is as large as $10.1mm \times 10.1mm$ and divided into 101×101 basic cells. Channel width $w_c = 100\mu m$ and inlet temperature $T_{in} = 300K$. More details are listed in Table 2, where h_c is the channel height, T^*_{max} and ΔT^* are constraints on T_{max} and ΔT.

Table 2: ICCAD 2015 Benchmark Statistics

#	Die Num	h_c (μm)	Die Power (W)	ΔT^* (K)	T^*_{max} (K)	Other Constraint
1	2	200	42.038	15	358.15	-
2	2	400	37.038	10	358.15	-
3	2	400	43.038	15	358.15	no channel in a restricted area
4	3	200	43.438	10	358.15	matched inlets/outlets across layers
5	2	400	148.174	10	338.15	-

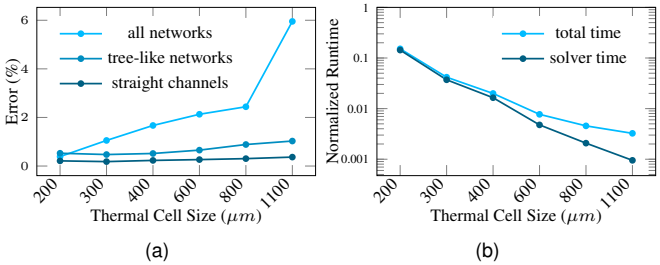

Figure 9: (a) Accuracy of 2RM compared to 4RM. (b) Speed-up of 2RM relative to 4RM.

The 2RM method reduces the problem size and thus accelerates simulation significantly, which however may lose some accuracy. To examine whether the accuracy loss is limited, an experiment with 5 benchmarks, 40 network samples, 6 thermal cell sizes and 13 pressures is conducted. The network samples cover straight-channel networks, the proposed tree-like networks, and many styles of manual designs generated during our early exploration.

Amoung the $5 \times 40 \times 6 \times 13 = 15600$ 2RM simulations, the error of each is evaluated by its average relative error of thermal nodes in the source layers (compared with 4RM simulation). Errors of all networks with different benchmarks, different pressures and the same thermal cell size are then averaged. The same computation is also conducted for all tree-like networks and all straight-channel networks. The result in Fig. 9(a) shows that accuracy decreases as the thermal cell size increases. Error is also affected by the network structure with straight-channel networks having the smallest. Besides, accuracy is also related to the benchmark and the pressure (details are omitted due to space limitation). Nevertheless, errors of 2RM simulation with small thermal cell sizes are all very small.

The runtime of 2RM depends on the thermal cell size. Fig. 9(b) shows the runtime speed-up over 4RM model. Here, a 4RM simulation takes 3.37 s for test cases 1, 2, 3 and 5 (with two dies), and 15.62s for case 4 (with three dies).

In general, when the thermal cell size is small, the runtime saving by enlarging thermal cells is significant while accuracy loss is small. For example, simulating with tree-like networks and $400\mu m$ thermal cells results in only 0.517% errors compared with 4RM, but the runtime is reduced from 3.37s to 0.07s. However, when thermal cell becomes very large, little runtime is consumed by the LA solver and the overhead dominates. The speed-up is thus increasingly less, while the accuracy still keeps worsening.

As a good trade-off between accuracy and runtime, $400\mu m$ thermal cell is adopted for the 2RM simulations in solving pumping power minimization (Problem 1) and thermal gradient minimization (Problem 2). With the multi-core computer, 64 neighboring $\boldsymbol{N}$ solutions are evaluated simultaneously in each iteration.

As mentioned in Section 1, nearly all previous works about liquid-cooled 3D-ICs assume straight microchannels. We thus use regular straight-channel networks as baselines. For each test case, straight channels of diverse global directions are evaluated by the network evaluation process and the best is the baseline. For Problem 1, there is no feasible baseline solution on case 5, due to the high and highly varied die power, and tight T^*_{max}. In case 3, there is a region forbidding microchannels. To satisfy the requirement, that region is filled by solid cells and surrounded by liquid cells, in both baseline networks and our tree-like network designs.

