

TIMiTIC: A C++ based Compact Thermal Simulator for 3D ICs with Microchannel Cooling

Piotr Zając

Department of Microelectronics and Computer Science, Lodz University of Technology

*Corresponding Author: pzajac@dmcs.pl, +48 42 631 26 53

Abstract

CFD simulation of liquid-cooled integrated circuits is very time-consuming. However, in the case of ICs with straight integrated microchannels, the flow is very regular and very often, the CFD simulation is not needed to correctly predict the hydrodynamic and thermal behaviour. In this paper, we present a thermal simulator, written in C++, which significantly reduces the thermal simulation time of such chips, while maintaining acceptable accuracy. For a typical test chip we have observed the average relative error 3.7 % for the worst analysed steady-state case, while the simulation time was only 10 ms, compared to 20 minutes for full CFD simulation.

1 Introduction

The increased power density in vertically-stacked 3D integrated circuits (IC) creates the need for novel cooling concepts. Microchannel-based cooling has been proposed by many researchers as a potential solution [14][17]. To verify the effectiveness of this novel cooling idea, finite element analysis is typically used [7][8]. However, the FEA-based thermal simulation of such ICs is very time-consuming [9], mostly due to the fact that it includes full CFD simulation. With 3D stacks composed of multiple layers, each layer comprising many microchannels and each microchannel consisting of hundreds or even thousands of mesh nodes, the simulation time may be very long. Therefore, many simpler compact models have been proposed. An excellent survey of these approaches is presented in [15], where authors concentrate on speeding up the simulations by splitting the 3D chip into blocks to take advantage of the parallelization. One of the most popular simulators for 3D ICs with liquid cooling is 3D-ICE [1], written in C. The 3D ICE thermal model is in general very thoroughly described, however not enough attention is paid to accurate modelling of solid-fluid convection, which is indispensable for obtaining correct results. For example, authors use the Shah and London Nusselt number correlation [11], which is only valid from a certain distance from the inlet, while in [1] it is used for the entire channel. In fact, in some cases the proposed formula may result in obtaining negative Nusselt numbers, which is obviously not physically realistic. Therefore, in this paper we describe TIMiTIC: Thermal simulator of Integrated Microchannel cooling for Three-dimensional Integrated Circuits, written in C++, which provides the user with high flexibility in terms of solid-fluid modelling. Additionally, the program is free for download [16] and has very user-friendly interface which gives the user many other simulation options.

2 Thermal model

The simulator uses a RC grid model to represent the vertically stacked 3D chip (see Figure 1). Each chip layer is divided into a grid of “solid” nodes. Since the thickness of the layers is much smaller than their width and length, only one node is

used in our model in the vertical direction (z direction). In y direction (along the channels), the number of nodes is specified by the user. In the x direction, the number of nodes is equal to the number of channels. If the layer contains microchannels, additional “fluid” nodes are added to the grid, with fluid flow represented by temperature-controlled heat flow sources. Thus, for example a chip with 2 layers with channels and 2 layers without channels, with 10 channels per layer and 100 nodes along the channel will be discretized in our model into 6000 nodes.

The model uses only one node for the solid around the channel which is equivalent to assuming that the temperature in the solid around the channel is uniform. This is a valid assumption, considering that the conduction resistance of silicon cells is significantly lower than the convection resistance between solid and fluid. To prove this point, we show in Figure 2 the typical temperature distribution in and around the channel obtained using full CFD simulation. The temperature of the silicon around the channel is almost the same and therefore, this entire area can be treated as one node in our model.

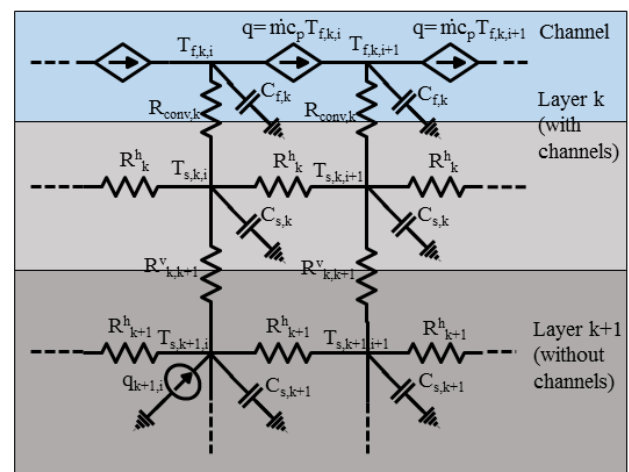


Figure 1: Thermal model of the chip (side view). Note that the model is 3D but only 2D version is shown for clarity.

