Thermal Modelling of Modern Processors Using FEM and Compact Model

Melvin Galicia, Piotr Zając, Cezary Maj, Michał Szermer, and Andrzej Napieralski

***Abstract*—A variety of thermal models has been proposed to predict the temperatures inside modern processors. In this paper, we describe and compare two such approaches, a detailed FEM-based simulation and a simpler architectural compact model. It is shown that both models provide comparable results when it comes to predicting the maximal temperature, however there are also non-negligible differences when estimating thermal gradients within a chip. Furthermore, transient simulation results show some differences in temperature profile during processor heating.**

***Index Terms*—thermal simulation, finite-element method, multi-core, thermal modelling**

I. Introduction

FOR many years the analysis of thermal phenomena in high-performance processors has been a crucial research field, the results obtained by researchers have allowed identifying potential thermal problems and proposing appropriate solutions. As a result, processors' performance continue to rise although the power density dissipated inside the chip approaches the value of 100 W/cm^2 [1]. Thus, thermal modelling is of vital importance to accurately predict the temperatures inside modern multicore processors, manufactured in nanometer technologies. A wide variety of thermal modelling methodologies has been proposed by many research groups. Depending on the models' application, they vary in terms of complexity and accuracy. One of the most accurate modelling methods is based on Finite Element Method, used for example in such tools as ANSYS [2] and COMSOL [3]. However, this approach suffers from being quite time-consuming, especially if the modelled structure is complex and if many transient time points have to be calculated during transient analysis. A good trade-off between complexity and accuracy can be found in models based on thermal solvers which use Green's functions [4]. Another approach, used for example in Hotspot thermal modelling tool [5], is based on dividing the simulated structure into an array of RC elements. While in some applications such an approach is sufficient, in some cases a more detailed simulation is needed. In this paper, we analyze an Intel's quad-core processor: we use both ANSYS and Hotspot to simulate such a processor and compare the obtained results. Both tools are later compared in terms of accuracy and speed; based on the conclusions, advantages and disadvantages of both approaches are pointed out. This paper is organized as follows: in section II we describe the analyzed processor and how the power data for each chip unit was obtained. Section III presents in detail thermal modelling approaches used by ANSYS and Hotspot. In the next sections, IV and V, we present the simulations of the processor in ANSYS and Hotspot including steady and transient states, the obtained results as well as the comparison and discussion. In section VI the most important conclusions are provided.

II. Analyzed Processor

Intel's high-performance i7-2700k processor from the Sandy Bridge family was chosen for our analysis (see Fig. 1). It was fabricated in 32 nm process technology, operates nominally at 3.5 GHz frequency and contains four cores, graphics unit (PG), system agent and memory controller (SA) and L3 cache memory (L3). Additionally, in order to provide a more realistic power distribution within the core, in the floorplan that we used in the simulations (see bottom of Fig. 1) each core was divided into three separate blocks, CoreEX (corresponding to hottest execution units), CoreA and CoreB.

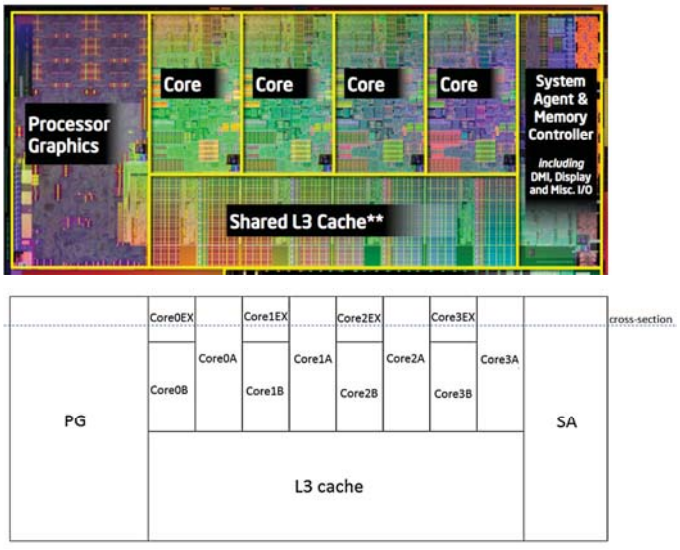

Fig. 1. Sandy Bridge die photo (top), processor floorplan used in simulations (bottom). The dashed line shows the processor cross-section used in simulations

M. Galicia, P. Zając, C. Maj, M. Szermer and A. Napieralski are with the Department of Microelectronics and Computer Science, Lodz University of Technology, ul. Wólczańska 221/223, 90-924 Lodz, Poland (emails: {mgalicia, pzajac, cmaj, szermer, napier}@dmcs.pl).