For solving Problem 1, the four stages consist of 60, 40, 40 and 30 iter-

Table 3: Result for Pumping Power Minimization (Problem 1)

Case #		1	2	3	4	5
Baseline (straight channels)	$P_{sys}\ (kPa)$	12.98	6.23	7.85	9.71	N/A
	$T_{max}\ (K)$	322	314	321	314	N/A
	$\Delta T\ (K)$	15.0	10.0	15.0	10.0	N/A
	$\boldsymbol{W_{pump}\ (mW)}$	**10.41**	**6.91**	**8.34**	**11.65**	**N/A**
Manual (1st place in ICCAD Contest)	$P_{sys}\ (kPa)$	8.86	5.54	6.98	9.45	40.1
	$T_{max}\ (K)$	357	336	328	336	338
	$\Delta T\ (K)$	15.0	10.0	15.0	10.0	10.0
	$\boldsymbol{W_{pump}\ (mW)}$	**1.72**	**1.51**	**3.36**	**2.96**	**113.96**
Ours	$P_{sys}\ (kPa)$	8.72	5.13	5.81	8.27	40.10
	$P_{system}\ (kPa)$	358	336	337	335	338
	$\Delta T\ (K)$	15.00	10.0	15.0	10.00	10.00
	$\boldsymbol{W_{pump}\ (mW)}$	**1.66**	**1.37**	**1.90**	**2.68**	**113.96**

Table 4: Result for Thermal Gradient Minimization (Problem 2)

Case #		1	2	3	4	5
Baseline (straight channels)	$P_{sys}\ (kPa)$	26.08	14.43	17.82	26.51	45.81
	$T_{max}\ (K)$	316	309	316	308	338
	$W_{pump}\ (mW)$	42.0	37.0	43.0	43.4	148.2
	$\boldsymbol{\Delta T\ (K)}$	**8.75**	**5.42**	**11.42**	**4.76**	**26.48**
Ours	$P_{sys}\ (kPa)$	16.51	8.96	11.46	13.80	40.06
	$T_{max}\ (K)$	338	319	327	321	338
	$W_{pump}\ (mW)$	5.67	5.66	6.56	4.16	113.80
	$\boldsymbol{\Delta T\ (K)}$	**5.54**	**3.81**	**7.12**	**3.87**	**9.64**

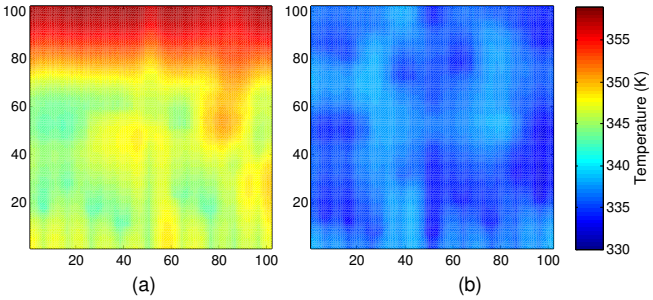

Figure 10: Temperature results on bottom source layer of case 1 for (a) pumping power minimization (Problem 1), and thermal gradient minimization (Problem 2).

ations, and 8, 4, 2 and 1 round(s), respectively. The whole SA searching takes about 40 min for cases 1-3 and about 240 min for case 4. In the difficult case 5, SA cannot find a feasible solution with tree-like structure, so the cooling system is designed manually.

Because Problem 1 is exactly the formulation in ICCAD 2015 Contest, the contest benchmarks are used directly. The result is shown in Table 3. Compared with the baseline, our method achieves up to 84.03% improvement on W_{pump}. The W_{pump} of our approach also outperforms the first place in the ICCAD 2015 Contest[1] by 16.35% on average. To the best of our knowledge, the network designs of the first place rely heavily on manual search, while our results are generated by automatic SA searching except for case 5.

For solving Problem 2, the three stages consist of 80, 20 and 20 iterations, and 8, 2 and 1 round(s), respectively. The whole searching takes about 180 min for case 4, and 30 min for the others.

To evaluate our algorithm for Problem 2, all settings in the ICCAD 2015 benchmark are kept except that ΔT^* is replaced by the constraint W^*_{pump} on W_{pump}. Table 4 shows the result when W^*_{pump} is set to 0.1% of the die power. For cases 1-4, our SA-based approach achieves as much as 37.65% improvement on ΔT compared to the baseline. Due to the difficulty of case 5, manual design is used, where the cooling network with flexible topology is still much better than the straight-channel network.

Fig. 10 shows the resulted temperature maps of the bottom source layer for case 1. Here, the map of Problem 1 is hotter in general and implies smaller W_{pump}, but its ΔT is more significant. In the contrary, result of solving Problem 2 has much smaller ΔT with larger W_{pump}. In practice, the problem formulation can be chosen according to preference between W_{pump} and ΔT.

[1] Only the result of the first place is listed because the final score of the first is 29× and 2596× better than the second and third respectively, as reported by the contest organizer.

