Architecture-level Thermal Behavioral Models For Quad-Core Microprocessors *

Duo Li
Dept of Electrical Engineering
University of California
Riverside, CA 92521
dli@ee.ucr.edu

Sheldon X.-D. Tan
Dept of Electrical Engineering
University of California
Riverside, CA 92521
stan@ee.ucr.edu

Murli Tirumala
Intel Corporation
2200 Mission College Blvd
Santa Clara, CA 95052
murli.tirumala@intel.com

ABSTRACT

In this paper, we study a new architecture level thermal modeling problem from behavioral modeling perspective to address the emerging thermal related analysis and optimization problems for high-performance quad-core microprocessor designs. We propose a new approach to build the thermal behavioral models by using transfer function matrix from the measured thermal and power information at the architecture level. The new method builds behavioral thermal model using generalized pencil-of-function (GPOF) method, which was developed in the communication community to build the rational modeling from the measured data of real-time systems. To effectively model transient temperature changes, we propose two new schemes to improve the GPOF. First we apply logarithmic-scale sampling instead of traditional linear sampling to better capture the temperature changing characteristics. Second, we modify the extracted thermal impulse response such that the extracted poles from GPOF are guaranteed to be stable without accuracy loss. Experimental results on a practical quad-core microprocessor show that generated thermal behavioral models match the measured data very well.

1. INTRODUCTION

As CMOS technology is scaled into the nanometer region, the power density of high-performance microprocessors will increase drastically. The exponential power density increase will in turn lead to average chip temperature to raise rapidly [2]. Higher temperature has significant adverse impacts on chip performance and reliability. Excessive on-chip temperature leads to slower transistor speed, more leakage power consumption, higher interconnect resistance, and reduced reliability.

One way to mitigate the high temperature problem to put multiple CPU or cores into one single chip [9, 1, 3]. In this way, one can simply increase the total throughput by parallel computation, and have lower voltage and frequency to meet thermal constraints. But the thermal effects are influenced by the placement of cores and caches. So it is very important to consider the temperature during the floorplanning and architecture design of multi-core microprocessor.

The estimated temperature at the architecture level can then be used to perform power, performance, and reliability analysis, together with floorplanning and packaging design [12]. As a result, design decision is guided by temperature and design is optimized theoretically without potential thermal problems. To facilitate this temperature-aware architecture design, it is important to have accurate and fast thermal estimation at the architecture level. Both architecture and CAD tool community are currently lacking reliable and practical tools for thermal architecture modeling. Existing work on the HotSpot project [8, 12] tried to resolve this problem by generating the architecture thermal model in a bottom-up way based on the floorplanning of the function units. But this method is difficult to set up for new architecture with different thermal and packaging configurations. Also the resulting model work for only single CPU architecture and the accuracy may not sufficient as many approximations are made.

This paper, we propose a new thermal behavioral modeling approach for fast temperature estimation at the quad-core thermal architecture level at early design stage. The new approach builds the transfer function matrix from the measured architecture level thermal and power information. It first builds behavioral thermal model using generalized pencil-of-function (GPOF) method [6, 7, 11], which was developed in the communication community to build the rational modeling from the measured data of real-time and electromagnetism systems. However, direct use of GPOF will not generate stable useful thermal models. Based on the characteristics of transient temperature behaviors, we make two new improvements over the traditional GPOF: First we apply logarithmic-scale sampling instead of traditional linear sampling to better capture the temperatures change over the time. Second, we modify the extracted thermal impulse response such that the extracted poles from GPOF are guaranteed to be stable without accuracy loss. Experimental results on a practical quad-core microprocessor show that the generated thermal behavioral models can be built very efficient and the resulting model match well the measured temperature for non-training data.

The rest of this paper is organized as the follows: Section 2 presents thermal modeling problem we try to solve. Section 3 reviews a generalized pencil-of-function (GPOF) method for extracting the poles and residues from the transient response of a real-time system and electromagnetism. Section 4 presents our new thermal behavioral modeling approach based on the GPOF. Section 5 presents the experimental results and Section 6 concludes this paper.

