

# Efficient Transient Thermal Simulation of 3D ICs with Liquid-Cooling and Through Silicon Vias

Alain Fourmigue, Giovanni Beltrame, Gabriela Nicolescu  
Ecole Polytechnique Montreal, Canada

alain.fourmigue@polymtl.ca, giovanni.beltrame@polymtl.ca, gabriela.nicolescu@polymtl.ca

**Abstract**—Three-dimensional integrated circuits (3D ICs) with advanced cooling systems are emerging as a viable solution for many-core platforms. These architectures generate a high and rapidly changing thermal flux. Their design requires accurate transient thermal models. Several models have been proposed, either with limited capabilities, or poor simulation performance.

This work introduces an efficient algorithm based on the Finite Difference Method to compute the transient temperature in liquid-cooled 3D ICs. Our experiments show a 5x speedup versus state-of-the-art models, while maintaining the same level of accuracy, and demonstrate the effect of large through silicon vias arrays on thermal dissipation.

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

## I. INTRODUCTION

The constant struggle for higher integration has led to the design of three-dimensional integrated circuits (3D ICs). The feasibility and advantages of 3D ICs was proven [9] and all the major players in the market are showing considerable interest in their commercial exploitation.

The high density of 3D ICs introduces new challenges related to temperature control; the considerable thermal flux in a very compact device can lead to high temperatures and uneven temperature distributions. This can ultimately reduce the performance and expected lifetime of these devices [13].

Standard air-cooling can be insufficient to remove excess heat from high-performance 3D ICs. Flowing liquid coolant in microchannels etched on the silicon substrate was proposed as a viable alternative [10]. Exploring the design space to find the optimal cooling solution for a particular device requires fast (to improve the design schedule), accurate (to avoid re-spins), and memory efficient (to model realistic devices) thermal models.

Rapid simulation is particularly important to allow the designer to evaluate multiple alternatives while the layout and power output of the device are still subject to change.

It is worth noting that the majority of academic and commercially available thermal models are limited to steady-state behaviour, and are unable to take into account a continuously variable power output, at it is introduced by modern power management techniques. The steady-state assumption doesn't hold for all systems and and transient thermal models are becoming more and more widespread.

Recent transient models require a high computational load and vast memory use, leading to long simulation times [13], [15], [16], or low accuracy [8]. In addition, limitations of these

models prevent calculating the effect that large through silicon via (TSVs) arrays can have on the temperature. Such arrays are becoming a standardized way to interconnect processor and memory layers.

In this work, we introduce a highly efficient and accurate algorithm for computing the temperature in 3D ICs with liquid cooling and TSVs. The proposed algorithm was evaluated on three realistic 3D IC stacks. For all of our test cases, the error was always lower than  $0.5^{\circ}C$  when compared to a reference simulator. Our algorithm is also 5 times faster than current state-of-the-art algorithms, and scales better to larger circuits than any other approach found in literature. In addition, the high efficiency of our algorithm allows us to study the effect of TSV array (such as WideIO pads) on the heat dissipation of a target system.

This paper's contributions can be summarized as:

- an algorithm which computes the transient temperature of liquid-cooled 3D ICs with linear-time complexity and second-order accuracy in time.
- a highly parallel and scalable implementation
- a study of the effect of large TSV arrays (i.e. WideIO) on the overall temperature of a 3D IC

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; Section VI study the thermal influence of TSVs arrays; Finally, Section VII draws some conclusions and discusses future work directions.

## II. CONVENTIONAL THERMAL MODELING OF LIQUID-COOLED 3D ICs

Liquid-cooled 3D ICs are made of several stacked dies with microchannels etched into the silicon bulk of one or more dies, as shown in Figure 1. A coolant flows in the microchannels and removes the heat from the chip.

