Explicit Transient Thermal Simulation of Liquid-Cooled 3D ICs

Alain Fourmigue, Giovanni Beltrame, Gabriela Nicolescu
Ecole Polytechnique Montreal, Canada
alain.fourmigue@polymtl.ca, giovanni.beltrame@polymtl.ca, gabriela.nicolescu@polymtl.ca

***Abstract*—The high heat flux and compact structure of three-dimensional circuits (3D ICs) make conventional air-cooled devices more subsceptible to overheating. Liquid cooling is an alternative that can improve heat dissipation, and reduce thermal issues. Fast and accurate thermal models are needed to appropriately dimension the cooling system at design time. Several models have been proposed to study different designs, but generally with low simulation performance. In this paper, we present an efficient model of the transient thermal behaviour of liquid-cooled 3D ICs. In our experiments, our approach is 60 times faster and uses 600 times less memory than state-of-the-art models, while maintaining the same level of accuracy.**

***Index Terms*—3D ICs, Liquid-cooling, Compact Thermal Model, Finite Difference Method**

I. Introduction

Three-dimensional integrated circuits (3D ICs) are a class of devices gaining considerable interest from all the major players in the market. One of the main challenges for the design of effective 3D ICs is temperature control [9]. Their compact structure can lead to high temperatures, with uneven distributions, effectively reducing the performance and expected lifetime of these devices.

As air-cooling struggles to dissipate the high heat flux generated by 3D ICs, microchannel liquid cooling is shaping up as an effective alternative [8]. Integrated microchannels etched in the silicon substrate can be used to circulate fluid, effectively providing liquid cooling. Being present only in the substrate, microchannels do not reduce transistor real estate and can collect a high heat flux. However, microchannels need to be dimensioned according to the expected heat flux, liquid flow rate and the range of acceptable temperatures. Determining these parameters requires fast, accurate, and memory efficient thermal models.

To avoid impacting on the design schedule, a fast simulation is of paramount importance to allow the designer to evaluate multiple alternatives while the layout and power output of the device are still subject to change.

Most available thermal models focus on steady-state behaviour, and are unable to consider the continuously changing power output of modern systems, that is introduced by task migration policies and DVFS techniques. Determining the transient thermal behaviour of a circuit requires evaluating the temperature at small time intervals. Current models deal with the complexity of transient temperature evaluation with high computational load, memory use, and long simulation times [9], [12], [13], or low accuracy [4].

978-3-9815370-0-0/DATE13/©2013 EDAA

In this work, we introduce a highly accurate model (error less than $0.1°C$) which is 60x faster and uses 600x less memory than the current state-of-the-art for a typical 3D IC layout. It is worth noting that, to the best of our knowledge, our model also scales better than any approach found in literature, meaning that the speedup factor would be higher for more complex devices.

This paper's contributions can be summarized as:

- a model which directly computes the transient temperature using an explicit formulation of the finite difference method, with linear complexity that does not require any matrix inversion
- a model with low memory requirements, that allows to model realisticly sized systems (e.g. 2x2cm)
- a highly parallel and scalable implementation

Though beyond the scope of this paper, the proposed methodology can be extended to model more complex phenomena such as two-phase cooling, where the heat transfer coefficients change dynamically (for which matrix inversions are highly impractical).

The remainder of this paper is organized as follows: Section II gives a primer on thermal modeling of liquid-cooled 3D ICs; Section III discusses related work; Section IV introduces our approach; Section V details the experimental results; Finally, Section VI draws some conclusions and discusses future work directions.

II. Background on thermal modeling of liquid-cooled 3D ICs

A typical 3D IC with microchannel cooling is made of several stacked dies with the microchannels etched into the silicon bulk of one or more dies, as shown in Figure 1. When considering this structure, three types of heat transfer can take place: conduction, convection, and advection.