The power density in CoreEX unit is slightly higher than in other two core units. The power dissipated in each unit is composed of two main components, static power and dynamic power. Static power is a result of subthreshold and gate leakage [6] and therefore depends mostly on the number of transistors in the unit and on the temperature. On the other hand, dynamic power heavily depends on the activity of the unit, i.e., whether the unit is actively used. The static and dynamic components of power of each processor unit are shown in Tables I-III.

The detailed methodology for obtaining these data was described in our previous papers [7][8]. These data were obtained for the case when the processing cores were under heavy load, but the graphics unit was not extensively used. Since it may be interesting to run simulations for various power distributions within the processors, in this paper we use the data from Table I to analyze the following three cases of power distribution:

- all four cores and graphics unit dissipate 100% of power
- cores 0 and 1 dissipate 100% of power while cores 2 and 3 are idle and therefore only dissipate static power; graphics unit dissipates twice the dynamic power of the case 1
- all cores dissipate 50% of dynamic power; graphics unit dissipates twice the dynamic power of the case 1

The power data for all three cases of power distribution in the processor are summarized in Tables I-III.

TABLES I-III
POWER DISTRIBUTION INSIDE THE PROCESSOR

Case 1	**Power (W)**		
	static	***dynamic***	***total***
L3 cache	6.90	9.45	16.35
PG	3.46	7.00	10.46
SA	2.62	4.65	7.27
Core(0,1,2,3)A	0.86	7.24	8.10
Core(0,1,2,3)B	0.58	4.82	5.40
Core(0,1,2,3)EX	0.29	2.69	2.98

Case 2	**Power (W)**		
	static	***dynamic***	***total***
L3 cache	6.90	9.45	16.35
PG	3.46	7.00x200%	17.46
SA	2.62	4.65	7.27
Core([0,1],[2,3])A	0.86	[7.24], [0]	[8.10], [0.86]
Core([0,1],[2,3])B	0.58	[4.82], [0]	[5.40], [0.58]
Core([0,1],[2,3])EX	0.29	[2.69], [0]	[2.98], [0.29]

Case 3	**Power (W)**		
	static	***dynamic***	***total***
L3 cache	6.90	9.45	16.35
PG	3.46	7.00x200%	17.46
SA	2.62	4.65	7.27
Core(0,1,2,3)A	0.86	7.24x50%	4.05
Core(0,1,2,3)B	0.58	4.82 x50%	2.70
Core(0,1,2,3)EX	0.29	2.69 x50%	1.49

III. THERMAL MODELS

One of the most popular thermal modelling approaches is based on electrical analogy of thermal phenomena [9]. Heat transfer can be expressed as a current that flows through the thermal resistances. A voltage drop on each resistance corresponds to the temperature difference and an ambient temperature is expressed as a voltage source. The circuit that consists of these elements allows modelling the steady state of system. The transient response needs one other element which is capacitance. Both resistance and capacitance give temperature rise and fall analogous to the RC time constant. Heat flow is then described by the same differential equation as the current flow. The RC model can be successfully used in modelling of microprocessors. Typical chips have the structure that consist of silicon die covered by a spreader made of highly conductive material and a heat sink that dissipates the heat to the ambient (Fig. 2).

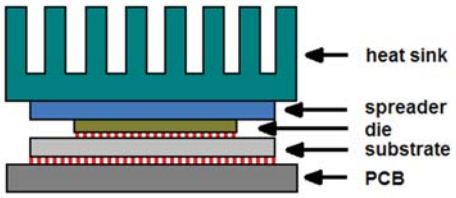

Fig. 2. Typical processor package

The RC model of such system consist three vertical conductive layers (die, spreader and heat sink) and one connective layer (heat sink to air connection). The number of RC elements depends on the division of die layer. Typically the die is divided into blocks that correspond to each unit of each core. Therefore each block needs to be modelled with resistors to each neighboring block on the same layer and adjacent ones. The spreader is divided into the block that lies under the die and trapezoidal blocks that are not cover by die. The heat sink is similarly divided as the spreader. All of these blocks are modelled with resistors in the same way as described for the die. The value of each resistance is simply calculated using thermal conductivity of material and dimensions of the block. The convection of heat to the ambient is modelled with one resistor whose resistance corresponds to the thermal resistance sink to ambient (can also include forced air convection). Each block is also modelled with one capacitor (connected between a block node and ambient node) whose value depends on thermal capacitance of the material and its volume. The graphical presentation of RC model for a system with die divided into three blocks is presented in the Fig. 3.