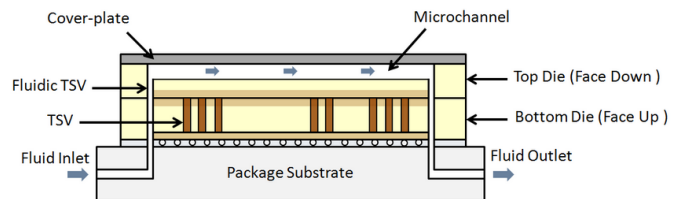


Fig. 1: A typical liquid-cooled 3D IC

Thermal modeling of such 3D ICs is conventionally performed by solving the heat equation which governs the temperature in the chip. The heat equation corresponds to an energy balance on a volume  $V$ :

$$\int_V c_v \frac{\partial T}{\partial t} dv = \int_S \vec{q} \cdot \vec{ds} + \int_V P_{vol} dv \quad (1)$$

The term on the left is the variation of energy in  $V$ ,  $c_v$  is the volumetric heat capacity of the material in  $V$ . The first term on the right is the heat exchanged through  $V$ 's surface  $S$ , the heat flux  $q$  through  $S$  can be conductive, convective or advective. The second term on the right represents the power generated inside the volume  $V$ .

Solving Equation (1) allows to determine the temperature everywhere in the chip at any time. 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 (1) is a complex task. Because of the complex structure of liquid-cooled 3D ICs with different materials and different types of heat transfer, it is usually not possible to analytically solve Equation (1) [3]. Rather, a numerical approach like the Finite Difference Method (FDM) is usually adopted [10], [13], [6], [15], [8], [16].

To apply the FDM, the chip is first divided into small cubic cells, referred to as thermal cells. To handle the different materials the circuit is divided in such a way that each thermal cell contains only one type of material, either solid or liquid.

Figure 2 shows the discretization of the chip into thermal cells and the heat flux  $q$  received by a thermal cell from the neighboring cells in the *north*, *south*, *east*, *west*, *top*, and *bottom* directions.

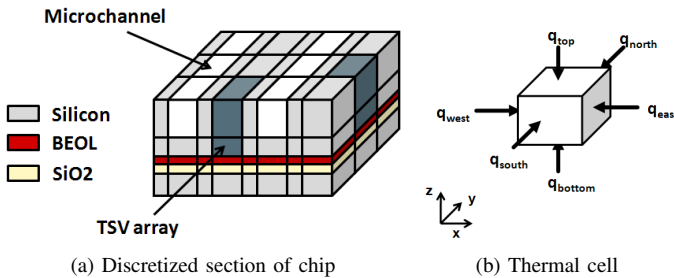


Fig. 2: Discretization of the chip into thermal cells

Using first-order approximations of the heat flux  $q$ , Equation (1) is discretized and transformed into an algebraic equation which can be written in matrix form as follows:

$$\mathbf{C} \frac{d\mathbf{T}}{dt} = \mathbf{G}\mathbf{T} + \mathbf{P} \quad (2)$$

where  $\mathbf{C}$  is the capacitance matrix,  $\mathbf{G}$  is the conductance matrix  $\mathbf{T}$  is the temperature vector and  $\mathbf{P}$  the power vector. More details on the discretization and the derivation of Equation (2) can be found in [15], [8].

In this paper, we focus on solving Equation (2) which is a computationally intensive task. Thermal models of practical interest need to solve Equation (2) with hundreds of thousands of unknown variables [15], [3]. This makes simulations

very expensive in terms of computational and memory requirements. We propose a very efficient algorithm to solve Equation (2) with linear-time complexity and second-order accuracy in time.

### III. RELATED WORK

Recently, a lot of research has been dedicated to the thermal modeling of liquid-cooled 3D ICs [10], [13], [6], [15], [8], [16]. Most of these models discretize the heat equation using the Finite Difference Method which works particularly well with the rectangular geometries of ICs. Then, these models need to solve the algebraic problem formed by Equation (2). Table I summarizes the characteristics of the most recent thermal models.

Existing thermal models of liquid-cooled 3D ICs can be classified as steady-state or transient models. Steady-state models [10], [13], [6] use preconditioned iterative methods, and are generally very efficient (using GPUs [6] in particular). However, they do not consider the transient behavior of the simulated device, and can lead to inaccurate results. This is particularly important when considering circuits with power management features, where the power dissipation changes continuously.

Sridhar *et al.* developed a simulation library, called 3D-ICE [15] which computes the transient temperature of liquid-cooled 3D ICs using the backward Euler method. 3D-ICE was validated against a prototype and has a good level of accuracy, but lacks simulation speed due to its high memory requirements and mostly single-threaded implementation. 3D-ICE runs in  $O(N^{1.7})$  time where  $N$  is the number of cells used to discretize the system, leading to unfeasible simulation times for devices of realistic dimensions.