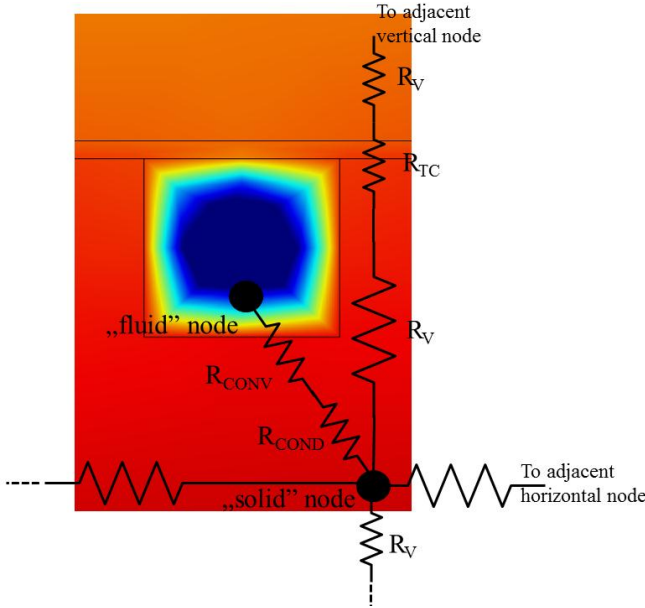


Figure 2: Thermal model of the chip (cross-sectional view, no capacitances are shown for clarity). Microchannel cover is modelled by thermal contact with resistance R_{TC} .

The simulator enables specifying every chip parameter (dimensions, material properties, channel parameters, power trace for each node, etc.) in the XML file. The simulator reads the XML file, uses these data to create the conductance matrix G , capacitance matrix C and heat source matrix Q , and then the equation (1) is solved.

$$C \frac{dT}{dt} = -GT(t) + Q \quad (1)$$

Figure 1 shows a part of the chip with two layers, one with channels and one without channels. R_h is the equivalent horizontal conduction resistance, R_v is the equivalent vertical conduction resistance, C_s is the capacitance of the solid, C_f is the capacitance of the fluid, \dot{m} is the mass flow, c_p is fluid's specific heat, R_{conv} is the convection resistance between solid and fluid, T_s is the solid temperature and T_f is the fluid temperature. k denotes the layer number and i denotes the node number in horizontal direction. Each node can optionally dissipate power, represented by the the heat flow source q .

The vertical walls of the chip stack are treated as adiabatic boundaries. The top and bottom surfaces of the chip can be defined by the user as convective boundaries, otherwise they will be also considered adiabatic. Channel outlets are simply modelled by connecting the last temperature-controlled heat source to ground (ambient temperature). Similarly, modelling channel inlets is performed by connecting the first temperature-controlled heat source to the inlet temperature.

Since the RC model in solids is well known and understood [13], in this paper we only describe the modelling of convection resistance between the solid and the fluid and the advection in the fluid.

2.1 Convection between solid and fluid

The accurate modelling of the convection process between the solid and the fluid is crucial for obtaining correct simulation results but at the same time, it is the most difficult. In fact, modelling this process with a resistance is not entirely correct due to the fact that the heat transfer coefficient along the heated channel changes with the power profile along the channel. However, it has been shown that for microchannel-cooled integrated circuits it is acceptable to use a simple resistance (2).

$$R_{conv} = \frac{1}{hA_{wall}} \quad h = \frac{kNu}{D_h} \quad (2)$$

A_{wall} is the surface of the contact wall between solid and fluid, k is the conductivity of the fluid and D_h is the hydraulic diameter of the channel. Nu is the Nusselt number and is a function of the distance from the inlet. Multiple approximations have been suggested in literature which define the Nusselt number, therefore in our software there are several different Nusselt number correlations which can be chosen by the user. In particular, Shah and London [11] and Hausen [12] correlations are implemented. It is also worth mentioning that, for maximum flexibility, there is an option of specifying custom Nusselt number correlation, in which the user directly chooses the Nu values along the channel.