Conduction: In all the solid state parts of the circuit, heat transfer takes place by conduction [7]. Conductive heat flux q_{cd} in a homogeneous medium with thermal conductivity k, can be described by Fourier's law [14].

$$\overrightarrow{q_{cd}} = -k\overrightarrow{\nabla T} \tag{1}$$

Convection: At the wall of the microchannels, the fluid velocity is rendered null by viscosity, and heat is transferred

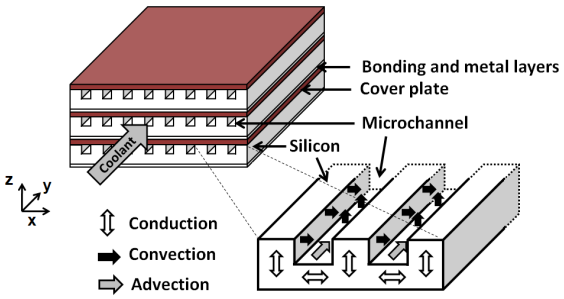


Fig. 1: Heat transfers in liquid-cooled 3D ICs

between the wall and the liquid flowing inside the microchannel. This mechanism is called forced convection, and can be described by Newton's law [7].

$$\overrightarrow{q_{cv}} = h(T_{wall} - T_{liquid})\overrightarrow{n}_{wall} \tag{2}$$

The heat transfer coefficient h depends on the particular liquid considered, and on the dynamics of the flow, which are influenced by the geometry of the microchannel.

Advection: Advection takes place inside the microchannels, due to the energy transported by the fluid's bulk motion. Advective heat flux q_{ad} through a section S of a microchannel can be modeled using:

$$\overrightarrow{q_{ad}} = \int_S c_v T \overrightarrow{u}\,\overrightarrow{ds} \tag{3}$$

where c_v is the volumetric heat capacity of the fluid, u its velocity and T its temperature at the considered section.

Heat equation for liquid-cooled 3D ICs: The heat equation governs the evolution of the temperature T in time (t) and is obtained by writing an energy balance on a volume V:

$$\int_V c_v \frac{\partial T}{\partial t} dv = \int_S \overrightarrow{q}.\overrightarrow{ds} + \int_V P_{vol} dv \tag{4}$$

where c_v is the volumetric heat capacity of the material. The first term on the right is the heat exchanged through V's surface, while the second term represents the power P_{vol} generated inside the volume V.

Solving the heat equation allows to determine the temperature everywhere in the target volume. To obtain a unique solution, the boundary conditions and the initial temperature distribution have to be specified. Solving a partial differential equation such as Equation (4) is a complex task: there are no analytical methods applicable to the heterogeneous structure of 3D ICs and the various types of heat transfer involved. As a result, numerical methods are most commonly used.

Finite Difference Methods (FDMs): Finite Difference Methods use discrete approximations to replace continuous derivatives in Equation (4) by finite differences that can be more easily computed. For this purpose, the chip is discretized into small cubic cells and time is divided into small time steps Δt. Several discretization schemes can be used to approximate the derivatives, and can be classified as explicit and implicit schemes.

Explicit schemes express the future temperature of a cell in terms of the temperature of the cell and its neighbors at present time. Therefore, the temperature can be computed directly and in lockstep with the time increments. One drawback of this technique is that it is subject to a stability criterion that constrain the choice of the time step: small cells force small time steps, and therefore many simulation iterations.

Implicit schemes formulate the space derivatives in terms of the temperatures at the next time step. Implicit formulations do not permit the direct calculation of the cells' temperature, but require instead to solve a set of equations at each step of the simulation. Implicit schemes require complex calculations at each time step, but are inherently stable: larger time steps can reduce the number of iterations at the expense of accuracy.

The trade off rests between the number of iterations needed, the computation requirement of each time step, and the desired level of accuracy. In this paper we show that for the modelling of 3D ICs, the additional iterations introduced by an explicit method are more than compensated by the reduced complexity of each time step.

III. Related work

Several methods have been proposed to build thermal models of liquid-cooled 3D ICs [8], [9], [3], [12], [4], [13]. Most of these methods are based on a finite difference formulation, which is particularly suited to the rectangular geometries of ICs. Nevertheless, most methods require solving large linear systems of equations with hundreds of thousands unknown variables, making them very expensive in terms of computational and memory requirements. Table I summarizes the characteristics of the most recent approaches for modeling liquid-cooled 3D ICs.