Fourmigue *et al.* [8] 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 (truncation) error. The splitting error is difficult to control and reduces the confidence in the results. To overcome this limitation, Fourmigue *et al.* proposed another linear-time algorithm [7] based on the forward Euler method. Though accurate, this method has to satisfy a convergence criterion whose value depends on the spectral radius of the matrices in Equation 2. For detailed models of TSV based devices, the convergence criterion dramatically increases the number of iterations and leads to longer simulation times.

Liu *et al.* [12] use an iterative method called GMRES to compute the transient temperature in liquid-cooled 3D ICs. At each time step of the transient simulation, an approximate solution of the discretized heat equation is computed iteratively starting from an initial guess. Despite the use of a preconditioner to increase the rate of convergence, Liu *et al.* report a small speedup: their GPU implementation is only to  $2.3\times$  faster than a parallel LU solver. This makes GMRES orders of magnitude slower than the model proposed in this paper running on CPU. In addition, the accuracy of GMRES was not validated against real measurements or other tools.

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

| Author        | Analysis     | Solver                      | Complexity       | Implementation |
|---------------|--------------|-----------------------------|------------------|----------------|
| Kim [10]      | Steady state | Successive Under Relaxation | $O(NM)$          | CPU            |
| Mizunuma [13] | Steady state | Gauss-Seidel                | $O(NM)$          | CPU            |
| Feng [6]      | Steady state | Conjugate Gradient          | $O(NM)$          | GPGPU          |
| Sridhar [15]  | Transient    | SuperLU                     | $O(N^{1.7})$     | CPU            |
| Fourmigue [8] | Transient    | Operator Splitting          | $O(N)$           | CPU            |
| Liu [12]      | Transient    | GMRES                       | $O(NM)$          | GPGPU          |
| Vincenzi [16] | Transient    | Neural Network              | $O(N^x)^\dagger$ | GPGPU          |

$N$  is the problem size,  $M$  is the number of iterations to achieve convergence

$^\dagger 1 < x < 2$  depending on the chosen accuracy

Vincenzi *et al.* propose to accelerate transient analysis using neural networks [16]. They train a neural network to mimic the thermal behavior of the modeled chip using a conventional thermal simulator such as 3D-ICE. After training, 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 [16] 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 ALGORITHM

The proposed algorithm efficiently computes the transient temperature in liquid-cooled 3D ICs by solving Equation (2). The proposed algorithm is based on the D'Yakonov scheme [5] (DYA) which is second-order accurate in time, unconditionally stable [14], and has linear-time complexity.

To compute the transient temperature, the time is divided into small time steps  $\Delta t$ . The time derivative in Equation (2) is discretized using the Crank-Nicholson scheme which leads to:

$$\mathbf{C} \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t} = \mathbf{G} \frac{\mathbf{T}^{n+1} + \mathbf{T}^n}{2} + \mathbf{P}^{n+\frac{1}{2}} \quad (3)$$

where  $\mathbf{T}^n$  is the temperature expressed at time  $t_n = n\Delta t$  and  $\mathbf{P}^{n+\frac{1}{2}}$  is the power generated between time points  $t_n$  and  $t_{n+1}$ .

The transient temperature in the circuit can be computed by solving Equation 3 at each time step in the simulation. A naive approach to solve Equation 3 is a gaussian elimination which has cubic complexity. More sophisticated solvers exist to take advantage of the sparse nature of the matrix involved, e.g. 3D-ICE uses SuperLU [4] with a complexity of  $O(N^{1.7})$  in the number of thermal cells. The proposed algorithm has instead linear-time complexity and second order accuracy in time.

Consider  $\mathbf{A} = \frac{\Delta t}{2} \mathbf{C}^{-1} \mathbf{G}$  and  $\mathbf{U}^{n+\frac{1}{2}} = \Delta t \mathbf{C}^{-1} \mathbf{P}^{n+\frac{1}{2}}$ . Equation 3 can re-organized as follows:

$$(I - \mathbf{A}) \mathbf{T}^{n+1} = (I + \mathbf{A}) \mathbf{T}^n + \mathbf{U}^{n+\frac{1}{2}} \quad (4)$$

The algebraic operator  $A$  is expressed in terms of its components  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  along the  $x, y, z$  directions.