2.2 Advection modelling

Like already demonstrated in [1], heat transport in a channel can be described as temperature-controlled heat source. Indeed, the amount of heat moved along the channel is proportional to the mean temperature of the fluid (3).

$$q = \dot{m}c_p T_m \quad (3)$$

c_p is the specific heat of the fluid and \dot{m} is the mass flow $\dot{m} = \rho Au$, where ρ is the fluid density, A is the channel cross-section and u is the average fluid velocity. It should be noted there that the mean temperature of the fluid T_m is actually calculated as velocity-weighted average temperature across the channel cross-section, see (4). It is important to remember this when comparing the channel temperatures produced by TIMiTIC with the results obtained with CFD simulation (see section 4.1).

$$T_m = \frac{\int uT dA}{\int u dA} \quad (4)$$

3 Description of the TIMiTIC simulator

The simulator is written in modern C++. Fast and reliable libraries are used both for solving a linear system in steady-state simulation and for solving a system of differential equations in transient simulation (see section 3.3 for more information). Additionally, functions and types from Boost C++ library [4] are frequently used in the code. In its current version the program works with command-line interface however the graphical interface will be included in the future version of the software. The advantages of the simulator are:

- the use of modern, fast mathematical libraries as solvers, which reduce simulation time with respect to full CFD simulation,
- the ability to define the Nusselt number along the channel: the user can choose from multiple Nusselt number correlations found in literature or specify a custom function,
- the possibility to define thermal contacts between layers, which enables taking into account very thin layers without adding more nodes to the model (and therefore without increasing simulation time).

```

<config>
  <logFilename>tsim.log</logFilename>
  <dataFilename>data.txt</dataFilename>
  <resultsFilename>results.txt</resultsFilename>
  <steadyStateSolver>eigenSparseLU</steadyStateSolver>
  <materials>
    <material1>
      <name>silicon</name>
      <thermalCond>130.0</thermalCond>
      <density>2330.0</density>
      <specificHeat>700.0</specificHeat>
    </material1>

    Other materials can be defined here...

  </materials>
  <fluidMaterials>
    <material1>
      <name>water</name>
      <thermalCond>0.591</thermalCond>
      <density>1000.0</density>
      <specificHeat>4184.0</specificHeat>
      <dynamicViscosity>0.000653</dynamicViscosity>
      <PrandtlNumber>3.56</PrandtlNumber>
    </material1>

    Other fluid materials can be defined here...

  </fluidMaterials>
  <channels>
    <channel1>
      <name>channel1</name>
      <width>0.0001</width>
      <height>0.0001</height>
      <fluidVelocity>0.5</fluidVelocity>
      <nusseltCorr>ShahLondon</nusseltCorr>
      <nusseltNb>3.61</nusseltNb>
      <inletTemperature>300.0</inletTemperature>
      <fluidMaterial>water</fluidMaterial>
    </channel1>

    Other channels can be defined here...

  </channels>
  <transientSimulation>
    <enabled>>false</enabled>
    <initialTemp>300.0</initialTemp>
    <ambientTempTransient>300.0</ambientTempTransient>
    <initialTime>0.0</initialTime>
    <simulationTime>0.25</simulationTime>
    <timeStep>0.001</timeStep>
    <odeintStepper>runge_kutta_dopri5</odeintStepper>
    <absoluteError>1.0e-6</absoluteError>
  </transientSimulation>
</config>

```

Listing 1 XML configuration file (part 1)