Several models have been proposed to perform steady-state analysis [8], [9], [3]. These models are based on preconditioned iterative methods, and can be numerically efficient, especially when implemented on GPUs [3]. However, they provide no information on the transient behavior of the circuit and can lead to inaccurate results, especially when considering circuits in which the power dissipation changes continuously.

The 3D-ICE simulation library [12] allows to compute the transient temperature using the backward Euler method. 3D-ICE was validated against a prototype and has a good level of accuracy, but its performance is greatly reduced by the high memory requirements and little degree of parallelism. The method used by 3D-ICE has polynomial complexity, resulting in intractable simulation times when considering chips of realistic dimensions.

Fourmigue *et al.* [4] perform transient analysis and use an operator splitting technique to achieve linear complexity and a high degree of parallelism. This technique is very efficient but introduces an additional splitting error on top of the normal discretization error. The splitting error is difficult to control and reduces the confidence in the results. In this paper, we

TABLE I: Recent thermal models of liquid-cooled 3D ICs

Author	Analysis	Solver	Complexity	Implementation
Kim [8]	Steady state	Successive Under Relaxation	$O(N)$	CPU
Mizunuma [9]	Steady state	Gauss-Seidel	$O(N)$	CPU
Feng [3]	Steady state	Conjugate Gradient	$O(N)$	GPGPU
Sridhar [12]	Transient	Backward Euler	$O(N^{1.5})$	CPU
Fourmigue [4]	Transient	Operator Splitting	$O(N)$	CPU
Vincenzi [13]	Transient	Neural Network	$O(N^x)^\dagger$	GPGPU

† $1 < x < 2$ depending on the chosen accuracy

present a technique with the same complexity, comparable simulation times, but better accuracy and higher confidence: our approach introduces only local truncation errors, the behavior of which is well known and bound.

Vincenzi *et al.* propose to accelerate transient analysis using neural networks [13]. They train a neural network to mimic the thermal behavior of the modeled chip using a conventional thermal simulator such as 3D-ICE. Once trained, the neural network can be used to simulate the chip using different power maps as long as the physical structure of the chip remains unmodified. The GPU implementation of [13] reports a speedup of 35-100x against 3D-ICE running on CPUs, depending on the level of accuracy. Beside the fact that GPUs are required to provide acceptable simulation times, this work suffers from another limitation: design space exploration involving any variation in microchannel dimensions, number of layers, or coolant flow rates requires new training with a conventional thermal simulator. The work proposed in this paper does not have any of these limitations, and provides higher accuracy and speed while running of a common laptop computer.

IV. Proposed Methodology

Our approach to derive a fast and accurate thermal model for liquid-cooled 3D ICs is based on the Euler FDM [2]. The Euler method approximates the continuous derivatives using an explicit scheme with first order accuracy in time and space.

As explained in Section II, the time is divided into small time steps Δt and the chip is discretized into small cubic cells, referred as thermal cells. To properly handle the different materials, we discretize the circuit such that each thermal cells contains only one material, either solid or liquid. Figure 2 shows the discretization of the chip into thermal cells and the heat flux $q_{<dir>}$ received by a thermal cell from the neighboring cells in the *north, south, east, west, top,* and *bottom* directions.

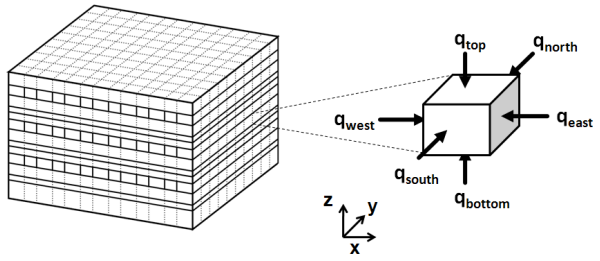


Fig. 2: Discretization of the chip into thermal cells

Using this formalism, Equation (4) is applied to each thermal cell and discretized as follows:

$$C\frac{\Delta T}{\Delta t} = q_{north} + q_{south} + q_{east} + q_{west} + q_{top} + q_{bottom} + P \quad (5)$$

where C is the thermal capacity of the cell and P the power generated in the cell. We compute conductive and convective heat fluxes between adjacent cells using a first order approximation of Equations 1 and 2:

$$q_{i\to j} \simeq \frac{T_i^n - T_j^n}{r_i + r_j} \quad (6)$$

which provides the heat flux $q_{i\to j}$ transfered by conduction from cell i to cell j at time $t_n = n\Delta t$ in terms of the cells' temperature at time t_n, and their thermal resistances r_i, r_j along the direction of the heat transfer.

The advective heat transfer inside the microchannels is:

$$q_{i\to j} \simeq c_v u S T_i^n \quad (7)$$

where c_v is the volumetric heat capacity of the coolant, u is the velocity of the coolant flow in the microchannel and S is the section of the microchannel. Equation (7) gives the heat flux transported by the coolant as it moves from cell i to the downstream neighboring cell j at time t_n.

The Euler method explicitly expresses the future temperature T^{n+1} of a cell in terms of its current temperature T^n and the temperature of the neighboring cells $T^n_{<dir>}$ at current time t_n, plus a contribution due to the power generated in the cell. The Euler method leads to:

$$T^{n+1} = T^n + \frac{\Delta t}{C}\Sigma\ g_{<dir>}(T^n_{<dir>} - T^n) + \frac{\Delta t}{C}P^n \quad (8)$$

The thermal conductance $g_{<dir>}$ with the neighboring cell in the direction $< dir >$ can be derived from Equations 6 and 7.

Using Equation (8), we directly compute the transient temperature for each cell without having to solve a whole set of equations at each time step. Due to its explicit formulation, the Euler method has linear complexity in the number of cells [2]. Since no matrix inversion is performed, the Euler method is suitable for lightweight and efficient implementations. Our approach has a low computational load per iteration, and it only needs to store the temperature values and the coefficients $\frac{\Delta t}{C} g_{<dir>}$ of Equation (8). As an additional optimization, we factor all the redundant coefficients due to the regular structure of the chip (cells with the same thermal capacity, resistance, etc.). This leads to a very low memory footprint implementation.

However, the Euler method has to satisfy a stability criterion to ensure the local truncation error introduced by the FDM approximation will not grow indefinitely as the simulation proceeds. This criterion can be derived from the Euler method in matrix form:

$$[T]^{n+1} = M[T]^n + [U]^n. \tag{9}$$

Stability is achieved if all the complex eigenvalues of matrix M are within a circle of radius 1 centered at -1 [2]. This severely constrains the time step to low values (on the order of $10\mu s$ for typical liquid-cooled 3D ICs), increasing the number of iterations needed for the simulation of a given time interval. Nonetheless, when considering modern circuits, the fast variations in power generation due to task migration [10] and DVFS schemes [5] already constrain the time step to sub-millisecond values (see Section V). The result is that the stability criterion does not excessively affect the performance, making the Euler method suited to this particular application.

To take advantage of today's muti-core machines, we propose an efficient parallel implementation of the Euler method based on domain decomposition. We partition the discretized chip in equal size blocks (as shown in Figure 3), allocating the temperature values in consecutive memory locations. The blocks can be evaluated in parallel, computing the transient temperature using Equation (8). As an example (and in no way a limitation), Figure 3 shows an 8-way partition.

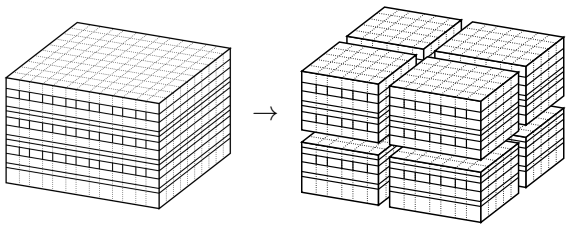


Fig. 3: Partioning the discretized chip into blocks

The parallelization is outlined by Algorithm 1: we first compute the contributions to each cell's temperature coming from neighbors inside each block. Then, the contributions between blocks is computed considering each interface sequentially.