$$(I - (\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)) \mathbf{T}^{n+1} = (I + \mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z) \mathbf{T}^n + \mathbf{U}^{n+\frac{1}{2}} \quad (5)$$

The main idea in DYA is to add a perturbation term  $PT$  to the right-hand side of Equation 5.

$$PT = (\mathbf{A}_x \mathbf{A}_y + \mathbf{A}_y \mathbf{A}_z + \mathbf{A}_x \mathbf{A}_z) (\mathbf{T}^{n+1} - \mathbf{T}^n) + \mathbf{A}_x \mathbf{A}_y \mathbf{A}_z (\mathbf{T}^{n+1} + \mathbf{T}^n) \quad (6)$$

Adding the perturbation term, Equation 5 becomes:

$$(I - \mathbf{A}_x)(I - \mathbf{A}_y)(I - \mathbf{A}_z) \mathbf{T}^{n+1} = (I + \mathbf{A}_x)(I + \mathbf{A}_y)(I + \mathbf{A}_z) \mathbf{T}^n + \mathbf{U}^{n+\frac{1}{2}} \quad (7)$$

which is equivalent to the following equations:

$$(I - \mathbf{A}_x) \mathbf{T}^{n+\frac{1}{3}} = (I + \mathbf{A}_x)(I + \mathbf{A}_y)(I + \mathbf{A}_z) \mathbf{T}^n + \mathbf{U}^{n+\frac{1}{2}} \quad (8)$$

$$(I - \mathbf{A}_y) \mathbf{T}^{n+\frac{2}{3}} = \mathbf{T}^{n+\frac{1}{3}} \quad (9)$$

$$(I - \mathbf{A}_z) \mathbf{T}^{n+1} = \mathbf{T}^{n+\frac{2}{3}} \quad (10)$$

where  $\mathbf{T}^{n+\frac{1}{3}}, \mathbf{T}^{n+\frac{2}{3}}$  are intermediate variables with no physical meaning.

To compute the transient temperature, we solve in sequence these three equations at each time step in the simulation. Since  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  are tridiagonal matrices, these equations can be solved in linear-time using the Thomas algorithm [2].

Consider the chip is discretized with a grid of dimensions  $N_x, N_y, N_z$  along the  $x, y, z$  directions. Matrices  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  contain respectively  $N_y N_z, N_x N_z$  and  $N_x N_y$  independent tridiagonal systems of size  $N_x, N_y$  and  $N_z$ . Since the Thomas algorithm has linear complexity with the size of the system, the proposed model requires  $O(3 \times N)$  operations at each time step.

In addition, all the independent systems of matrices  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  can be solved in parallel. We take advantage of this large amount of parallelism through a simple OpenMP implementation. Our experiments show a speedup of  $3.59 \pm 0.41$  times on a two-hyperthreaded four-core machine, demonstrating near-ideal parallel code execution. The proposed DYA algorithm is expected to scale well on machines with a higher number of cores.

Aside from efficiency, accuracy is also a mandatory characteristic for compact thermal models. The global error of a

finite difference method such as DYA can be estimated by summing the local errors over all the iterations [2]. At each iteration, DYA introduces an error  $PT$  on top of the local truncation error  $O(\Delta t^3)$  due to the Crank-Nicolson scheme. Considering that the matrices  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  are proportional to  $\Delta t$ , and that  $\mathbf{T}^{n+1} - \mathbf{T}^n = \frac{dT}{dt} \Delta t + O(\Delta t)$ , it can be shown that  $PT$  is  $O(\Delta t^3)$ . Therefore, the global error for simulating a time interval  $L$  is  $\frac{L}{\Delta t} \times O(\Delta t^3) = O(\Delta t^2)$ . The DYA method is second-order accurate in time. More in-depth proof can be found in [14].

## V. EXPERIMENTAL RESULTS

To evaluate the proposed DYA algorithm, we developed a thermal model for liquid-cooled 3D ICs, using the conventional approach based on the Finite Difference Method. We implemented the proposed DYA algorithm in C and used it as the solver for our thermal model.

We compare our thermal model against the state-of-the-art 3D-ICE [15], a freely distributed tool, which was itself validated against a prototype and a commercial simulator. For a fair evaluation, we also implemented a parallel version of the SPO solver [8] and we compare against this method which is, to the best of the authors' knowledge, the most efficient approach in the state-of-the-art. Our experiments aimed at evaluating the accuracy and the performance of our algorithm on a variety of realistic 3D stacks.