3.1 Input XML file

A part of a sample configuration XML file is shown in Listing 1. Most of the parameters are self-explanatory, the user can define materials and their properties, channel properties and simulation parameters. What is important, it is possible to choose between several Nusselt number correlations: in this example, Shah and London correlation is specified. The user can also define his own custom correlation. Once these parameters are defined, the next section of the XML file describes the concrete simulated chip (see Listing 2). After specifying basic chip parameters like chip size and grid granularity, it is possible to define the entire 3D chip stack (layers section). There is no limit to the number of layers that can be inserted. Each layer can contain channels (but not necessarily has to), the user can also choose the specific positions for each channel in the layer. Additionally, it is possible to define thermal contacts between layers. Thermal contacts were introduced to model thin layers without adding more nodes to the model (and therefore without increasing simulation time). The thermal contacts are simply modelled by adding a thermal resistance between two nodes based on the conductance of the thermal contact specified in W/m^2K . It should be also noted that it is possible to add convection on top or bottom of the chip by specifying a heat transfer coefficient on these boundaries.

3.2 Input power trace

For power input, the program allows defining group of nodes which correspond to chip units. Then, for each unit, power in particular time instants can be defined. To make this easier, the program allows the user to only specify the time instants when the power for a given unit changes. Therefore, defining the power trace becomes easy, especially if there are few changes in power dissipation during simulation time.

3.3 Solvers

The user can choose from two steady-state solvers (ublas [5] and EigenSparseLU [3] and four transient solvers from the odeint library [2]: Runge-Kutta-Dopri, Runge-Kutta-Cash-Karp, classic 4th order Runge-Kutta and Bulirsch-Stoer. Apart from classic Runge-Kutta, these transient solvers use adaptive time stepping, so there can be more time points in the simulation that the number specified by the user. However, using adaptive stepping significantly decreases the probability that the solver will fail to converge. Of course basic parameters like simulation time, simulation step, absolute and relative error can be also defined in the XML file.

3.4 Output files

The output text files are the simulation log file, the data file and the results file. The log file contains the information about any potential errors which may have occurred during simulation. The data file contains the information about conductance, capacitance and heat source matrices. It may be useful to analyse this file in case the temperature results are different from what was expected. Finally, the results file contains the calculated temperature values. The user can

```

<chip>
  <xsize>0.003</xsize>
  <yssize>0.01</yssize>
  <xgridsize>6</xgridsize>
  <ygridsize>100</ygridsize>
  <ambientTemperature>300.0</ambientTemperature>
  <layers>
    <layer1>
      <name>Bottom layer</name>
      <thickness>0.0002</thickness>
      <materialName>silicon</materialName>
      <channelName>channel1</channelName>
      <channelPositions>0 1 2 3 4 5</channelPositions>
    </layer1>
    <layer2>
      <name>Spreader</name>
      <thickness>0.001</thickness>
      <materialName>copper</materialName>
    </layer2>
  </layers>

```

Other layers can be defined here...

```

</layers>
<thermalContacts>
  <thermalContact1>
    <layer1Name>Bottom layer</layer1Name>
    <layer2Name>Top layer</layer2Name>
    <conductance>1000000.0</conductance>
  </thermalContact1>

```

Other thermal contacts can be defined here...

```

</thermalContacts>
</chip>

```

Listing 2: XML configuration file (part 2)

specify which temperatures are to be printed in the file: they can be temperatures across a cross-section or all temperatures in a layer. Note that the resulting temperatures are written to a file in the format which makes them easy to plot using free Gnuplot software [6].

4 Validation of the proposed simulator

To validate our model, simulations of a 3D chip with microchannels were performed in COMSOL Multiphysics [10] using fully coupled CFD simulation. Then, the obtained results were compared with those produced by TIMiTiC for four test cases (named A, B, C and D). The test chip represented the structure of real liquid-cooled ICs, however to reduce the time needed for CFD simulation, the test chip width and the number of channels were reduced. Thus, the test chip may be seen as a model of a part of a real IC (see Figure 3).

Table 1 lists the chip parameters which were held constant for all simulations. The material properties for silicon, copper and water were taken from appropriate material databases. The variable test chip parameters were top and bottom layer thickness, TIM thickness, cover material thermal conductivity, inlet temperature, fluid velocity and the power dissipated in the bottom layer (see Table 2). Additionally, to provide power variation across the layer, the bottom layer was divided into four regions with different power dissipation.