7. CONCLUSION AND FUTURE WORK

In this work, we investigate liquid cooling networks for better trade-offs between energy efficiency and thermal profile. Specifically, we first develop a fast and accurate 4RM-based thermal modeling method for liquid cooling networks. Design optimization methodologies which minimize pumping power and thermal gradient respectively are then proposed. In experiments, the cooling network achieves as much as 84.03% pumping power saving or 37.65% thermal gradient reduction compared to straight channels, showing the great potential of cooling networks in solving the challenges of liquid cooling.

Future work includes combining cooling networks with run-time thermal management techniques (e.g., DVFS and adjustable flow rates) to handle dynamic die power. Moreover, since channel layers are shared by TSVs and microchannels, another line of research is co-optimization between them for a better global benefit.

8. REFERENCES

[1] M. Horowitz *et al.*, "Scaling, power, and the future of CMOS," in *Proc. IEDM*, 2005, pp. 7–15.

[2] H. Esmaeilzadeh *et al.*, "Dark silicon and the end of multicore scaling," in *Proc. ISCA*, 2011, pp. 365–376.

[3] S. Borkar, "3D integration for energy efficient system design," in *Proc. DAC*, 2011, pp. 214–219.

[4] C. Serafy *et al.*, "Unlocking the true potential of 3-D CPUs with microfluidic cooling," *IEEE TVLSI*, vol. 24, no. 4, pp. 1515–1523, 2016.

[5] T. Brunschwiler *et al.*, "Forced convective interlayer cooling in vertically integrated packages," in *Proc. ITherm*, 2008, pp. 1114–1125.

[6] C. Serafy *et al.*, "High performance 3D stacked DRAM processor architectures with micro-fluidic cooling," in *Proc. 3DIC*, 2013, pp. 1–8.

[7] B. Dang *et al.*, "Integrated microfluidic cooling and interconnects for 2D and 3D chips," *IEEE TAP*, vol. 33, no. 1, pp. 79–87, 2010.

[8] C. R. King Jr *et al.*, "Electrical and fluidic C4 interconnections for inter-layer liquid cooling of 3D ICs," in *Proc. ECTC*, 2010, pp. 1674–1681.

[9] T. Brunschwiler *et al.*, "Heat-removal performance scaling of interlayer cooled chip stacks," in *Proc. ITherm*, 2010, pp. 1–12.

[10] M. M. Sabry *et al.*, "GreenCool: An energy-efficient liquid cooling design technique for 3-D mpsocs via channel width modulation," *IEEE TCAD*, vol. 32, no. 4, pp. 524–537, 2013.

[11] S. Garimella *et al.*, "On-chip thermal management with microchannel heat sinks and integrated micropumps," *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1534–1548, 2006.

[12] H. Qian *et al.*, "An efficient channel clustering and flow rate allocation algorithm for non-uniform microfluidic cooling of 3D integrated circuits," *Integration, the VLSI Journal*, vol. 46, no. 1, pp. 57–68, 2013.

[13] B. Shi *et al.*, "Hybrid 3D-IC cooling system using micro-fluidic cooling and thermal TSVs," in *Proc. ISVLSI*, 2012, pp. 33–38.

[14] ——, "Optimized micro-channel design for stacked 3-D-ICs," *IEEE TCAD*, vol. 33, no. 1, pp. 90–100, 2014.

[15] M. M. Sabry *et al.*, "Energy-efficient multiobjective thermal control for liquid-cooled 3-D stacked architectures," *IEEE TCAD*, vol. 30, no. 12, pp. 1883–1896, 2011.

[16] T. Van Oevelen *et al.*, "Numerical topology optimization of heat sinks," in *International Heat Transfer Conference*, 2014, pp. 10–15.

[17] A. Sridhar *et al.*, "ICCAD 2015 contest in 3D interlayer cooling optimized network," in *Proc. ICCAD*, 2015, pp. 912–915.

[18] Y. J. Kim *et al.*, "Thermal characterization of interlayer microfluidic cooling of three-dimensional integrated circuits with nonuniform heat flux," *Journal of Heat Transfer*, vol. 132, no. 4, p. 041009, 2010.

[19] H. Mizunuma *et al.*, "Thermal modeling and analysis for 3-D ICs with integrated microchannel cooling," *IEEE TCAD*, vol. 30, no. 9, pp. 1293–1306, 2011.

[20] A. Sridhar *et al.*, "3D-ICE: A compact thermal model for early-stage design of liquid-cooled ICs," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2576–2589, 2014.

[21] T. L. Bergman *et al.*, *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2011.

[22] R. K. Shah *et al.*, *Laminar Flow Forced Convection in Ducts*. Academic Press, 1978.

[23] "Eigen," http://www.eigen.tuxfamily.org/.