This model is commonly used in thermal simulators like Hotspot [5]. To increase the accuracy of model, each block can be divided into smaller blocks increasing the number of RC elements needed to describe the system. Moreover, additional layers may be included in the model such as thermal interface material (TIM) or package substrate. However, this model requires some assumptions to simplify the calculations, which in the end may lead to inaccurate results.

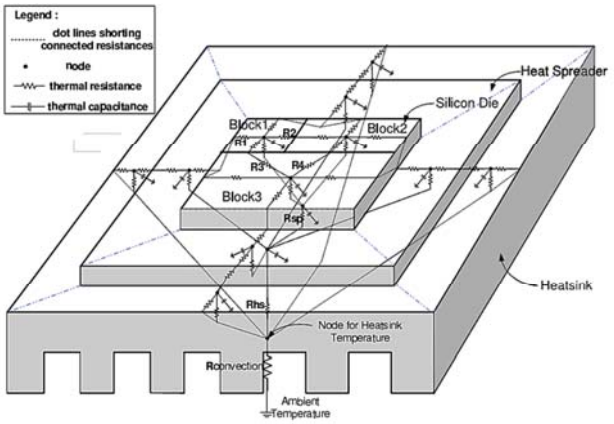

Fig. 3. Processor package model used in Hotspot [5]

A more precise simulation can be obtained by using finite element method (FEM). Then, the real shape of microprocessor and cooling system is modelled. The model is divided into many nodes depending on mesh quality and for each node the differential equation of heat transfer is solved:

$$\frac{\partial T}{\partial t} = \alpha \Delta T + \frac{q}{c_p \rho} \tag{1}$$

where T is the temperature, t is the time domain, Δ is the Laplace operator, ρ is the mass density, c_p is the specific heat capacity, α is the thermal diffusivity and q is the internal heat per unit volume. Therefore, the solution allows obtaining a complete distribution of the temperature inside the structure instead of RC model where the number of nodes is much smaller. Higher precision of the simulation is of course related to longer simulation time. The comparison of these two models may be very interesting, since the obtained results can be used to calibrate and optimize RC models.

IV. Steady-State Simulation Results

A. Simulations using ANSYS

In this section, it is used the ANSYS finite element program to model a complete chip package. The processor model was designed following the floorplan used for thermal analyses in [7]. For modelling package layers such as heat sink, heat spreader, chip die, substrate and thermal interface materials, we followed the approach found in [10]. However, since the author in [10] modeled a different processor, some parameters needed to be modified to model better the proposed Intel's i7 Sandy Bridge processor.

Typical materials have been selected for each element within the package as well as their thermal properties. The heat sink is considered to be made of aluminium and the heat spreader of copper. Thermal interface material of epoxy is placed between the heat sink and the heat spreader as well as between the heat spreader and the die, each with different thermal conductivity. Several lines of spherical solder bumps are placed right below the die at the borders to create contact with the FR-4 epoxy substrate. The ambient temperature is set to 30°C while two different convective heat transfer boundary conditions are applied to the heat sink and substrate respectively. All these layers can be clearly identified on the side view of the model of the chip package as shown in Fig. 4.

The power applied to the chip die follows the data shown in Tables I-III. It was modelled as a heat flow falling perpendicularly to the chip die surface.