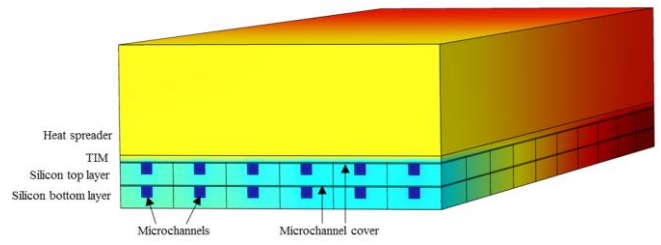


Figure 3: Test chip for model validation showing sample temperature results obtained with CFD simulation in COMSOL (inlet side view)

Table 1: List of constant test chip parameters

Test chip parameters	
Size	3 mm x 10 mm
Bottom layer	Silicon
Top layer	Silicon
Thermal interface material	$k=2$ W/mK, $\rho=2000$ kg/m ³ , $c_p=700$ J/kgK
Microchannel cover	$\rho=3000$ kg/m ³ , $c_p=700$ J/kgK, $h=10$ μ m
Heat spreader	Copper, $h=1$ mm
Top layer power	6 W
Cooling fluid	Water
Channel height	100 μ m
Channel width	100 μ m

Table 2: List of variable test chip parameters

Variable chip parameters				
Name	Case A	Case B	Case C	Case D
Layer thickness	200 μ m	150 μ m	250 μ m	150 μ m
TIM thickness	60 μ m	120 μ m	40 μ m	100 μ m
Cover thermal conductivity	10 W/mK	20 W/mK	15 W/mK	25 W/mK
Inlet temperature	300 K	320 K	330 K	310 K
Fluid velocity	0.5 m/s	1 m/s	1.1 m/s	1.5 m/s
Bottom layer power, region 1	7 W	6 W	10 W	5 W
Bottom layer power, region 2	1 W	3 W	0 W	5 W
Bottom layer power, region 3	4 W	2 W	6 W	2 W
Bottom layer power, region 4	4 W	4 W	1 W	7 W

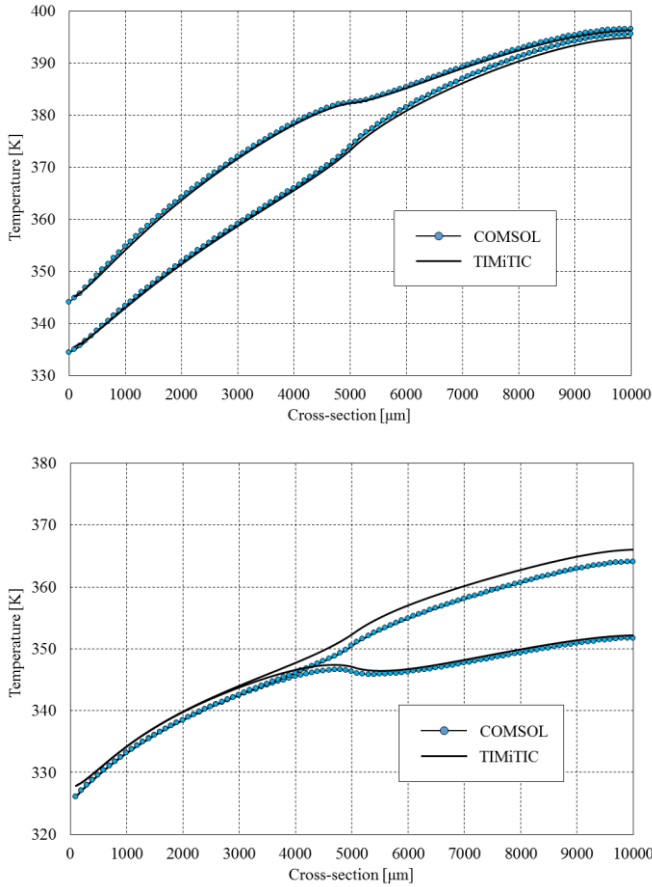


Figure 4: Comparison of temperature results for two chip cross-sections, case A (top, best match) and case D (bottom, worst match)