### A. Accuracy evaluation

To validate the accuracy of the proposed DYA algorithm, we simulate three structures, referred to as test stacks, with a typical material composition, dimensions, and realistic power output. The test stacks are composed of two stacked dies (die A and die B) that surmount a silicon interposer. Each stack is equipped with a different cooling systems, so that DYA can be validated under different temperature and dissipation conditions. Test stack 1 and test stack 2 use liquid-cooling, while test stack 3 is air-cooled. Figure 3 shows the geometry of the test stacks used in our experiments.

The transient behavior of the test stacks is simulated for  $200ms$ . This time interval is sufficiently long to study the transient response of the circuit, being five times the  $RC$  time constant of the test stacks structure (around  $40ms$ ). Figure 4 shows the power dissipation of die A and die B. To model a realistic power dissipation case (e.g. power island), die A and die B are switched off for  $50ms$  after  $100ms$  and then switched on again. Thus, we can also determine if the model captures accurately a decrease in temperature.

We assume adiabatic boundary conditions on all the faces of test stack 1 and test stack 2, since liquid-cooled 3D ICs are usually enclosed by a manifold [15]. The vertical and bottom faces of test stack 3 are assumed to be adiabatic and the top surface is assumed to be convective. In all experiments, the initial temperature is set at  $26^\circ C$  (ambient temperature). For test stacks 1 and 2, the temperature of the incoming coolant is also set at  $26^\circ C$ .

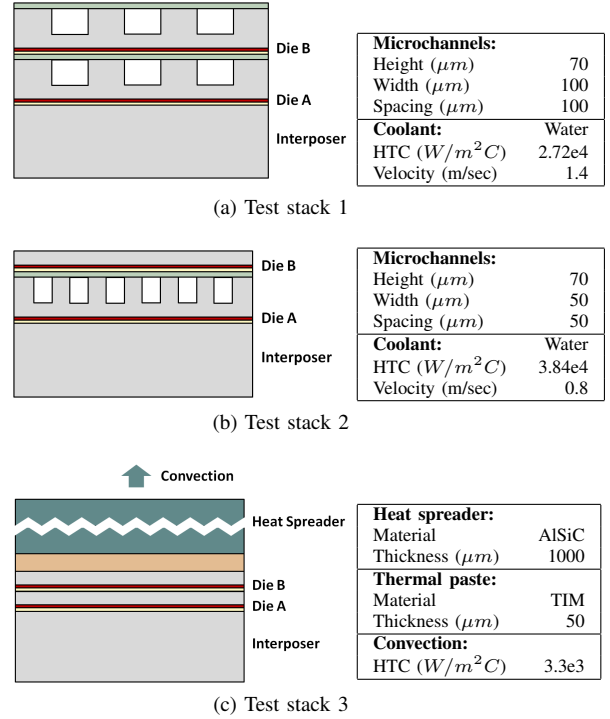


Fig. 3: Test stacks used for the accuracy evaluation

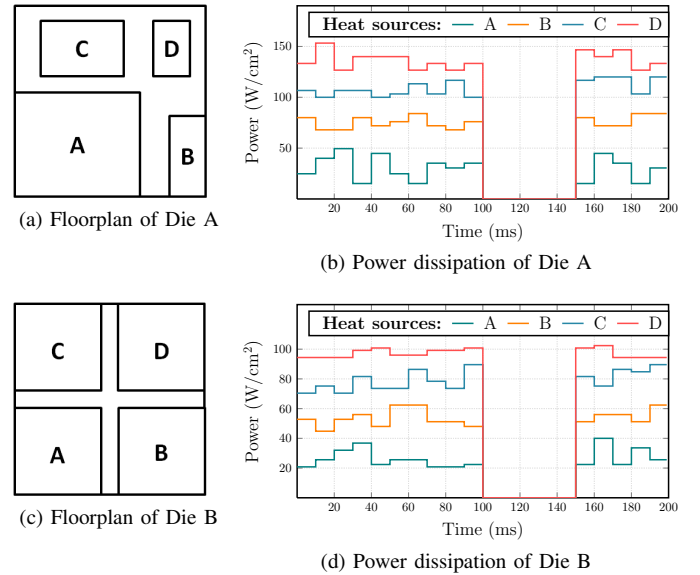


Fig. 4: Power dissipation of the test stacks

Test stack 1 and test stack 2 are meshed with  $100\mu m \times 100\mu m$  cells and  $50\mu \times 50\mu m$  cells, respectively, according to their microchannels dimensions. Test stack 3 is meshed with  $100\mu m \times 100\mu m$  cells.