2. ARCHITECTURE-LEVEL THERMAL MODELING PROBLEM

We first present the new thermal behavioral modeling problem. Basically we want to build the behavioral model, which is excited by the power input and product the temperature outputs for the specific locations in the floorplan of the quad-core microprocessor. Our behavioral models are created and calibrated with the measured temperature and

*This work is funded by NSF CAREER Award under grant CCF-0448534, NSF Award under grant CCF-0541456 and UC Micro via Intel Corporation.

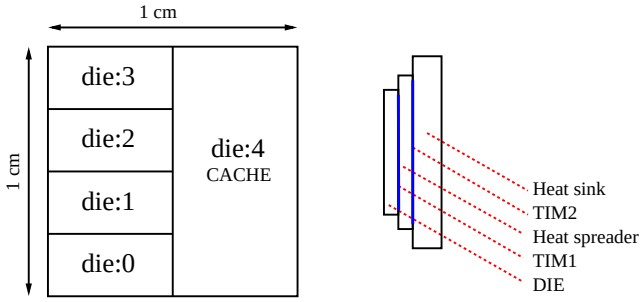


Figure 1: Quad-core architecture

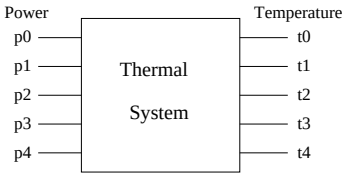


Figure 2: Abstracted system

power information from the real chips. The benefit of such behavioral thermal models is that it can easily built for many different architecture with different thermal conditions and thermal parameters (thermal conductivity, cooling configuration).

Since the given the temperature data are transient and changing over time, we need to capture the transient behavior of the temperature, which can be achieved by building the impulse functions between temperature and power in the time domain.

In this paper, we specifically look at quad-core microprocessor architecture from our industry partner to validate new thermal modeling method. The architecture of this multi-core microprocessor is shown in Fig. 1, where there are four CPU cores (die 0 to die 3) and one cache core (die 4). The temperature of each die is reported on the die bottom face in 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 inputs are the power traces of all the cores, and the outputs are the temperature of them, respectively.

Such system can be described by the impulse-response function matrix $\mathbf{H}$

$$\mathbf{H(t)} = \begin{bmatrix} h_{00}(t) & h_{01}(t) & h_{02}(t) & h_{03}(t) & h_{04}(t) \\ h_{10}(t) & h_{11}(t) & h_{12}(t) & h_{13}(t) & h_{14}(t) \\ h_{20}(t) & h_{21}(t) & h_{22}(t) & h_{23}(t) & h_{24}(t) \\ h_{30}(t) & h_{31}(t) & h_{32}(t) & h_{33}(t) & h_{34}(t) \\ h_{40}(t) & h_{41}(t) & h_{42}(t) & h_{43}(t) & h_{44}(t) \end{bmatrix} \quad (1)$$

where h_{ij} are the impulse response function for output port i due to input port j.

Given a power input vector for each core $\mathbf{u}(t)$, the transient temperature can be then computed

$$\mathbf{y}(t) = \int_0^t \mathbf{H}(t-\tau)\mathbf{u}(\tau)d\tau \quad (2)$$

Equation (2) can be written in frequency domain as in (3).

$$\mathbf{y}(s) = \mathbf{H}(s)\mathbf{u}(s) \quad (3)$$

where $\mathbf{y}(s)$, $\mathbf{u}(s)$ and $\mathbf{H}(s)$ are the Laplace transform of $\mathbf{y}(t)$, $\mathbf{u}(t)$ and $\mathbf{H}(t)$, respectively. $\mathbf{H}(s)$ is called the transfer-function matrix of the system where each $h_{ij}(s)$ can be represented as the partial fraction form or the pole-residue form (4).

$$h_{ij}(s) = \sum_{k=1}^{n} \frac{r_k}{s - p_k} \quad (4)$$

where $h_{ij}(s)$ is the transfer function between the jth input terminal and the ith output terminal; p_k and r_k are the kth pole and residue. Once transfer functions are computed, the transient responses can be easily computed.