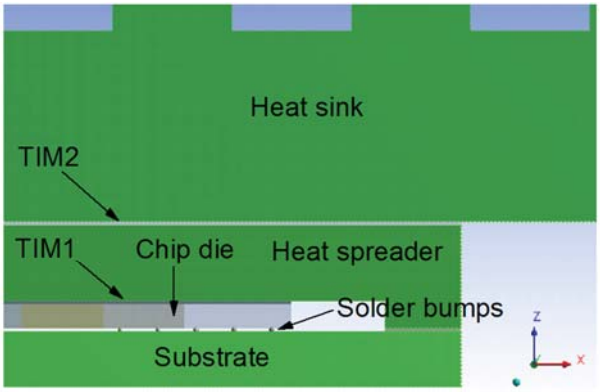

Fig. 4. Layers of the processor model designed in ANSYS

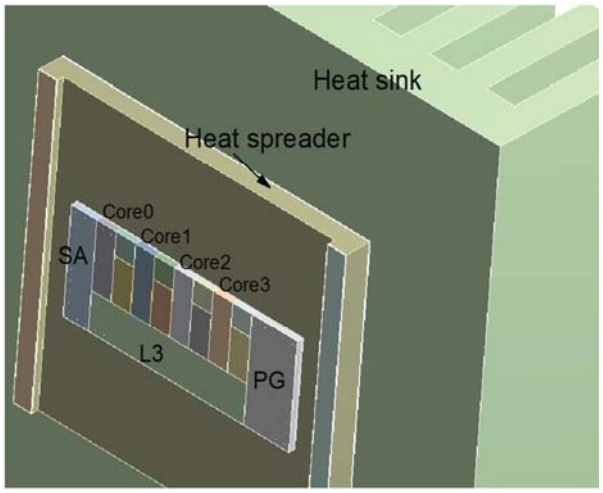

Fig. 5. Sandy Bridge processor inside the complete chip package designed in ANSYS (bottom view). Note that the substrate layer was hidden so that the chip die is visible

The complete list of elements composing the chip package with the parameters that were considered for the simulation analysis and its values are given in Table IV.

TABLE IV
Chip Package Parameters

Element	*Parameters*	*Value*
Heat Sink Aluminum	Base size (mm)	60x60x5
	Heat transfer coefficient (W/m^2K)	150
Heat spreader (Copper)	Base size (mm)	30x30x2
	Thermal Conductivity (W/mK)	600
Sandy Bridge Die (Silicon)	Size (mm)	25x10x0.6
	Thermal Conductivity (W/mK)	90
TIM_1 (Epoxy)	Size (mm)	25x10x0.1
	Thermal Conductivity (W/mK)	8
TIM_2 (Epoxy)	Size (mm)	60x60x0.1
	Thermal Conductivity (W/mK)	3
Bumps (Solder)	Radius (mm)	0.06
	Thermal Conductivity (W/mK)	1
Substrate (FR-4 epoxy)	Size (mm)	30x30x1.4
	Thermal Conductivity (W/mK)	17
	Heat transfer coefficient (W/m^2K)	10
Ambient	Temperature (°C)	30

B. Simulations using Hotspot

Hotspot processor model is less detailed, it assumes that the package is composed of four layers: processor die, thermal interface material, heat spreader and heat sink. The parameters of all these layers (dimensions, thermal conductivities, etc.) are defined in the configuration file; they all were set to the values corresponding to those used in ANSYS simulation. Note that Hotspot, unlike ANSYS, does not use heat transfer coefficient *h* for convection, but the thermal resistance; therefore, the necessary resistance was calculated based on ANSYS *h* coefficient and the heat sink surface.

The chip was divided into 128x64 blocks, each of which constitutes a node of the model's RC network. We also created a processor's floorplan file which represented exactly the Sandy Bridge processor shown in Fig. 1.

The used power data for three analyzed cases were also identical to the data used in case of ANSYS simulations (see Tables I-III). Finally, the simulations were run and the steady-state temperatures of all blocks were calculated. To visually compare two analyzed models, it was decided to plot the temperatures across the processor's horizontal cross-section (see Fig. 1). The temperature profiles are shown in Fig. 6.

C. Discussion

According to the shown temperature profiles, both tools give quite similar results regarding the maximal temperature in the chip: the maximal error is about 1.6°C for case 2. Also, the tools fairly agree about the general shape of the temperature profile. However, there is one difference that stands out for all three analyzed cases: Hotspot tool predicts lower temperatures at the edges of the chip. In other words, Hotspot model produces a higher temperature gradient in the chip whereas ANSYS temperature profiles are much more uniform. The highest difference of the temperature on the edge of the chip (case 2, left edge) reached 3.7°C. Although such a disagreement is not surprising, considering both models used differ in structure and complexity, it poses nevertheless an interesting question: where does this difference come from?

It seems that the FEM model should be more realistic as it does not use any simplification (regarding the structure modelling, not the numerical calculations). Thus, one should consider Hotspot simplifications towards possible errors. The first one is the way of modelling the heat transfer in lateral direction between different layers. If we go back to figure 3 we can see that the heat transfer from block 1 to the left trapezoid of spreader is modelled with two lateral resistances. In reality the connection of these blocks is not exactly in the same plane. They are connected only on the edge. In [11] we can see that these resistances do not take it into account. Thus, it underestimates the resistance between these two blocks. Additionally, some simplification is introduced by the fact that the convection is modelled with one resistance. If our assumptions are correct, these Hotspot simplifications lead to obtain higher gradient across the structure because of lower temperatures on the edges of the chip.