Due to the robust validation of 3D-ICE (against a commercial simulator and a prototype), we take it as the accuracy reference for our simulations. We first simulate the test stacks with 3D-ICE using time steps ranging from  $5\mu s$  to  $1ms$ , to

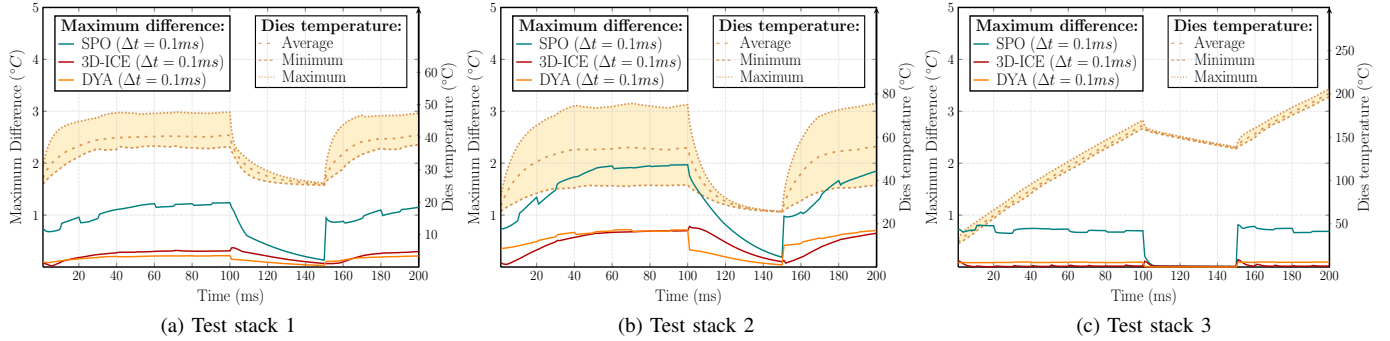


Fig. 5: Maximum difference between reference and DYA, SPO and 3D-ICE used with  $\Delta t = 0.1ms$

determine its convergence rate. We find that 3D-ICE results do not differ by more than  $10^{-3}^{\circ}C$  for time steps  $\Delta t < 5\mu s$ . Therefore, we use the thermal maps produced by 3D-ICE with a time step  $\Delta t = 5\mu s$  as a reference.

Table II presents the maximum temperature difference with reference found for the three test stacks simulated with DYA, SPO and 3D-ICE, using different time steps.

TABLE II: Maximum temperature difference ( $^{\circ}C$ ) between reference and 3D-ICE, SPO, DYA used with different time steps

|              |        | Time step $\Delta t$ |       |       |        |        |
|--------------|--------|----------------------|-------|-------|--------|--------|
|              |        | 1ms                  | 0.5ms | 0.1ms | 0.05ms | 0.01ms |
| Test stack 1 | 3D-ICE | 3.8                  | 1.9   | 0.37  | 0.17   | 0.02   |
|              | SPO    | 10.2                 | 5.36  | 1.24  | 0.63   | 0.12   |
|              | DYA    | 4.9                  | 2.2   | 0.22  | 0.08   | 0.004  |
| Test stack 2 | 3D-ICE | 7.9                  | 4.02  | 0.78  | 0.37   | 0.042  |
|              | SPO    | 16.6                 | 8.76  | 1.97  | 1.10   | 0.179  |
|              | DYA    | 8.5                  | 5.32  | 0.72  | 0.25   | 0.009  |
| Test stack 3 | 3D-ICE | 1.17                 | 0.65  | 0.068 | 0.008  | 0.001  |
|              | SPO    | 4.75                 | 3.10  | 0.81  | 0.43   | 0.092  |
|              | DYA    | 3.6                  | 2.0   | 0.097 | 0.026  | 0.008  |

Figure 5 shows the evolution of the maximum temperature difference between the reference, DYA, SPO, and 3D-ICE during the simulation. The temperature of the dies is also reported.