The remaining important problem is to find the poles and residues for each transfer function h_{ij} from the measured thermal and power information. It turns out the generalized pencil-of-function can be used for this propose. But we cannot simply apply GPOF method as we show in the Section 4. In the following section, we will briefly review the GPOF method before we present our improvements.

3. REVIEW OF GENERALIZED PENCIL-OF-FUNCTION METHOD

Generalized pencil-of-function (GPOF) method can be used to extract the poles and residues from the transient response of a real-time system and electromagnetism [6, 7, 11]. Specifically, GPOF can work for such a system that can be expressed in sum of complex exponentials:

$$y_k = \sum_{i=1,M} r_i e^{(p_i \Delta t k)} \quad (5)$$

where, k = 0, 1, ..., N-1, is the number of sampled points, r_i is the complex residues, p_i are the complex poles, and Δt is the sampling interval. M is the number poles we used to build the transfer function. Let's define

$$z_i = e^{(p_i \Delta t k)} \quad (6)$$

which become the poles in Z-plane. For real value y_k, both r_i and p_i should be in complex conjugate pairs. Let's define the new vector of node temperatures (in our problem) as $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_L$ where,

$$\mathbf{y}_i = [y_i, y_{i+1}, ..., y_{i+N-L-1}]^T \quad (7)$$

where L can be viewed as sampling window size. Based on these vectors, we can define the matrices Y_1 and Y_2 as

$$Y_1 = [\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{L-1}] \quad (8)$$

$$Y_2 = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L] \quad (9)$$

Then one can obtain the following relationship among the Y_1, Y_2 and the pole and residue vectors Z_0 and R based on the structure of Y_1, Y_2:

$$Y_1 = Z_1 R Z_2 \quad (10)$$

$$Y_2 = Z_1 R Z_0 Z_2 \quad (11)$$

where

$$Z_1 = \begin{bmatrix} 1 & 1 & ... & 1 \\ z_1 & z_2 & ... & z_M \\ \vdots & \vdots & ... & \vdots \\ Z_1^{N-L-1} & Z_2^{N-L-1} & ... & Z_M^{N-L-1} \end{bmatrix} \quad (12)$$

$$Z_2 = \begin{bmatrix} 1 & z_1 & ... & z_1^{L-1} \\ & & ... & \\ \vdots & \vdots & ... & \vdots \\ 1 & z_M & ... & Z_M^{L-1} \end{bmatrix} \quad (13)$$

$$Z_0 = diag[z_1, z_2, ..., z_M] \quad (14)$$

$$R = diag[r_1, r_2, ..., r_M] \quad (15)$$

So the problem we need to solve is to find the pole and residue vector Z_0 and R efficiently. It turns out that this can be easily computed by observing that

$$\begin{aligned} Y_1^+ Y_2 &= Z_2^+ R^{-1} Z_1^+ Z_1 R Z_0 Z_2 \\ & Z_2^+ Z_0 Z_2 \end{aligned} \quad (16)$$

Hence, the poles is the eigenvalues of $Y_1^+ Y_2$, where $+$ indicate the (Moore-Penrose) pseudo-inverse, as Y_1 is not a square matrix. As a result, one can obtain the Z_0 by using

$$Z = D^{-1} U^H Y_2 V \quad (17)$$

where Z is a $M \times M$ matrix and V and U comes from the singular value decomposition (SVD) of Y_1:

$$Y_1 = UDV^H \quad (18)$$

After the Z is computed, we can obtain the pole vector Z_0 by performing the eigen-decomposition of Z, $Z_0 = eig(Z)$, where $eig(X)$ is to obtain the eigenvalue vector from matrix X. Once Z_0 is obtained, we can compute the residue vector R by using either (10) or (11).

For GPOF method, it allows $M \leq L \leq N - M$, which means that we can allow the different window size and pole numbers. Typically, choosing $L = N/2$ can yield better results.

4. NEW ARCHITECTURE-LEVEL THERMAL BEHAVIORAL MODELING METHOD

In this section, we present our new thermal behavioral modeling approach based on the GPOF method mentioned in the previous section.