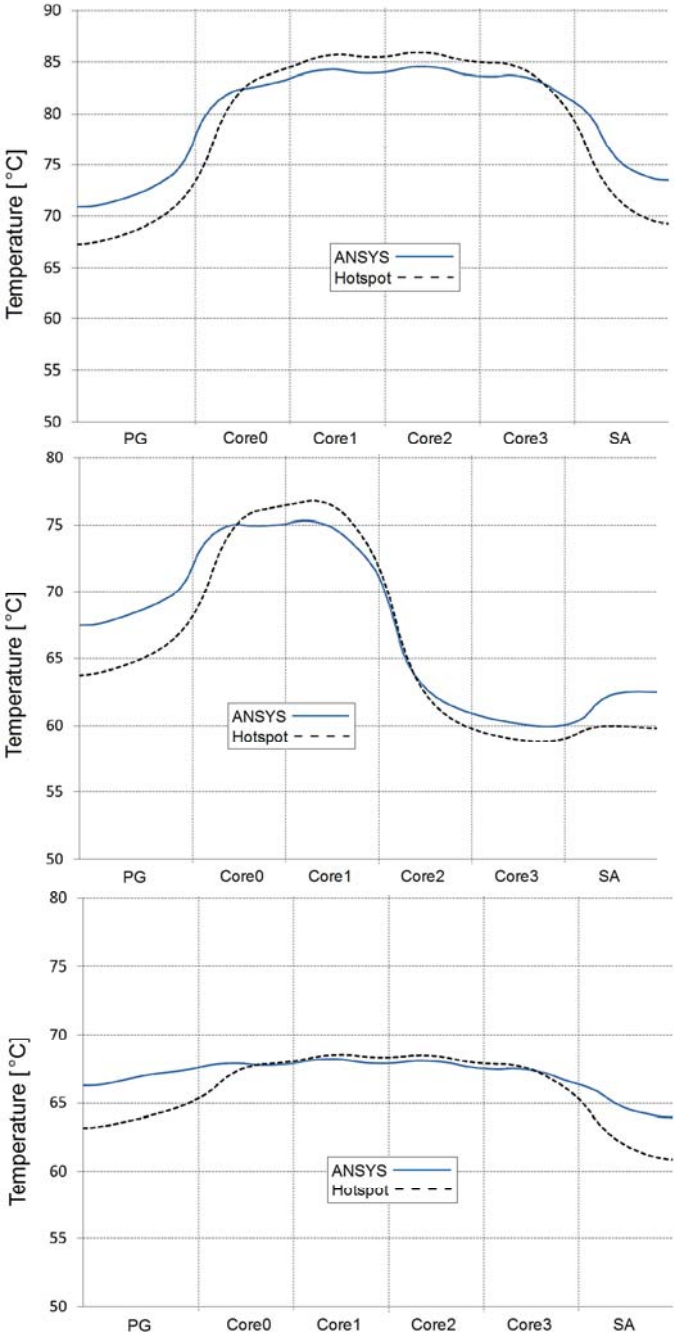

Fig. 6. Simulation results for case 1 (top), case 2 (middle) and case 3 (bottom)

What was also discovered by our simulations is that the secondary heat transfer path (through the substrate) is of secondary importance. We observed a negligible change in ANSYS results when the substrate convection was turned off. Moreover, the impact of the thermal interface material between heat sink and heat spreader (TIM2) was also quite insignificant. When it comes to simulation time, it was found that Hotspot simulation time is about the order of magnitude shorter than ANSYS simulation time for steady-state analysis (considering 128x64 grid for Hotspot and default mesh for ANSYS). On Intel's i7 processor the respective simulation durations for ANSYS and Hotspot were 40 and 5 seconds, respectively. When the grid size was reduced 4 times, to 64x32, Hotspot's simulation time was reduced to about 1.4 seconds.

V. Transient Simulation Procedure

As in previous section, simulations were carried out in ANSYS and Hotspot, however now in order to obtain the transient behaviour that precedes the steady-state. The very same power data summarized in Tables I-III was used for all three cases of power distribution in the processor as well as the chip package parameters listed in Table IV. Since it is well known that the way temperature vary over time, i.e. transient behaviour, is non-linear, a more dynamic and fast variation is expected in the early moments of the package heating. For this reason, we decided to set up a simulation where more temperature points will be calculated during the first moments and consecutively slowly decreasing the numbers of points per decade. With such set up, a more meaningful comparison is obtained in terms of differences between the FEM and compact model.