Table 3: Summary of obtained results for all cases (steady-state)

Summary of steady-state results				
	Case A	Case B	Case C	Case D
Average error	0.4 K	0.81 K	0.4 K	1.32 K
Maximum error	0.76 K	1.61 K	1.33 K	2.14 K
Average relative error	0.56%	2.1%	1.25%	3.73%
Maximum relative error	1.75%	7.6%	6.91%	11.1%

4.1 Steady state simulations

In Figure 4 sample results for cases A and D are presented, which correspond to the best and worst case, respectively, in terms of agreement with COMSOL results. The data points represent the solid temperatures of the bottom layer and are plotted across leftmost and rightmost chip cross-sections. The shape of the curves are identical and the temperature differences are minimal for case A and acceptably small for

case D. The Table 3 summarises the results obtained for all cases. For each case, we have compared the temperatures in 100 points across two cross-sections (200 points in total) and calculated the absolute and relative error. The relative error was calculated by dividing the absolute error by the temperature rise with respect to the inlet temperature. After a detailed analysis of results, the following conclusions were made:

- The predicted mean temperatures in the channel are very close to those reported by COMSOL, consequently the modelling of advection process is very accurate.
- The model for conduction across solids is also very good, although some error is introduced due to the fact that the equation for calculating equivalent horizontal conduction resistance when channels are present in a layer is currently simplified, perhaps a more complex formula is needed.
- The most important source of differences between TIMiTIC and COMSOL results is the calculation of convection resistance, in particular the Nusselt number in the entry region of the channel. The Nusselt number vary greatly in the entry region, so even a small error in the formula may result in a slightly inaccurate value of convection resistance. As a result, the faster the flow is, the longer the entry region is and the bigger impact of potentially incorrect Nusselt number values have on the final chip temperatures. This is the reason why the errors for the case A are very low (slowest flow, short entry region) and they are higher for the case D (fastest flow, long entry region).

4.2 Transient simulations

To better quantify the potential error in transient simulation, it was decided to use the parameter values for the best steady-state case (case A). The power values were also the same as for the case A, they were applied to the chip at the time $t=0$ and remained unchanged afterwards. In Figure 5 we traced the obtained heating curves for both COMSOL and TIMiTIC. Across all simulated time instants the error does not exceed 1.5 K, which is a very good result. An interesting observation is that TIMiTIC gives slightly lower temperatures in the final heating phase. The reason is quite straightforward and comes from the model assumptions: the microchannel cover layer is modelled as thermal contact in TIMiTIC and therefore its thermal capacity is neglected, resulting in a slightly lower total chip capacitance and slightly faster heating. The simulation took over one hour for COMSOL and just over 30 minutes for TIMiTIC, resulting in a speedup factor of only 2. It is worth emphasizing here that such comparison should always be treated with caution, because the total simulation time depends on many parameters like relative error, adaptive stepping algorithm, etc. Therefore, the comparison of transient simulation times across different simulators is very difficult and may vary significantly depending on the particular simulated chip and power trace. Nevertheless, the obtained speedup may be seen as slightly disappointing, especially considering the fact that for steady state simulation a much higher speedup was observed.

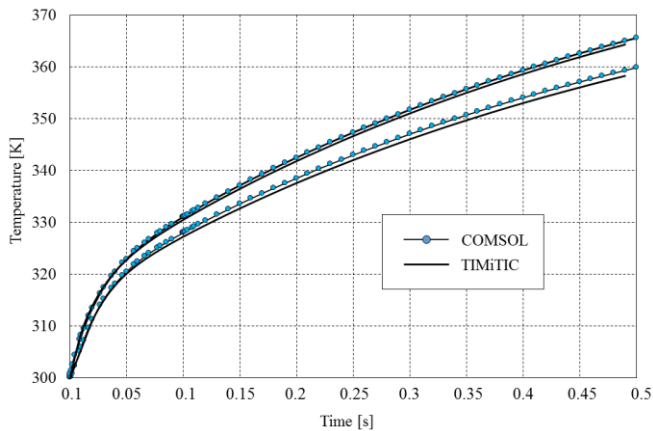


Figure 5: Comparison of heating curves obtained by transient simulation.

