

Architecture Level Thermal Modeling for Multi-Core Systems using Subspace System Method *

Thom Jefferson A. Eguia, Riuqing Shen, Sheldon X.-D. Tan, Eduardo H. Pacheco, Murli Tirumala

Abstract – This paper proposes a new architecture-level thermal modeling method to address the emerging thermal related analysis and optimization problem for high-performance multi-core microprocessor design. The new approach, called *Thermsid*, builds the thermal behavioral models from the measured or simulated thermal and power information at the architecture level. *Thermsid* builds the thermal models by first generating a Hankel matrix of Markov parameters, from which state matrices are obtained through minimum square optimization. Compared to existing behavioral thermal modeling algorithms, the proposed method is much more general as it does not require step temperature responses, leading to greater flexibility during the modeling process. Experimental results on a real quad-core microprocessor show that *Thermsid* provides accurate thermal behavioral models comparable existing approaches.

Index Terms – Thermal analysis, architecture thermal modeling, multicore processor

I. INTRODUCTION

As VLSI technology is scaled into the nanometer region, the power density of high-performance microprocessors increases drastically. The exponential power density increase will, in turn, lead to a rapid rise in average chip temperature [3]. Higher temperatures have significant adverse effects on chip packaging cost, performance, and reliability. Excessive on-chip temperature leads to slower transistor speed owing to reduced carrier mobility, more leakage power consumption as leakage currents grow exponentially with temperature, higher interconnect resistance, and reduced reliability [5, 4].

One way of mitigating this problem is by placing multiple cores into a single multi-core CPU [8, 1, 2]. In this way, one can simply increase the total throughput via task-level parallel computation, thus lowering voltage and frequency to meet thermal constraints. In this design, thermal effects are influenced by the placement of cores and shared caches. Therefore, it is very important to consider temperature during the floorplanning and architecture design of multi-core microprocessors.

Existing work on HotSpot [6, 10] attempts to solve this problem by generating the architectural thermal model in a bottom-up manner based on the internal structure/architecture of the microprocessor. However, bottom-up compact models may suffer from accuracy loss, and have to be calibrated

with hardware if more accurate models are required. Also, the generated models may not easily accommodate changing parameters such as thermal conductivity, thermal conditions (ambient temperatures) and packaging configurations when producing parameterized thermal models [11, 12]. Recently, a top-down behavioral architecture level thermal modeling method, *ThermPOF*, was proposed [7], where temperature impulse responses are used to build the thermal models through the matrix pencil method.

In this paper, we propose a new thermal behavioral modeling approach for fast temperature estimation at the architecture level for multicore microprocessors. Similar to *ThermPOF* [7], the new approach builds the system matrices from measured or simulated thermal and power information. However, the proposed method uses a more general subspace system identification method to build the model, which eliminates the need for step temperature responses. The new method, called *Thermsid*, first generates a Hankel matrix of Markov parameters from the measured input and output data via subspace projection and reduction. Then, the discrete state matrices are obtained through minimum square optimization. Experimental results on a real multicore microprocessor show that *Thermsid* can provide thermal behavioral models that closely match the measured data, with accuracy similar to *ThermPOF*. Furthermore, while not considered in this paper, the system identification technique is immune to system noise.

II. ARCHITECTURE-LEVEL THERMAL MODELING PROBLEM

We first present the new thermal behavioral modeling problem. In general, we want to build a behavioral model which is excited by the power input, and produces the temperature outputs in specific locations at the architectural level of the multi-core microprocessor. Our behavioral models are created and calibrated with the measured or simulated temperature and power information from the chip.

Since the given temperature data are transient and changing over time, we need to capture the transient behavior of the temperature. This can be attained by building an impulse response function between temperature and power in the time domain.

In this paper, we study a quad-core microprocessor architecture from our industry partner, Intel Corporation, to validate the new thermal modeling method. The architecture of the multi-core microprocessor is shown in Fig. 1, where there are four CPU cores (die 0 to die 3) and one shared cache core (die 4). TIM stands for thermal interface material. The temperature of each is measured on the bottom face of the center of each die. We can abstract this quad-core CPU into a linear system with 5 inputs and 5 outputs as shown in Fig. 2 (the input p_i and output port t_i will be shared as shown later). The inputs are the power

*Thom A. Eguia and Sheldon Tan are with Department of Electrical Engineering University of California, Riverside, CA 92521, USA. Eduardo Pacheco and Murli Tirumala are with Intel Corporation, 2200 Mission College Blvd, Santa Clara, CA 95052. This work is supported in part by NSF CAREER Award under grant CCF-0448534, NSF Grant CCF-0902885, UC Micro Program #07-101 and #08-11 via Intel Corp and in part by the visiting scholar fund of ASIC & System State-Key-Lab of Fudan University.