For a linear time-invariant system, the sum of complex exponential form shown in (5) essentially is the impulse response in the s-domain. So we need to apply the GPOF method to the thermal impulse responses, which in generally cannot be obtained directly from measurement. Instead, we measure the thermal step responses for each core (center of the core) excited by the some power inputs in the given multi-core microprocessor. Then impulse responses are obtained by performing the numerical differentiation of the step responses.

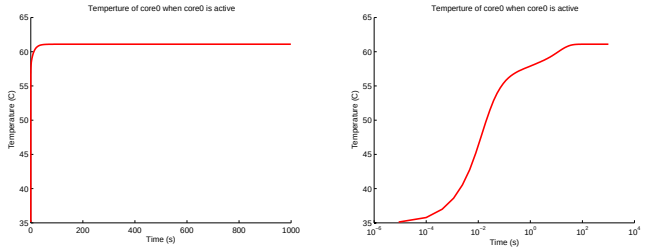
But directly applying the GPOF to the computed thermal impulse responses (from the measured data) may not lead to accurate and stable models as shown below. In the following, we will present two improvement schemes in the new method such that the resulting models are accurate and stable.

4.1 Logarithmic scale sampling for poles and residues extraction

The first problem we face for the thermal modeling is that linear sampling in the traditional GPOF method does not work for our thermal data.

According to GPOF method reviewed in Section 3, we know that matrices Y_1 and Y_2 are constructed from the sampled data and the sampling time interval Δt must be the same. However, how to obtain sample data from the observed temperature curve became a big issue in our CPU temperature simulation, because the step temperature response often goes up drastically in the first few seconds and gradually tends to reach a steady state after a relatively long time.

This can be observed in Fig. 3(a), which is step temperature responses for $core0$ ($die : 0$) when only $core0$ is driven by a step $20W$ power sources beginning at $t = 0$ (which is called *active* in this paper). The environment temperature (initial temperature when no input power at the beginning) is $35^\circ C$. We observe that almost all the temperature increase occurs within the first second, from $35^\circ C$ to $57.9^\circ C$, where $61.1^\circ C$ is the final temperature when $core0$ reaches a steady state.



(a) Linear time scale thermal step response.

(b) Logarithmic time scale thermal step response.

Figure 3: The transient temperature change of core0 when core0 is excited by 20W power input.

Hence we observe that temperature changes can be very rapidly in a very short time and gradually reach a steady state for a long time. This can cause the modeling problem for GPOF method if linear sampling is used. To capture the thermal change information, sampling interval needs to very small, but this will lead to very large number of sampled data (or intervals) due to the long tail of the thermal response reaching the steady state. As a result, we have to use a very big N and consequently very big L, which cause large dimensions of matrices Y_1 and Y_2 in GPOF. As a result, the following matrix operations such as multiplication, inverse or singular value decomposition (SVD) become very expensive.

In this paper, we propose to use logarithmic-scale sampling (log-scale sampling for short) to mitigate this problem. If we use the log-scale sampling for the same temperature responses in Fig. 3(a), we obtain the log-sampled temperature response in Fig. 3(b), which clearly show how the temperature changes over the log-scale time gradually.

After the logarithmic operation of time, the converted time, which is $ln(\text{time})$, will become negative. So we need to offset it to make sure that temperature response always starts at $t = 0$. And the offsetting will not affect GPOF operations. After we obtain the transfer function from GPOF, we need to compute the response in original time scale. We can get the response back by using (19),

$$\mathbf{y}'(t) = \mathbf{y}(ln(t) + t_0) \quad (19)$$

where $\mathbf{y}'(t)$ is the response in normal time scale $\mathbf{y}(t)$ is the response in log-scale; t_0 is the offset for this transfer function.