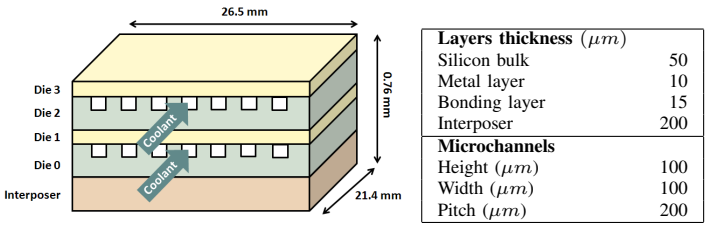
for *block* **do in parallel**
  foreach *cell in block* **do**
    Get contribution to T_{cell}^{n+1} from neighbors in *block*
  end
end
for *block* **do in parallel**
  foreach *face of block* **do**
    foreach *cell of face* **do**
      Get contribution to T_{cell}^{n+1} from neighbors located outside *block*
    end
  end
end

Algorithm 1: Proposed parallel algorithm to apply the Euler method to a block-partitioned domain

V. Experimental results

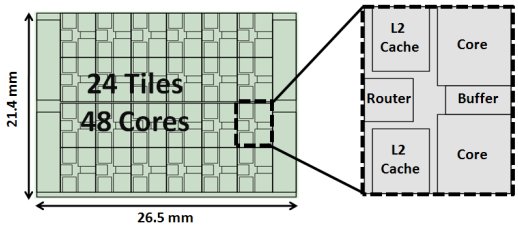
To validate the proposed model and determine its performance we compare against the state-of-the-art 3D-ICE [12], a freely distributed tool. 3D-ICE was itself validated against a prototype and a commercial simulator.

As a target, we consider the layout of a realistic liquid-cooled 3D IC shown in Figure 4, composed of four dies, that surmount a silicon interposer:

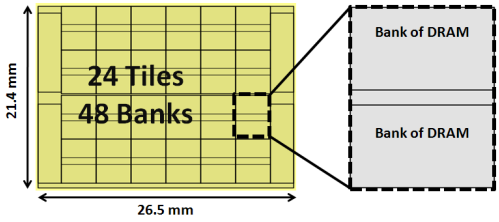


Layers thickness (μm)	
Silicon bulk	50
Metal layer	10
Bonding layer	15
Interposer	200
Microchannels	
Height (μm)	100
Width (μm)	100
Pitch (μm)	200

(a) Geometry of the target structure



(b) Floorplan of die 0, die 2



(c) Floorplan of die 1, die 3

Fig. 4: Layout of the target structure

- the floorplan for dies 0 and 2 (see Figure 4b) are based on the specifications of Intel's Single-Chip Cloud Computer (SCC) [6] a 24-tile, 48-core IA-32 45nm processor with a maximum power dissipation of 125W. Given the large size of the chip ($567mm^2$) and its high power dissipation, the chip is a good candidate for microchannel cooling, especially when stacked with additional memory layers Integrated microchannels of typical dimensions (see Figure 4a) are present at the back side of dies 0 and 2
- the floorplan for dies 1 and 3 (see Figure 4c) includes realistic memory architecture (24 tiles and 48 DRAM banks)
- Through-Silicon Vias (TSVs) were not considered as their impact on the global temperature distribution is rather limited [4]. Nevertheless, they could be easily added

We assume adiabatic boundary conditions on all the faces of the target, since liquid-cooled 3D ICs are usually enclosed

by a manifold [12]. Water flows in the microchannels with a velocity of $1.4m/sec$. The initial temperature and the temperature of the incoming coolant are taken to be $26°C$ (ambient temperature).

We simulated the transient behavior of the target for $200ms$. This time interval is sufficiently long to study the transient response of the circuit, being three times the RC time constant of the target structure (around $60ms$). The power dissipated by our target is estimated assuming random activity in all cores. The power values for each unit were extracted from the SCC's specifications. Figure 5 shows the overall power dissipation of the four dies.