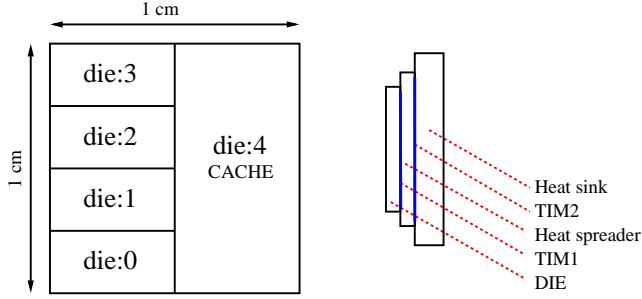


Figure 1: The quad-core architecture.

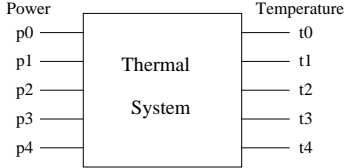


Figure 2: The abstracted model system.

traces of all the cores, while the outputs are the measured temperatures.

Such a system can be described as a discrete linear time invariant system with state space model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k, \end{aligned} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is a stable matrix, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$, and $D \in \mathbb{R}^{l \times m}$. The input vectors $u_k \in \mathbb{R}^{m \times 1}$ and output vectors $y_k \in \mathbb{R}^{l \times 1}$ the measured power input traces and temperature responses, respectively.

Given input u_k and output y_k , the problem at hand is how to generate state matrices A , B , C , and D , where D is typically considered a matrix of zeros. In the following section, the subspace system method for finding these state matrices are described.

III. THERMAL MODELING BY SUBSPACE SYSTEM METHOD: THERMSID

To determine the state matrices of a core's thermal system, we apply a recently proposed subspace system identification method called Numerical Algorithm for Subspace State Space System Identification (N4SID) [9]. Based on N4SID, we develop the thermal modeling algorithm, *ThermSID*, which first generates a Hankel matrix of Markov parameters from the measured input and output data via subspace projection and reduction. State matrices are then generated via the minimum square optimization. We will first introduce some concepts used in N4SID, followed by the solution algorithm.

We suppose a core's thermal system is linear time invariant (LTI) with noise, where the state space model is described in (1). For the system in (1), $h_k = CA^{k-1}B$ for $k \geq 1$ are called Markov parameters, which are the values of discrete-time impulse responses. The Hankel matrix of the Markov parameters can be defined as

$$\mathcal{H}_{i,j} = \begin{bmatrix} h_i & h_{i+1} & \dots & h_{i+j} \\ h_{i+1} & h_{i+2} & \dots & h_{i+j+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{i+j} & h_{i+j+1} & \dots & h_{i+2j} \end{bmatrix}. \quad (2)$$

It can be proven that

$$\mathcal{H}_{1,N-1} = \mathcal{M}_O(N)\mathcal{M}_C(N), \quad (3)$$

where \mathcal{M}_C and \mathcal{M}_O are the extended controllability and observability matrices, which have the following form

$$\mathcal{M}_C(j) = \begin{bmatrix} B & AB & \dots & A^{j-1}B \end{bmatrix}, \mathcal{M}_O(j) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{j-1} \end{bmatrix}. \quad (4)$$

Considering s points of history and r points of future data, there are three main steps for the subspace algorithm:

- 1) Develop an estimate \hat{x}_k for the state x_k ,

$$y_{future}(k) = \mathcal{M}_O(r)\hat{x}_k + S_{future}u_{future}(k) \quad (5)$$

where

$$y_{future}(k) = [y_k, y_{k+1}, \dots, y_{k+r-1}]^T, \quad (6)$$

$$u_{future}(k) = [u_k, u_{k+1}, \dots, u_{k+r-1}]^T, \quad (7)$$

$$S_{future} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ CB & 0 & \dots & 0 \\ \vdots & & \ddots & 0 \\ CA^{r-2}B & CA^{r-3}B & \dots & 0 \end{bmatrix}. \quad (8)$$

- 2) Use \hat{x}_k in (5) to estimate $\hat{y}_{future}(k)$.