5 Conclusions

In this paper, the C++ based simulator called TIMiTIC was presented, which allows a fast thermal simulation of ICs with integrated microchannels. In comparison with other approaches, more focus was given on modelling the convection resistance between solid and fluid. The validation with respect to full CFD simulation showed a very good agreement (average absolute error 1.3 K, average relative error 3.7% for the worst analysed case). The largest differences were observed for fast flows, which indicates that the Nusselt number in the entry region might be modelled better. The speedup with respect to CFD was around five orders of magnitude for steady-state. However, the transient simulation in TIMiTIC was only 2 times faster than in COMSOL. The simulator is available of the webpage [16] and will be gradually improved in future versions. In particular, graphical interface will be developed, to make the interface more user friendly. In terms of the model improvements, currently there are some constraints regarding the chip structure, which will be removed in future software versions. An effort will be also made to optimize the transient simulation to further reduce the simulation time.

Literature

[1] Sridhar et al. "3D-ICE: Fast Compact Transient Thermal Modeling for 3D-ICs with Inter-Tier Liquid Cooling", Proc. Int'l Conf. Computer-Aided Design (ICCAD), pp. 463-470, 2010

[2] K. Ahnert, M. Mulansky, "Odeint – solving ordinary differential equations in C++", Proc. AIP Conf. Numerical Analysis Applied Mathematics, vol. 1389, pp. 1586-1589, 2011.

[3] Eigen C++ Linear Algebra Library, <http://eigen.tuxfamily.org>

[4] B. Schling, "The Boost C++ Libraries", XML Press, 2011

[5] Basic Linear Algebra Library, www.boost.org/libs/numeric/ublas

[6] T. Williams, C. Kelley, "Gnuplot 4.5: an interactive plotting program", 2011, <http://gnuplot.info>

[7] P. Zając, C. Maj, A. Napieralski, "Peak temperature reduction by optimizing power density distribution in 3D ICs with microchannel cooling", Microelectronics Reliability, Volume 79, 2017, pp.488-498

[8] P. Zając, C. Maj, M. Galicia, A. Napieralski, "Coupled thermo-fluidic simulation for design space exploration of microchannels in liquid-cooled 3D ICs", 2016 MIXDES - 23rd International Conference Mixed Design of Integrated Circuits and Systems, Lodz, Poland, 2016, pp. 257-261

[9] P. Zając, A. Napieralski, "Novel thermal model of microchannel cooling system designed for fast simulation of liquid-cooled ICs", Microelectronics Reliability, 2018, vol. 87, no. , p. 245-258, ISSN: 0026-2714

[10] COMSOL Multiphysics® Modeling Software, software available at www.comsol.com

[11] R. K. Shah, and A. L. London, "Laminar Flow Forced Convection in Ducts", Academic Press, New York, 1978

[12] W. M. Kays, "Numerical Solutions for Laminar Flow Heat Transfer in Circular Tubes," Trans. ASME,77, , 1955, pp. 1265–1274

[13] T.-Y. Wang, C. C.-P. Chen, "3-D-thermal ADI: A linear-time chip level transient thermal simulator", IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 21, no. 12, pp. 1434-1445, Dec. 2002

[14] P. Zając, M. Janicki and A. Napieralski, "On the applicability of single-layer integrated microchannel cooling in 3D ICs," 2018 19th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), Toulouse, France, 2018, pp. 1-6

[15] A. Fourmigue, G. Beltrame and G. Nicolescu, "Explicit transient thermal simulation of liquid-cooled 3D ICs," 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2013, pp. 1385-1390

[16] Thermal simulator of Integrated Microchannel cooling for Three-dimensional Integrated Circuits (TIMiTIC) webpage, timitic.dmc.pl

[17] J. M. Koo, S. Im, L. Jiang and K. E. Goodson, "Integrated microchannel cooling for three-dimensional circuit architectures", ASME J.Heat Transfer, vol. 127, pp. 49-58, 2005