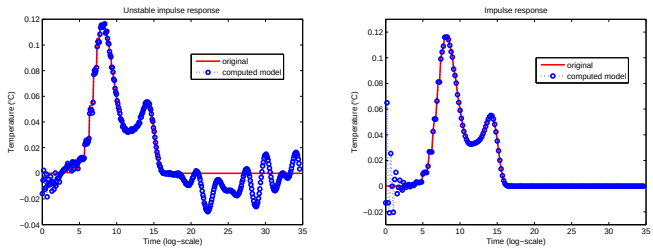
4.2 Stable poles and residues extraction

4.2.1 *Stable pole extraction*

The second problem with the GPOF method is that it will not always generate stable poles for a given impulse response. Actually GPOF model can give a very good matching for a given impulse response for the sampled interval while using positive poles. But outside the sampled interval, the response from the model by GPOF can be unbounded due to the positive poles.

Fig. 4(a) shows the extracted impulse response compared to the original one for one of the cores. For this example, the sampled time interval is from 0 to 1000 seconds. Except for

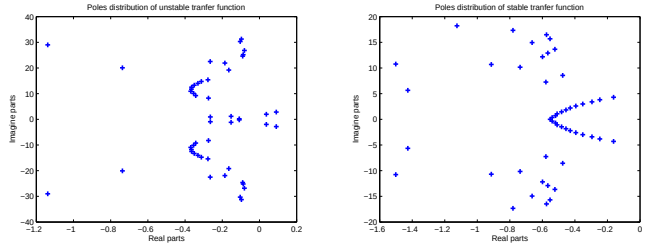
the very beginning (we will address this issue later), it can be seen that the computed model matches very well with the original model from time 0 to the $1000s$(the corresponding $x = 18.55$ in log-scaled x-axis with offset being 11.64). But outside the time interval, if we extend the time scale to 10^{10} seconds, they are significant difference between the two models. The computed models does not look like an impulse responses and will go unbounded actually owning to the positive poles. Fig. 5(a) shows the extracted poles where not all the poles extracted by GPOF are stable (negative real parts).



(a) Extracted impulse response with positive poles

(b) Extracted impulse response with only negative poles

Figure 4: Unstable and stable impulse response for Core0



(a) Extracted poles with positive poles

(b) Extracted poles with only negative poles

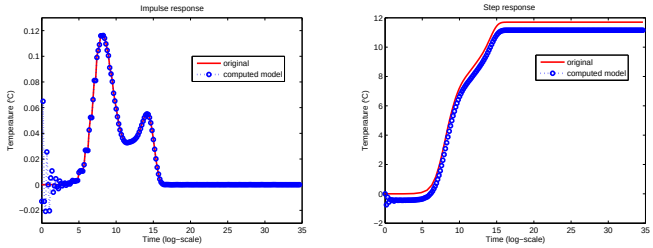
Figure 5: Poles distributions of unstable and stable extracted transfer function

To mitigate this problem, we propose to extend the time interval for zero-response time. For any impulse response, after sufficient time, the response will become zero (or numerically become zero) as the area integration of the impulse curve below is a constant. By sufficiently extending the time interval for zero-response time in a impulse response, we can make all the poles stable. The reason is that if we have positive poles, after sufficient long time, the response will always go non-zero and eventually become unbounded assuming all the poles are different numerically, which is always true practically. If we ensure the zero response for sufficient long time, all the poles must be stable to have the zero responses for very long time as response contributed by those poles will decay to zero.

Using the same example, if we extend the time interval to 10^{10} seconds, which actually does not increase significantly in log-scale, all the extracted poles become stable. Fig. 5(b) shows the extracted poles by extending zero-response time to $10^{10}s$ where all the poles are stable (with negative real part) and Fig. 4(b) shows the extracted stable impulse response. For different problems, we may need to find such a sufficient time period. For all our problem, we find $10^{10}s$ seems such a good sufficient time for our example.