- 3) Form block Hankel matrices for data and predicted responses, and select parameters to

$$\min \|Y_{future}^{data} - \hat{Y}_{future}^{pred}\|_F^2, \quad (9)$$

where $\|A\|_F^2 = \text{Trace}(A * A)$.

Here we choose the best linear mean-square for x_k ,

$$\hat{x}_k = K_1 y_{past}(k-s) + K_2 u_{past}(k-s) + K_3 u_{future}(k), \quad (10)$$

where

$$y_{past}(k-s) = [y_{k-s}, y_{k-s+1}, \dots, y_{k-1}]^T,$$

$$u_{past}(k-s) = [u_{k-s}, u_{k-s+1}, \dots, u_{k-1}]^T. \quad (11)$$

From (5) and (10), $\hat{y}_{future}(k)$ should be a linear combination of $y_{past}(k-s)$, $u_{past}(k-s)$ and $u_{future}(k)$. Rewrite it in matrix form as follows,

$$\hat{Y}_{future}^{pred} = L_1 Y_{past} + L_2 U_{past} + L_3 U_{future}, \quad (12)$$

where

$$\hat{Y}_{future}^{pred} = [\hat{Y}_{future}(s+1), \hat{Y}_{future}(s+2), \dots, \hat{Y}_{future}(N-r+1)],$$

$$Y_{past} = [y_{past}(1), y_{past}(2), \dots, y_{past}(N-r-s+1)],$$

$$U_{past} = [u_{past}(1), u_{past}(2), \dots, u_{past}(N-r-s+1)],$$

$$U_{future} = [u_{future}(s+1), u_{future}(s+2), \dots, u_{future}(N-r+1)].$$

It can be proven that $[L_1 \ L_2] = \mathcal{H}$, where L_i is chosen to optimize (9).

$$W = [Y_{past}^T \ U_{past}^T \ Y_{future}^T]^T, \quad (13)$$

then the solution of the optimization problem in (9) can be obtained by the LQ-factorization of W ,

$$W = \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ L_1 & L_2 & * \end{bmatrix} [Q_1 \ Q_2 \ Q_3]. \quad (14)$$

We perform SVD on $\mathcal{H} = [L_1 L_2] = U \Sigma V^*$, where Σ is used to identify the system order n . Afterwards, we can define the first n columns of U and V as U_1 and V_1 , respectively,

and the diagonal matrix with the first n singular values in Σ as Σ_1 . Then, we define

$$\mathcal{M}_O = U_1 \Sigma_1^{1/2}, \mathcal{M}_C = \Sigma_1^{1/2} V_1^*. \quad (15)$$

System matrix C is the first n rows of \mathcal{M}_O , while B is retrieved from the first n columns of \mathcal{M}_C . The matrix A can be obtained by

$$A = (\mathcal{M}_O^\dagger)^\dagger \mathcal{M}_O^\dagger, \quad (16)$$

where

$$\mathcal{M}_O^\dagger = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{M-1} \end{bmatrix}, \quad \mathcal{M}_C^\dagger = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{M-2} \end{bmatrix}. \quad (17)$$

Compared to a previously proposed thermal modeling algorithm, *ThermPOF* [7], thermal modeling using *ThermSID* is more flexible and general. Since step responses are used during the training process in *ThermPOF*, a change to log-scale is necessary to better capture step response behavior. On the contrary, random power input traces and the resulting temperature responses of the desired system can be used directly as the input and output data sets for *ThermSID*, thus avoiding a log scale change for time during the training process. Furthermore, while not considered in this paper, *ThermSID* is capable of generating state space models subject to system noise.

IV. EXPERIMENTAL RESULTS

The proposed *ThermSID* algorithm has been implemented in MATLAB 7.6 and tested on a quad-core microprocessor architecture from industry partner Intel Corp. We used the first half of the temperature response data as the training data set, while the latter half was used to verify the generated models. The results are then compared to the previously proposed thermal modeling algorithm, *ThermPOF*. In order to determine the accuracy of *ThermSID*, we observed the maximum peak errors, as well the mean error and standard deviation.