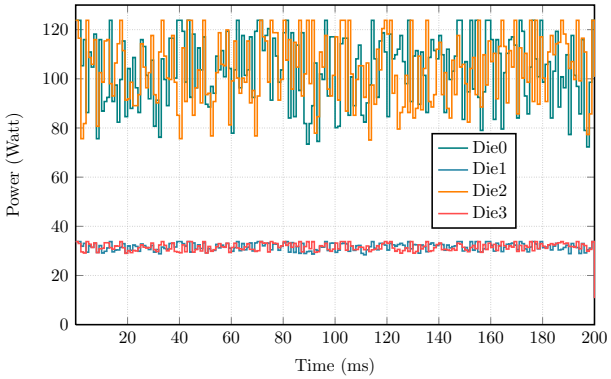


Fig. 5: Power dissipation used to simulate the target structure

SCC provides fine-grained DVFS, with ten microseconds time intervals [6], [11]. Several studies have shown the advantages of fine-grained DVFS [5] or thread migration policies at finer intervals than the conventional millisecond time scale of the OS schedulers [10]. We therefore consider power maps with a minimum interval of $1ms$ between values.

Figure 6 shows the evolution of power dissipation (used as input for the simulation) and the temperature (computed by the proposed model) for die 0 during the simulation.

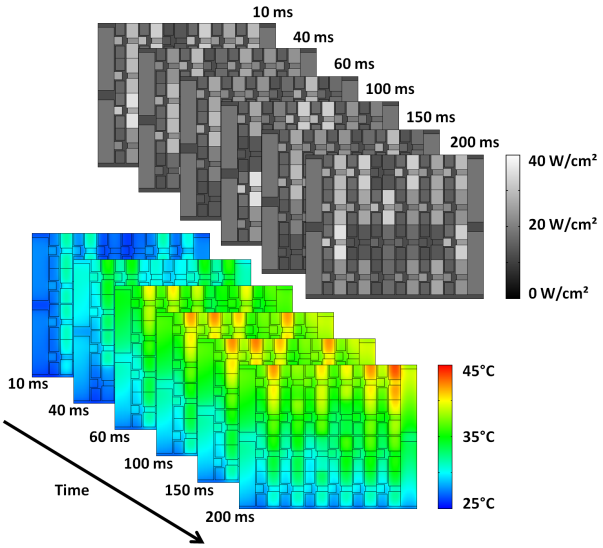


Fig. 6: Power dissipation and temperature evolution in die 0

A. Accuracy evaluation

The target structure is meshed with $100\mu m \times 100\mu m$ cells according to its microchannels dimensions, for a total of $214 \times 265 \times 18 \simeq 1.10^6$ cells. As 3D-ICE cannot handle problems of this size, only a quarter of the target structure (i.e. 6 tiles) was used for validation against 3D-ICE.

The target was first simulated with 3D-ICE using time steps ranging from $10\mu s$ to $1ms$, to determine its convergence rate. We found that 3D-ICE results do not differ by more than $10^{-3}°C$ for time steps $\Delta t < 10\mu s$. Therefore, we used the thermal maps produced by 3D-ICE with a time step $\Delta t = 10\mu s$ as a reference. The proposed method was used with a time step $\Delta t = 20\mu s$ to satisfy the stability criterion of the Euler method for the target structure meshed with $100\mu m \times 100\mu m$ cells.

Figure 7 shows the maximum temperature difference between the reference and the proposed method and 3D-ICE used with different time steps.

