TIMiTIC

Thermal simulator of Integrated Microchannel cooling for Three-dimensional Integrated Circuits

User's Guide

version 1.0.0

1. General information

The main purpose of this user's guide is to describe TIMiTIC, the Thermal simulator of Integrated Microchannel cooling for Three-dimensional Integrated Circuits. The program was written in C++.

The structure of the simulated chip, material properties and all simulation parameters can be defined by the user in the input XML file. The chip described in the file is then discretized into multiple cells, i.e., it is represented as a grid of RC elements. Heat transfer due to fluid flow in channels is modelled as temperature-controlled heat flow sources. Once the 3D grid is built, i.e, conductance matrix **G** and capacitance vector **C** are calculated**,** TIMiTIC uses the power trace defined in a separate XML file to add heat sources to the grid (vector **Q**). Finally, fast C++ libraries for solving the system of linear equations (steady-state) and system of differential equations (transient simulation) – see Eq. 1.

$$\boldsymbol{C}\frac{d\boldsymbol{T}}{dt} = -\boldsymbol{G}\boldsymbol{T}(t) + \boldsymbol{Q} \tag{1}$$

2. The simulated 3D chip

This document only briefly describes the thermal model, for more detailed information about the model, please read Author's publications [11].

2.1 Layers and channels

The model assumes that the simulated chip is a vertical stack of multiple layers. Each layer may be composed of a different material. For each material, the user can specify the values of thermal conductivity, density and specific heat. Each layer can contain channels: the user can specify the number and position of the channels, as well as the parameters of the fluid material: thermal conductivity, density, specific heat, Prandtl number and dynamic viscosity. Additionally, the user defines the size of the channels, the fluid velocity and the inlet fluid temperature. A very important aspect of the simulator is the option to define the Nusselt number along the channel, see section 4 for more information.

Currently, the simulator is limited by the following constraints when designing layers and channels:

- All layers have the same size in lateral directions (x, y directions)
- The microchannels are straight and are etched in the same direction (y direction)
- The aspect ratio of microchannels is rectangular
- The channels in the same layer have the same size, inlet temperature and inlet velocity (but channels in different layers can be different)

2.2 Chip discretization (building 3D grid)

The discretization of the chip in space domain is performed in three directions x, y and z so the chip is divided into M x N x O cells. Each layer is divided into the same number of cells (M x N). In z direction, always one node per layer is used so the number of cells in z direction O is equal to the number of layers. In lateral directions, the user can specify the number of cells.

However, in x direction, the number of cells M should be at least equal to the maximum number of channels present in any layer.

When channels are inserted, additional fluid cells are added to the grid. The modelling of the fluid is one dimensional (one cell in x and z directions is used for the fluid). The number of fluid cells per layer is then equal to C x N, where N is the number of cells in y direction and C is the number of channels in the given layer.

3. Input configuration file

All parameters of the simulation except for power trace are described using XML format in the input configuration file. In what follows, all field present in the XML (marked in bold) will be described.

3.1 Config section

This section contains the general parameters of the simulation as well as all materials and channels used in the design.

- **steadyStateSolver:** the algorithm used for solving steady-state simulation. The two potential options are "eigenSparseLU" (default) and "ublas". EigenSparseLU is the algorithm from the Eigen library [1] and ublas is from boost library [3].
- **materials:** this section lists all solid materials used in the chip design. For each material the user has to define the material name (**name**), thermal conductivity (**thermalCond**), density (**density**) and specific heat (**specificHeat**).
- **fluidMaterials:** this section lists all fluid materials used in the channels section. For each material the user has to define the material name (**name**), thermal conductivity (**thermalCond**), density (**density**), specific heat (**specificHeat**), Prandtl number (**PrandtlNumber**) and dynamic viscosity (**dynamicViscosity**).
- **channels:** this section lists all channel systems used in the chip design. For each channel system, the user has to define the channel system name (**name**), channel width (**width**) and height (**height**), inlet mean fluid velocity (**fluidVelocity**), temperature of fluid at the inlet (**inletTemperature**) and fluid material (**fluidMaterial**). The name of the fluid material has to correspond to one of the names defined in **fluidMaterials** section. Additionally, the user is able to define the Nusselt number correlation for the channel (**nusseltCorr**). More information about the Nusselt number correlations is given in section 4. If the chosen correlation is constant, the field **nusseltNb** defines the Nusselt number for the entire channel, otherwise this field is unused. If the chosen correlation is custom, the field **customNusselt** defines the Nusselt number along the channel. For other correlations, this field is unused. Note that the user has to specify a string of N numbers for this field (where N is the number of cells in y direction).
- **transientSimulation:** this section defines parameters of the transient simulation. **enabled** field defines if transient simulation is to be run (options are "true" and "false" **initialState** defines if initial temperature for the transient run is taken from the end of the steady-state run or is explicitly specified in the file. If the second option is chosen, **initialTemp** defines the starting temperature of all cells in the transient simulation. **ambTempTransient** defines the ambient temperature for the transient simulation. **initialTime**, **simulationTime** and **timeStep** specify the starting time, ending time and