A. Simulations using ANSYS

Due to the long running-time needed for transient simulations, an adjustment in the total number of nodes that cover the package was made in ANSYS. The mesh granularity for heat sink, heat spreader, the substrate and thermal interface materials was reduced and only for the chip die was kept the same as in previous sections. Such an adjustment may have a small impact on the analysis, which will be focused only on the temperature variations inside the chip die.

The three cases were simulated for at least 60 seconds (simulation time) from where the maximal temperature inside the chip die reached a value very close to the maximal temperature in steady-state. For instance, in case 1 the maximal temperature is 86.29K in steady-state and 84.62K after 60 seconds which already accounts for the 99.5% of the maximum expected temperature. Thus, after 60 seconds a very monotonous and slow temperature increase is expected until it finally reaches steady-state value. As mentioned before, the time stepping was implemented non-linearly: the first four simulated seconds account for more than 25% of total points.

It has to be emphasized that for transient simulation the point with maximal temperature was taken. Thus, the maximal temperature may be different from the one obtained in section IV, where the cross section may not contain the hottest point. In general, the differences are negligible.

B. Simulations using Hotspot

In the same way, the necessary adjustments were performed in Hotspot in order to meet similar quantity of nodes compared with those of ANSYS. However, it seems that Hotspot transient modelling is not optimized for long run simulations. It needs a very large number of computing cycles to obtain a single transient point value. Performing a 60 seconds simulation was a prohibitively long time consuming task.

Hence, a workaround to set up similar simulation was to carry out only the simulation of initial several seconds, until we can make sure that there exists only one time constant that determines the transient behaviour of the whole package. Consequently, the steady-state temperature value can be estimated by using a mathematical model that extrapolates the temperature in the remaining time points. It can be also compared with the temperature predicted by Hotspot when only using steady-state analysis. In this methodology, we defined two regions in time. The first region, where due to the different materials that compose the microprocessor package, it is believed that more than one time constant affects the package temperature, and the second region, where it is highly possible that there is only one time constant influencing the package behaviour.

In order to have a mathematical regression of Hotspot for the same 60 seconds time frame as in ANSYS, the mathematical model needs several points from the second time region of the simulation. This means that Hotspot simulation has to run for at least more than 10 seconds. Given such requirement, some parameters modifications were applied to Hotspot in order to reach that value in a relative reasonable time. The parameters "time_step" and "min_step" in Hotspot code were accordingly modified to reduce the simulation time.

C. Mathematical regression model for Hotspot data

The mathematical temperature model used to fit the values obtained in the second time region during the Hotspot simulation must obey an exponential behaviour and it has been selected as follows:

$$T(t) = \left(T_{SS} - T_a\right)\left(1 - \exp\left(-\frac{t}{\tau}\right)\right) + T_a \qquad (2)$$

where T is the temperature, t is the time domain, T_{SS} is the steady-state temperature, T_a is the initial (ambient) temperature and τ is the positive time constant of the system. Notice that the model will provide only the regression values from the twelve second ahead, and that the model was fit with points starting from the fifth second. Hence, the values of the equation (2) coefficients with 95% confidence bounds and goodness of fit are shown in Table V.

TABLE V
Regression Coefficients with 95% confidence bounds and goodness of fit

	T_{SS} [°C]	T_a [°C]	τ	R-square
Case 1	85.05 (85.01, 85.1)	56.72 (56.71, 56.72)	17.394 (17.349,17.436)	**1**
Case 2	75.91 (75.85, 75.98)	54.15 (54.14, 54.15)	17.167 (17.085,17.250)	1
Case 3	67.69 (67.63, 67.76)	45.9 0 (45.89, 45.9)	17.268 (17.185,17.352)	1

D. Discussion

As mentioned previously, the performed transient analysis focuses in the microprocessor chip die temperatures, since the microprocessor is the source of heat.

Table VI lists the steady-state temperatures, maximum and minimum, for the whole chip die obtained in previous section using the two different methods. These values represent our boundary conditions for the transient analysis. Since, the steady-state values are calculated for infinite time, we approached those values by using transient analysis in such a way that the final temperature reached at least a 99.5% of the steady-state value for all cases. Figure 7 shows the transient

plots of ANSYS beyond 60 seconds, Hotspot up to 17 seconds and the mathematical regression for Hotspot beyond 60 seconds for cases 1, 2 and 3.