The results show that DYA is four to five times more accurate than SPO when using the same time step. DYA also remains very close to 3D-ICE in terms of accuracy for each of the considered time steps. One notable difference is test stack 3, where DYA performs slightly worse than 3D-ICE when using the same time step. This difference is to all effects and purposes negligible, as it is generally less than  $1^{\circ}C$  with a temperature variation above  $200^{\circ}C$ . It is worth noting that DYA is orders of magnitude faster than 3D-ICE and it could run simulations at higher speed even when considering smaller time steps, effectively making DYA the most accurate model between the three presented here.

### B. Performance evaluation

For the performance evaluation, the time step is chosen so that the three methods, DYA, SPO, and 3D-ICE give results within  $1^{\circ}C$  of accuracy, which is more than sufficient for the

needs of 3D IC early design. According to Table II, we chose a time step  $\Delta t = 0.1ms$  for 3D-ICE and DYA and a time step  $\Delta t = 0.01ms$  for SPO.

Execution times and memory usage are reported in Figure 6.

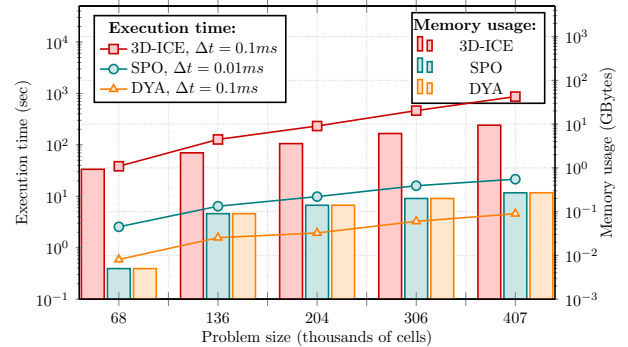


Fig. 6: Performance comparison between 3D-ICE, SPO and DYA at the same accuracy level

Results show that the proposed method outperforms 3D-ICE by almost two orders of magnitude. This result is mainly due to the fact that 3D-ICE uses a solver taking  $O(N^{1.7})$  time and requiring large amounts of memory. In addition, our model is highly parallel, while 3D-ICE is based on a solver with an inherently limited parallelism [15]. Considering SPO, Figure 6 shows that DYA is 5 to 6 times faster than SPO at the same level of accuracy. This is due to the fact that SPO has to run with much smaller time steps to overcome the splitting error introduced by the method. 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 runs in  $O(N^{1.7})$  time.

To the best of the authors' knowledge, whether considering accuracy or performance, our model surpasses any other model to be found in literature.

## VI. CASE STUDY: INFLUENCE OF LARGE TSV ARRAYS

We simulate a circuit composed of two stacked dies over a silicon interposer and equipped with a conventional heat sink (See Figure 7). Die 1 is based on the ARM Cortex A9 [11]. It contains 18 32nm processors with a maximum power dissipation of 31W. Die 2 is based on Samsung WideIO DRAM [9] with a maximum power dissipation of 1.2W. The two dies are interconnected by several WideIO arrays of  $4 \times 8$   $10\mu\text{m}$  copper TSVs with a pitch of  $50\mu\text{m}$ .

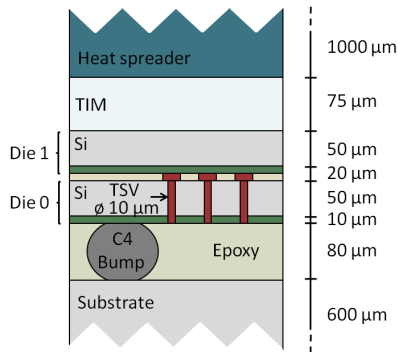


Fig. 7: Structure of the circuit

We take advantage of the efficiency of the DYA algorithm to perform a full chip analysis, modeling each TSV individually. This would not be feasible with any other state-of-the-art approach.

Our experiments show that an effect smaller than a  $0.5^\circ\text{C}$  difference for a temperature increase of  $25^\circ\text{C}$  from ambient. This means that TSVs arrays do not significantly affect the temperature of the circuit, but detailed simulation is needed to take into account the communication between layers and prevent hotspots.

To further validate our model, we compare against COMSOL [1]. To reduce the simulation time to acceptable values, we model only a  $1\text{mm} \times 1\text{mm}$  region in COMSOL as shown by Figure 8. The maximum difference between COMSOL and our model is  $0.6^\circ\text{C}$ .