step for transient simulation. The field odeintStepper defines the algorithm which solves the system of differential equations. The options are “runge_kutta_dopri5” (default), “runge_kutta_cash_karp54”, “runge_kutta4” or “bulirsch_stoer”. More information about these algorithms can be found in [2]. Note that runge_kutta4 algorithm is not an adaptive-step algorithm, so if it fails converge, smaller step is probably needed in **timeStep** field. The fields **absoluteError** and **relativeError** describe the accuracy of the algorithm, please refer to [2] for more details.

3.2 Chip section

This sections describes the 3D stack itself.

- **xsize** and **ysize** define the size of the chip stack (in meters) in x and y directions respectively.
- **xgridsize** and **ygridsize** define the number of cells in x and y directions, respectively.
- **ambientTemperature** specifies the ambient temperature for steady-state simulation
- **htcTop** and **htcBottom** define the heat transfer coefficient (HTC) on the top and bottom layer of the stack, respectively. The value equal to 0.0 equal to no convection (adiabatic boundary). Note that if you consider free convection boundary to air on the top or/and bottom layer, the HTC values would be in the order of 10-20 W/m^{2}K, so this convection will have very little impact on the simulation, as the HTC values between solid and channel fluid is typically in the order of 10000-20000 W/m^{2}K.
- **layers** section defines the layers present in the 3D stack. Note that the order of the layers is important, layers should be introduced from the bottom to the top, as they are present in the 3D stack. For each layer, the user has to specify its name (**name**), thickness (**thickness**) and material (**materialName**). Note that the chosen value in **materialName** field has to correspond to one of the names defined in **materials** section. Optionally, each layer can contain a channel system, defined in the field **channelName**. Note that the chosen value in **channellName** field has to correspond to one of the names defined in **channels** section. Finally, the field **channelPositions** describes the position of the channels along the x axis. The total number of potential channel positions is equal to **xgridsize** (number of cells in x direction). The field **channelPositions** is a string of numbers, where each number j can take value from 0 to **xgridsize**-1. If the field contain the number j, it means that the channel is present at position j. Normally, if channels are present in every position, this field will contain all numbers from 0 to **xgridsize**-1.
- **thermalContacts** section describes thermal contacts between the layers. For each thermal contact, the user has to specify two layer names (**layer1Name** and **layer2Name**). Note that the values in these fields have to correspond to names layer names defined in **layers** section. Of course, the chosen layers have to be adjacent. Moreover, for each thermal contact, the user has to define its thermal conductance expressed in W/m^{2}K.

3.3 Results section

This section defines which temperatures will be written to results file. There are three types of results which can be chosen, called probes: line probe, surface probe and transient probe. The first type are simply the temperatures of chip cells along a line which goes in x or y direction (results of steady-state simulation). The second type are temperatures of all cells in a layer (results of steady-state simulation). The third type are temperatures of a single cell across time (results of transient simulation).

- **lineProbes** section describes all line probes. **name** field is the name of the probe which will be printed in the results file, **type** field defines if the line goes through solid or fluid cells (options are "Solid" or "Fluid"). **lineDirection** specifies if the line goes in x or y direction (options are "x" or "y"). **lineNb** define the number of the cross-sectioning line across which the temperatures will be recorded. Note that this number has to be between 0 and **xgridsize**-1 if the line goes in y direction or from 0 to **ygridsize-**1 if the line goes in x direction.
- **surfaceProbes** section describes all surface probes. **name** field is the name of the probe which will be printed in the results file and **layer** field specifies the name of the layer across which the temperatures are to be recorded. Note that the value in this field has to correspond to one of the layer names defined in **layers** section.
- **transientProbes** section describes all transient probes. **name** field is the name of the probe which will be printed in the results file and **layer** field specifies the name of the layer contains the cell whose temperature is to be recorded. Note that the value in this field has to correspond to one of the layer names defined in **layers** section. **type** field defines if the cell is a solid cell or a fluid cell (options are "Solid" and "Fluid"). Finally, **xNode** and **yNode** fields define the x and y coordinates of the cell in the grid. Note that **xNode** has to be a number between 0 and **xgridsize**-1 and **yNode** has to be a number from 0 to **ygridsize-**1.