TABLE VI
MICROPROCESSOR CHIP DIE STEADY-STATE TEMPERATURES

	Hotspot		ANSYS	
	Max Temp [°C]	*Min Temp [°C]*	*Max Temp [°C]*	*Min Temp [°C]*
Case 1	85.93	66.87	86.29	67.55
Case 2	76.80	58.81	77.93	56.82
Case 3	68.52	60.53	68.17	62.73

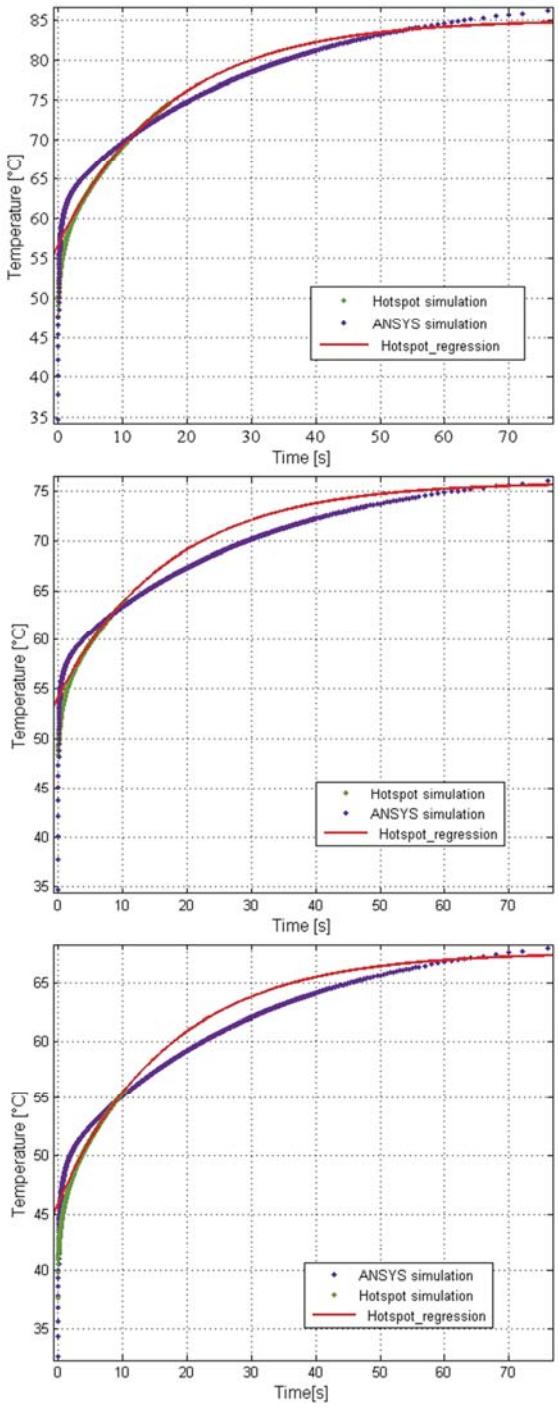

Fig. 7. Simulation results of transient analysis for case 1 (top), case 2 (middle) and case 3 (bottom)

As it can be seen, the maximum temperature values get very close to the ones predicted in steady-state. However, the temperatures slightly differ between FEM and the compact model during the time before steady-state. This difference is similar for all three cases: ANSYS predicts a faster temperature rise during the first seconds (initial heating of the package) whereas Hotspot predicts a much slower temperature rise during that time. However, after fast initial heating, the temperature rise predicted by FEM model slows down considerably, while that predicted by compact model continues to rise at roughly the same rate. As a result, compact-model temperatures surpass the values given by ANSYS in the middle time region. However, simulations showed that steady state values are reached almost after the same time and both models agree on the steady-state temperature. If we consider the FEM model as a reference, the obtained results seem to indicate that Hotspot compact model correctly models thermal resistances, but overestimates the thermal capacitance of the chip layers while underestimating the convection capacitance.

VI. CONCLUSIONS

In this paper, two thermal modelling approaches were compared using as an example a high-performance modern multi-core processor. Steady-state analysis for three different power inputs has shown that the RC network model gives comparable results to the FEM analysis, especially when determining the maximal temperature is concerned. Additionally, using such a model can reduce the simulation time by the order of magnitude. However, it must be also emphasized that the RC model overestimates the temperature gradients within the chip by several degrees. A probable reason for such results is that the lateral heat transfer between two blocks lying on different layers on the edge of the chip is not accurately modelled in the compact model. Additionally, transient analysis showed that compact model does not accurately describe the heat transfer inside the processor package. In result, the heating profiles are not consistent and typically the RC model underestimates the temperature during the initial and final phase of heating and overestimate in the middle.

ACKNOWLEDGMENT

This research was supported by the grant of Polish National Center of Science No. N515 509140.