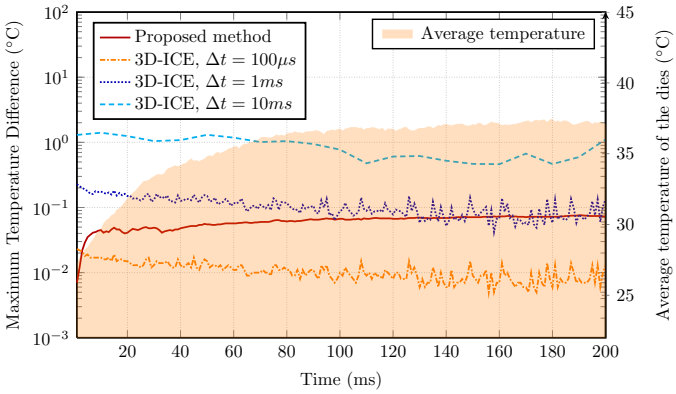


Fig. 7: Accuracy of the proposed method and 3D-ICE used with different time steps. The average temperature of the dies is reported in the background.

Figure 7 shows that the maximum difference at any point during the entire simulation between the temperatures obtained with the proposed method and the reference is lower than $0.1°C$ (1% w.r.t temperature increase) which is well within the accuracy of 3D-ICE. Figure 8 shows that the difference is lower than 0.06 95% of the time for the proposed method, with a more compact distribution than 3D-ICE.

One objection to our approach is that 3D-ICE could be used with larger time steps, especially considering that integrated circuits have a time constant RC on the order of tens of milliseconds [1], effectively filtering fast power generation variations. To test this hypothesis, we take average power values over $10ms$ periods for all functional units and we simulate the target using 3D-ICE with a time step $\Delta t = 10ms$. Results in Figure 7 show a significant loss of accuracy, with 3D-ICE producing values in excess of $1.4°C$ when compared to the reference, or 14% w.r.t the temperature increase.

B. Performance evaluation

To evaluate the performance of the proposed model, we compare to 3D-ICE v2.1 (the most recent version at the time

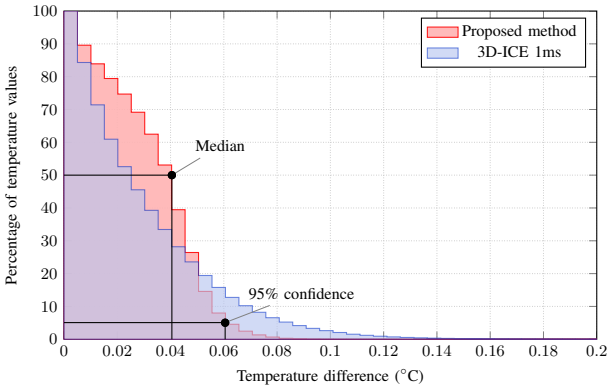


Fig. 8: Statistical distribution of deviation from reference for the proposed method and 3D-ICE

of writing), the fastest available tool that can simulate the transient behavior of liquid-cooled 3D-ICs with a guaranteed level of accuracy. All our experiments were carried on an Intel Xeon 3.4 GHz with 4 hyper-threaded cores and 16 GB RAM.

We consider several sections of the target, with sizes ranging from 68,000 to 407,000 cells. It is worth noting that 3D-ICE fails to work on more than 407,000 cells due to its memory limitations. We simulated each section for $200ms$, and used a time step $\Delta t = 1ms$ for 3D-ICE, so that the proposed method and 3D-ICE work at the same accuracy level (see Figure 7). Execution times and memory usage are reported in Figure 9.

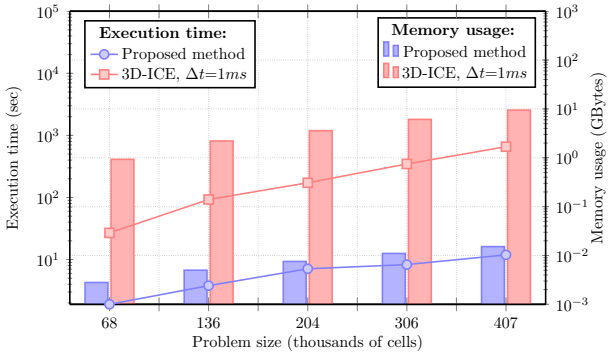


Fig. 9: Performance comparison between the proposed method and 3D-ICE at the same accuracy level

Results show that the proposed method outperforms 3D-ICE by almost two orders of magnitude. This result is mainly due to the fact that our model is highly parallel while 3D-ICE is based on a solver with an inherently limited parallelism [12]. It should be underlined that any meaningful design space exploration requires the evaluation of several configurations. Fast simulations and the ability to model realistic designs are mandatory characteristics for a usable thermal model.