4. Modelling of fluid-solid convection

The heat transfer coefficient (HTC) between the solid and fluid varies greatly along the channel. It is highest at the inlet and gradually decreases. When the flow becomes fully thermally developed, HTC stabilizes at a constant level. In literature, a dimensionless Nusselt number is used to describe the convection between the solid and the fluid. The HTC is proportional to the Nusselt number, so both quantities can be used almost interchangeably when describing the solid-fluid heat transfer along the channel. For more information about these concepts, the chapter "Internal flow" from [9] is recommended.

The function Nu(y) which describes the local Nusselt number at the distance y from the inlet is difficult to estimate because it depends on many factors: channel aspect ratio, whether the flow is thermally developing or simultaneously (hydrodynamically and thermally) developing, power dissipation profile in the solid (so called boundary conditions), etc. In literature there are many approximations of this function called Nusselt number correlations, which are based on measurements. It has to be emphasized, that there is no one perfect correlation formula which would be exact for all cases. Therefore, in TIMiTIC, there are multiple correlations which can be chosen by the user. More precisely, the user defines the desired correlation in **nusseltCorr** field in the XML configuration file. In the current version of the program, the available options are: “constant”, “Hausen“, “ShahLondon“, “CFDbased“ and “custom“.

- “constant” option means that the Nusselt number is constant along the channel, which corresponds to the case in which the fluid is fully developed hydrodynamically and thermally. In other words, it neglects all thermal entrance effects in the channel. Therefore, for some cases it may seriously underestimate the Nusselt number (and HTC) for the channel.
- “Hausen” option chooses Hausen correlation [5], which describe the average Nusselt number assuming uniform wall temperature boundary condition [9]. The program first calculates average Nu number at position x from the inlet using this correlation and then calculates the local Nusselt number at position x from the inlet. Because this correlation assumes uniform wall temperature, it may slightly underestimate the Nusselt number (and HTC) for the channel.
  The Hausen correlation is only valid for circular channels, consequently TIMiTIC uses formulas given by Shah and Sekulic [8] for the Nusselt number in fully developed flow in rectangular channels to appropriately modify the Hausen correlation depending on channel’s aspect ratio.
- “ShahLondon” option chooses the correlation for a rectangular channel given by Shah and London [4] assuming uniform heat flux boundary condition. This correlations has been used for example in [6]. However, it has to be emphasized that this correlation has some applicability limits (please consult [7] for more information), so it cannot be used for all cases.
- “CFDbased” option chooses the correlation that was developed specifically for TIMiTIC. It is based on the Shah and London correlation (see above) which was modified so that it gives the closest results to those obtained via CFD simulation. The applicability limits of the Shah and London correlation were also eliminated in this approach.
- “custom” option allows the user to explicitly set all Nusselt numbers for all cells along the channel in the **customNusselt** field.

5. Running TIMiTIC

Running TIMiTIC is available directly from a browser on the site timitic.dmcs.pl [12] (tested on Chrome and Firefox on desktop PC, not available on mobile). The code is executed on client side, thanks to Webassembly [10] technology. The graphical user interface is simple and intuitive: the user can simply click on visible buttons to upload two XML files (configuration and power trace) and click another button to run the simulation. When the program finishes, the user can save the results to disk as a text file. The simulation time for steady-state should be very fast (almost instantaneous) even for chips with a lot of cells. However, the transient simulation can take from seconds to even hours, depending on the chip configuration.

References

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

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

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

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

[6] A. Sridhar, A. Vincenzi, D. Atienza and T. Brunschwiler, "3D-ICE: A Compact Thermal Model for Early-Stage Design of Liquid-Cooled ICs," in IEEE Transactions on Computers, vol. 63, no. 10, pp. 2576-2589, Oct. 2014

[7] J. R Thome, 2006, "Wolverine Engineering Databook III"

[8] R.K. Shah, and D.P. Sekulic, (2003) "Fundamentals of Heat Exchanger Design". John Wiley & Sons, Hoboken, 941

[9] F. P. Incropera, and D. P. DeWitt. 2002. "Fundamentals of heat and mass transfer, 7th edition". New York: J. Wiley.

[10] WebAssembly, binary instruction format for a stack-based virtual machine, www.webassembly.org

[11] P. Zajac, "TIMiTIC: A C++ Based Compact Thermal Simulator for 3D ICs with Microchannel Cooling", to appear in 25th Internaltional Workshop Thermal Investigations of ICs and Systems (THERMINIC), 2019

[12] Thermal simulator of Integrated Microchannel cooling for Three-dimensional Integrated Circuits (TIMiTIC) website, timitic.dmcs.pl