REFERENCES

[1] International Technology Rodmap for Semiconductors (ITRS), 2009
[2] ANSYS® Workbench™ 14, available at: http://www.ansys.com
[3] COMSOL Multiphysics®, available at: http://www.comsol.com
[4] M. Janicki, G. De Mey, and A. Napieralski, “Thermal analysis of layered electronic circuits with Green’s functions”, Microelectronics Journal, Vol. 38, pp. 177-184, 2007.
[5] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, D. Tarjan, and K. Sankaranarayanan, "Temperature-Aware Microarchitecture." In Proceedings of the 30th International Symposium on Computer Architecture, June 2003
[6] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits”. Proceedings of the IEEE, 91(2):305--327, February 2003.

[7] M. Janicki, P. Zając, M. Szermer, A. Napieralski "Influence of Scaling on IC Temperature in FinFET Microprocessor Technologies" 20th International Conference Mixed Design of Integrated Circuits and Systems, Gdynia, Poland, 20-22 June 2013

[8] P. Zajac, M. Szermer, M. Janicki, C. Maj, P. Pietrzak, A. Napieralski, "Analysis of the effectiveness of core swapping in modern multicore processors," 19th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC, pp.385-388, 25-27 Sept. 2013

[9] A. Krum. Thermal management. In F . Kreith, editor, "The CRC handbook of thermal engineering", pages 2.1–2.92. CRC Press, Boca Raton, FL, 2000

[10] Guoping Xu, "Thermal Modeling Of Multi-Core Processors", Thermal and Thermomechanical Phenomena in Electronics Systems, ITHERM The Tenth Intersociety Conference on, San Diego, CA, USA, pp. 100-96, May 30-June 2, 2006

[11] K. Skadron, Mn R. Stan et al., „Temperature-Aware Microarchitecture: Extended Discussion and Results", University of Virginia, Departement of Computer Science, Technical report CS- 2003-08, April 2003

Melvin Galicia was born in Guatemala city, Guatemala, in 1984. He received the MSc degree from National Sun Yat-sen University, Republic of China Taiwan, in 2011. From 2013 he attends doctoral studies at Lodz University of Technology (LUT), Poland, in microelectronics discipline. His research focuses on modelling and simulation of high performance multicore microprocessors for the investigation of thermal problems and temperature reduction.

Piotr Zając received his diploma in electronics from the Faculty of Electrical, Electronic, Computer and Control Engineering, Technical University of Lodz, Poland (2005) and his PhD from the Department of Electrical and Computer Engineering, National Institute of Applied Sciences of Toulouse, France (2008). He is an Assistant Professor at the Department of Microelectronics and Computer Science at the Technical University of Lodz. His research interests include processor architecture and design, fault tolerance in multi-core systems and thermal challenges in modern ICs. He has published more than 30 papers in peer reviewed transactions, journals and conference proceedings.

Cezary Maj was born in Łódź, Poland, in 1982. He received the MSc degree in electronics from Lodz University of Technology (LUT), Poland, in 2005 and PhD degree in microelectronics from both LUT and Institut National des Sciences Appliquées de Toulouse, France. He specializes in microelectronics and his research focuses on modelling, fabrication and characterization of microsystems. He took part in a few project connected with development of sensors for various applications. Recently, he is involved in MEMS modelling and investigation of thermal problems in current multicore processors.

Michał Szermer was born in Łódź, Poland, in 1973. He received the MSc and PhD degrees from Lodz University of Technology, Poland, in 1998 and 2004, respectively. His research focuses on the integrated circuits design with special consideration of mixed-signal circuits. He took part in many projects connected with ASICs design. Recently, he is involved in MEMS modelling and design as a coordinator of the EduMEMS project in the frame of 7th FP of EU who focused on development of new MEMS structures for educational purposes.

Andrzej Napieralski received the MSc and PhD degrees from the Technical University of Lodz – TUL (Poland) in 1973 and 1977, respectively, and a DSc degree in Electronics from the Warsaw University of Technology, Poland and in Microelectronics from the Université de Paul Sabatier (France) in 1989. Since 1996 he has been Director of the Department of Microelectronics and Computer Science. Between 2002 and 2008 he was Vice-President of TUL. He is an author or co-author of over 960 publications and editor of 19 conference proceedings and 12 scientific Journals. He has supervised 45 PhD theses; six of them received the Prime Minister of Poland prize. In 2008 he received the Degree of Honorary Doctor of Yaroslaw the Wise Novgorod State University (Russia).