4.2.2 *Stabilizing the starting response*

After the log-scale sampling and numerical differentiation, the obtained impulse response become zero numerically for a short period as temperature changes at the very beginning is very slow. For example, we consider the temperature of $core1$ when only $core0$ is active. Assume that $core0$ is active at $t = 0$, in the first very short time, such as $t = 10^{-4}s$, temperature response of $core1$, due to the delay in thermal transmission, is probably still 0 and it may begin to increase at $t = 10^{-3}s$. Normally we consider the difference $10^{-4}s$ and $10^{-3}s$ as a small value, but in log-scale, this difference is translated to a period of time with zero responses at the beginning. And long zero-response time at beginning may cause the significant discrepancies as shown in Fig. 6(a), although the computed response tends to be accurate after some time period. This means this transfer function we obtained is not accurate enough. Fig. 6(b) shows a step response computed by the transfer function obtained in Fig. 6(a). Obviously, it has noticeable difference compared to the original one.



(a) Impulse response with large errors in the starting time.

(b) The corresponding step response for the inaccurate model. Here the zero temperature means the room temperature.

Figure 6: Impulse and step response computed by inaccurate model with large error in the starting time.

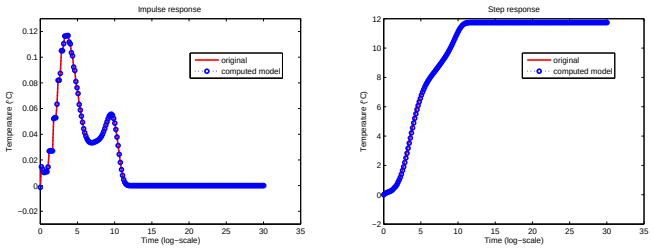
The reason for this problem is that the log-scaled impulse response is different than the typically impulse response from physical RLC circuit in which the response goes to non-zero immediately after $t = 0$. To resolve this problem, we propose to truncate the beginning zero-response time such that responses go to non-zero numerically immediately. This can be achieved by setting threshold temperature to locate the new zero time. During the simulation process, in all the actual time before the new zero time, the response will be set to zero. Fig. 7 shows the impulse and step response computed by accurate model after truncating the beginning zeros.

5. EXPERIMENTAL RESULTS

The proposed new algorithm has been implemented in Matlab 7.0 and tested on the quad-core microprocessor architecture shown in Fig. 1 from our industry partner. We first extracted transfer function matrix of the system through a training data set, which consists of the step responses for each core from other cores. After extracting the transfer functions, we could apply them to compute the thermal responses from any power inputs with any time varying inputs.

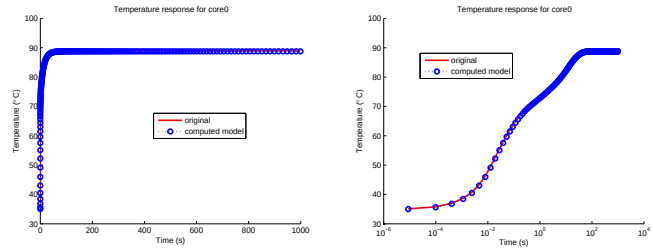
Table 2: Features of the difference between measured and computed temperatures

	Difference ($^\circ C$)			Difference percentage	
	Maximum	Mean	Std. deviation	Maximum	On average
Core0	0.46	0.25	0.08	0.89%	0.32%
Core1	0.27	0.18	0.07	0.42%	0.15%
Core2	0.37	0.16	0.08	0.73%	0.20%
Core3	0.46	0.24	0.08	0.88%	0.31%
Cache	0.31	0.16	0.08	0.51%	0.26%



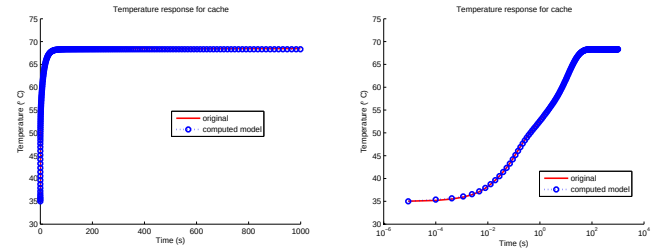
(a) Impulse response. (b) Step response.

Figure 7: Impulse and step response computed by accurate model with both improvements.



(a) Temperature response of core0 in linear scale. (b) Temperature response of core0 in log-scale.

Figure 8: Comparison results of core0's temperature when all cores are active (driven by 20W powers).