Our experiments confirm the parallelization potential of the proposed method: performance scales very well on the four-core machine used for the experiments, with an almost perfect utilization of all cores. The execution time and the memory usage of the proposed method grow linearly with the problem size, as opposed to 3D-ICE, whose solver has polynomial complexity. The low memory footprint of our method allows our tool to model liquid-cooled 3D ICs of realistic dimensions.

VI. Conclusions

This work presented a novel approach for the transient thermal modelling featuring low memory requirements and a high degree of parallelism. Our experiments show that the proposed approach is 60 times faster and requires 600 times less memory than state-of-the-art-models, while maintaining the same level of accuracy. Future work will exploit the advantages of the proposed approach for modelling two-phase cooling and using non-uniform meshes.

VII. Acknowledgment

The authors acknowledge the support of the ReSMiQ, the Microsystems Strategic Alliance of Quebec. The ReSMiQ unites the effort of researchers in seven Quebec Universities to promote advanced research in computer architectures, circuit design, testing and verification and system-level design.

References

[1] P. Chaparro, J. Gonzlez, G. Magklis, Q. Cai, and A. Gonzlez. Understanding the thermal implications of multi-core architectures. *IEEE Trans. Parallel Distrib. Syst.*, pages 1055–1065, 2007.

[2] S. D. Conte and C. W. D. Boor. *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill, New York, 3rd edition, 1980.

[3] Z. Feng and P. Li. Fast thermal analysis on GPU for 3D-ICs with integrated microchannel cooling. In *Proc. Int. Conf. Computer-Aided. Design (ICCAD)*, pages 551–555, Nov. 2010.

[4] A. Fourmigue, G. Beltrame, G. Nicolescu, and E. M. Aboulhamid. A linear-time approach for the transient thermal simulation of liquid-cooled 3D ICs. In *CODES+ISSS*, pages 197–206, 2011.

[5] S. Herbert, S. Garg, and D. Marculescu. Exploiting process variability in voltage/frequency control. *IEEE Trans. VLSI Syst.*, 20(8):1392–1404, 2012.

[6] J. Howard and al. A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. *Solid-State Circuits, IEEE Journal of*, 46(1):173–183, jan. 2011.

[7] S. G. Kandlikar. Single-phase liquid flow in minichannels and microchannels. In *Heat Transfer and Fluid Flow in Minichannels and Microchannels*. Elvier Science, 2006.

[8] 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 IC with non-uniform heat flux. *ASME Conference Proceedings*, 2009(43499):1249–1258, 2009.

[9] H. Mizunuma, Y.-C. Lu, and C.-L. Yang. Thermal modeling and analysis for 3-D ICs with integrated microchannel cooling. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(9):1293 –1306, sept. 2011.

[10] K. K. Rangan, G.-Y. Wei, and D. Brooks. Thread motion: fine-grained power management for multi-core systems. In *Proc. of the 36th Intl. Symposium on Computer architecture*, ISCA '09, pages 302–313, New York, NY, USA, 2009. ACM.

[11] G. Schrom, F. Faillet, and J. Hahn. A 60 MHz 50 W fine-grain package integrated VR powering a CPU from 3.3 V. In *Applied Power Electronics Conf.*, 2010.

[12] 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. Int. Conf. Computer-Aided. Design (ICCAD)*, pages 463–470, San Jose, USA, Nov. 2010.

[13] A. Vincenzi, A. Sridhar, M. Ruggiero, and D. Atienza. Accelerating thermal simulations of 3D ICs with liquid cooling using neural networks. In *Proc. of the Great Lakes Symposium on VLSI*, GLSVLSI'12, pages 15–20. ACM, 2012.

[14] Z. M. Zhang. Nano/microscale heat transfer. In *Overview of Macroscopic Thermal Sciences*. McGraw-Hill, Toronto, 2007.