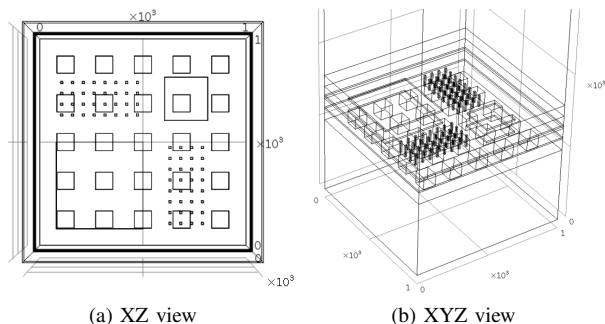


Fig. 8: COMSOL model of the circuit

## VII. CONCLUSIONS

This work presented an efficient algorithm for the transient thermal modelling of 3D ICs. The proposed algorithm features high speed, accuracy, and a high degree of parallelism. Experiments performed on three different 3D stacks show that our approach is 5 times faster than state-of-the-art-models, while maintaining the same level of accuracy. Or, conversely, our approach can provide 5 times the accuracy of state-of-the-art models when considering equal simulation times. The high efficiency of our algorithm allowed us to study the influence of large TSV arrays (such as WideIO pads) on the thermal dissipation of a target system. Future work will exploit our model to determine the effects of 3D interconnections and integration strategies on the global thermal dissipation of 3D ICs.

## VIII. ACKNOWLEDGMENT

The authors acknowledge the support of the ReSMiQ, the Microsystems Strategic Alliance of Quebec.

## REFERENCES

- [1] Comsol. <http://www.comsol.com>, July 2013.
- [2] S. C. Chapra and R. P. Canale. *Numerical Methods for Engineers*. McGraw-Hill, Inc., New York, NY, USA, 6th edition, 2009.
- [3] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang. Thermal simulation for vlsi systems. In *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, Boston, 2000.
- [4] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20:720–755, 1999.
- [5] E. G. D'Yakonov. Difference schemes of second-order accuracy with a splitting operator for parabolic equations without mixed derivatives. *Zh. Vychisl. Mat. I Mat. Fiz.*, 4:935–941, 1964.
- [6] Z. Feng and P. Li. Fast thermal analysis on GPU for 3D-ICs with integrated microchannel cooling. In *Proc. ICCAD*, pages 551–555, Nov. 2010.
- [7] A. Fourmigue, G. Beltrame, and G. Nicolescu. Explicit transient thermal simulation of liquid-cooled 3d ics. In *Proc. of DATE'13*, pages 1385–1390, 2013.
- [8] 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 *Proc. of CODES+ISSS'11*, pages 197–206, 2011.
- [9] J.-S. Kim and al. A 1.2v 12.8gb/s 2gb mobile wide-i/o dram with 4x128 i/os using tsv-based stacking. In *ISSCC*, pages 496–498, feb. 2011.
- [10] 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.
- [11] J. Koppanalil, G. Yeung, D. O'Driscoll, S. Householder, and C. Hawkins. A 1.6 ghz dual-core arm cortex a9 implementation on a low power high-k metal gate 32nm process. In *VLSI-DAT*, pages 1–4, april 2011.
- [12] X.-X. Liu, Z. Liu, S.-D. Tan, and J. Gordon. Full-chip thermal analysis of 3d ics with liquid cooling by gpu-accelerated gmres method. In *Quality Electronic Design (ISQED)*, 2012 13th International Symposium on, pages 123–128, march 2012.
- [13] H. Mizunuma, Y.-C. Lu, and C.-L. Yang. Thermal modeling and analysis for 3-D ICs with integrated microchannel cooling. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 30(9):1293–1306, sept. 2011.
- [14] J. Qin. The new alternating direction implicit difference methods for solving three-dimensional parabolic equations. *Applied Mathematical Modelling*, 34(4):890–897, 2010.
- [15] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschweiler, and D. Atienza. 3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling. In *Proc. of ICCAD*, pages 463–470, 2010.
- [16] A. Vincenzi, A. Sridhar, M. Ruggiero, and D. Atienza. Accelerating thermal simulations of 3D ICs with liquid cooling using neural networks. In *Proc. of GLSVLSI'12*, pages 15–20. ACM, 2012.