The experimental data contains each core's temperature, which is measured directly from the center of the dies. Initially, all cores are in zero state, and have an initial ambient temperature of 35 °C.

We verify the correctness of our model based on thermal benchmarks provided by Intel. The temperature on every core is driven to an initial state by a power of 10W on each of the cores for 1000s. We then apply a number of random power input traces from 0 to 1 second, with a time step of 0.1s, as shown in Fig. 3, where the step power is 20W for all cores.

The temperature response generated by the random input is partitioned. The first section, which is the temperature response from 0 to 0.6 seconds, is used as the training data set. The random power input traces and resulting temperature responses of cores 0 through 3 and the cache served as the inputs and outputs, respectively, for the *ThermSID* algorithm. The results for core 0 is shown in Fig. 4, where the solid line represents the original data while the marked like represents the output produced by the *ThermSID* model. From this figure, we can see that the model produces a highly accurate match of the training data set.

The latter section of the thermal data, which spans from 0.6 to 1 seconds, is used to verify the accuracy of the generated models. Fig. 5 and Fig. 6 show the results of core 2 and the cache, respectively, with the solid line representing the original data, the dash line representing the output generated by *ThermPOF*, and the marked line representing the

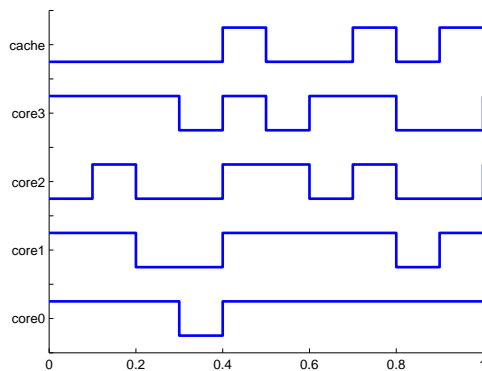


Figure 3: Random power input traces for cores 0 through 3 and cache

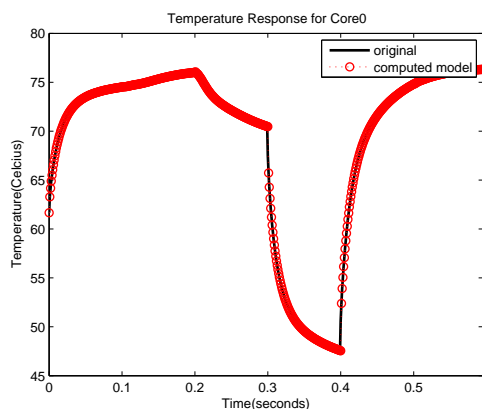


Figure 4: Temperature response of training data and computed model for Core 0

output generated by *ThermSID*. The results show that the models produced by *ThermSID* are just as accurate, and at times, more accurate than those produced by *ThermPOF* for the same order models. In this test, both algorithms produced models of order $M = 50$.

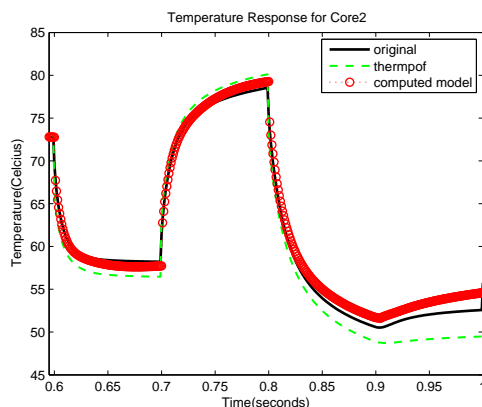


Figure 5: Thermal simulation results on Core 1

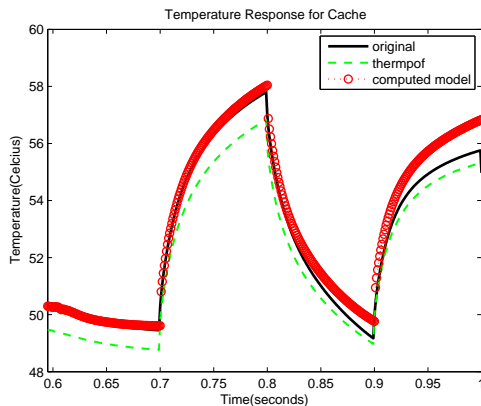
The maximum peak errors between the original and calculated temperature responses are shown in Table 1. For *ThermSID*, the max peak error is typically below 1 percent. Results show that *ThermSID* produces peak values that are