(a) Temperature response of cache in linear scale. (b) Temperature response of cache in log-scale.

Figure 9: Comparison results of cache's temperature when all cores are active (driven by 20W powers).

Our experimental data are each core's temperatures measured directly from the center of the dies as we introduced in Section 2, which are provided by our industry partner. At the beginning all the cores are in zero state and have an initial environmental temperature $35^\circ C$. From $t = 0$ each core is excited by a step power input of $20W$ simultaneously. And the temperature of each core is collected from $0s$ to $1000s$.

Table 1: Difference when temperatures achieve the steady state

	Measured Temp. ($^\circ C$)	Computed Temp. ($^\circ C$)	Difference percentage
Core0	88.96	88.78	0.22%
Core1	90.60	90.52	0.08%
Core2	90.04	88.94	0.11%
Core3	88.96	88.78	0.20%
Cache	68.46	68.32	0.20%

Now we verify the correctness of our model based on the given thermal data from our industry partner. For each transfer function, we set an order of 50. This is already enough for our model. In practice, temperature on each core or cache can be computed very fast by our model during any time interval. Because our model is directly based on the transfer function represented by poles and residues instead of state space equations.

We show our comparison results of core0 and cache in Fig. 8 and Fig. 9 under normal linear time scale and log-scale, respectively. The red solid curve represents the measured temperature and the blue curve represents our computed temperature. The simulation runs very fast and costs only few minutes. From these result figures, we can see that our model has very good accuracy. Actually, the temperatures of other cores match well too. We could not show them here due to limit of the maximum pages.

In Table 1 we show the temperatures when all the cores achieve the steady state and the differences percentage. The difference is only around 0.2%. Furthermore, Table 2 shows some statistical features of the differences over all the sampling time points, including the maximums, the means and the standard deviations. Also, the maximum and average percentages are given. From this table we can see the maximum difference is less than $0.5^\circ C$ and 1% and the average difference is less than $0.3^\circ C$ and 0.3% for all the cores.

6. CONCLUSION

In this paper, we proposed a new architecture level thermal behavioral modeling method. The new method, builds the thermal behavioral models by using transfer function matrix from the measured architecture thermal and power information. We applied the generalized pencil-of-function (GPOF) method to construct the impulse response functions from the measured thermal data. To effectively model the transient temperature changes, we have proposed two new schemes to improve the GPOF. First we applied logarithmic-scale sampling instead of traditional linear sampling to better capture the temperature changing characteristics. Sec-

ond, we modified the extracted thermal impulse response such that the extracted poles from GPOF are guaranteed to be stable without accuracy loss. Experimental results on a practical quad-core microprocessor have demonstrated that generated thermal behavioral models match the measured data very well. The proposed method can also be applied to thermal modeling of VLSI circuits at different granularity.

7. REFERENCES

[1] "Intel multi-core processors (white paper)," 2006, http://www.intel.com/multi-core.

[2] "International technology roadmap for semiconductors(itrs) 2005, 2006 update," 2006, http://public.itrs.net.

[3] "Multi-core processors - the next evolution in computing (white paper)," 2006, http://multicore.amd.com.

[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] Y. Hua and T. Sarkar, "Generalized pencil of function method for extracting poles of an em system from its transient responses," *IEEE Trans. on Antennas and Propagation*, vol. 37, no. 2, pp. 229–234, Feb. 1989.

[7] ——, "On svd for estimating generalized eigenvalues of singular matrix pencils in noise," *IEEE Trans. on Signal Processing*, vol. 39, no. 4, pp. 892–900, Apr. 1991.

[8] 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.

[9] 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.

[10] M. Santarina, "Thermal integrity: a must for low-power-ic digital design," in *Electronic Design Network (www.edn.com)*, Sept. 15 2005.

[11] T. Sarkar, F. Hu, Y. Hua, and M. Wick, "A real-time signal processing technique for approximating a function by a sum of complex exponentials utilizing the matrix pencil approach," *Digital Signal Processing - A Review Journal*, vol. 4, no. 2, pp. 127–140, Apr. 1994.

[12] 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.