Table 1: Maximal error peaks (M = 50)

core	measured	thmsid peak	thmpof peak	thmsid diff	thmpof diff	thmsid %	thmpof %
Core 0	77.27	78.11	77.34	0.84	0.07	1.09	0.09
Core 1	78.86	79.61	78.85	0.75	0.12	0.95	0.13
Core 2	78.55	79.25	80.12	0.73	1.57	0.93	2.00
Core 3	76.48	77.09	77.34	0.61	1.03	0.80	1.34
Cache	57.80	58.04	56.83	0.24	0.99	0.42	1.71

Table 2: Statistics of errors between measured and computed temperatures

core	Thermsid max error	ThermPOF max error	Thermsid mean error	ThermPOF mean error	Thermsid error std	ThermPOF error std
Core 0	1.56	1.32	0.72	0.55	0.34	0.20
Core 1	3.91	1.70	0.81	0.70	0.48	0.49
Core 2	3.90	3.24	0.86	1.83	0.55	0.60
Core 3	3.78	3.69	0.80	1.25	0.38	0.44
Cache	1.06	1.14	0.36	0.65	0.30	0.24

**Figure 6: Thermal simulation results on the Cache core**

just as accurate as those of *ThermPOF*.

Error statistics between the original and calculated temperature responses are shown in Table 2. The maximum error among the respective data sets, mean, and standard deviation are given. For *Thermsid*, the mean error is below 1 degree, with standard deviation of less than .5 degrees. Results show a slightly better performance by *Thermsid* compared to *ThermPOF*.

Overall, the experimental results show that *Thermsid* provides accurate results comparable to that of *ThermPOF*. Furthermore, it is a much more general algorithm as it avoids any change of scale in the initial training process.

V. CONCLUSION

In this paper, we proposed a new architecture-level thermal modeling method, called *Thermsid*, which builds the thermal behavioral models from the measured or simulated thermal and power information. Based on the N4SID algorithm, *Thermsid* builds the thermal models by first generating a Hankel matrix of Markov parameters, from which state matrices are obtained through minimum square optimization. Compared to existing behavioral thermal modeling algorithms, the proposed method is much more general in that it does not require step temperature responses, leading to greater flexibility during the modeling process. Experimental results on a real quad-core microprocessor show that *Thermsid* provides accurate thermal behavioral models comparable to *ThermPOF*.

VI. REFERENCES

- [1] “Intel multi-core processors (white paper),” 2006, <http://www.intel.com/multi-core>.
- [2] “Multi-core processors - the next evolution in computing (white paper),” 2006, <http://multicore.amd.com>.
- [3] “International technology roadmap for semiconductors(itrs) 2007 edition,” 2007, <http://public.itrs.net>.
- [4] D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” in *Proc. of Intl. Symp. on High-Performance Comp. Architecture*, 2001, pp. 171–182.
- [5] S. Gunther, F. Binns, D. Carmean, and J. Hall, “Managing the impact of increasing microprocessor power consumption,” in *Intel Technology Journal*, First Quarter 2001.
- [6] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, “Compact thermal modeling for temperature-aware design,” in *Proc. Design Automation Conf. (DAC)*, 2004, pp. 878–883.
- [7] D. Li, S. X.-D. Tan, and M. Tirumala, “Architecture-level thermal behavioral characterization for multi-core microprocessors,” in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, 2008, pp. 456–461.
- [8] Y. Li, B. C. Lee, D. Brooks, Z. Hu, and K. Skadron, “CMP design space exploration subject to physical constraints,” *Proc. IEEE Int. Symp. on High Performance Computer Architecture (HPCA)*, pp. 15–26, 2006.
- [9] P. V. Overschee and B. D. Moor, “N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems,” *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.
- [10] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature aware microarchitecture,” in *Proc. IEEE International Symposium on Computer Architecture (ISCA)*, 2003, pp. 2–13.
- [11] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, “A systematic method for functional unit power estimation in microprocessors,” in *Proc. Design Automation Conf. (DAC)*, June 2006, pp. 554–557.
- [12] —, “Efficient power modeling and software thermal sensing for runtime temperature monitoring,” *ACM Trans. on Design Automation of Electronics Systems*, vol. 12, no. 3, pp. 1–29, 2007.