Efficient Thermal Simulation for Run-Time Temperature Tracking and Management

Hang Li[‡], Pu Liu[†], Zhenyu Qi[†], Lingling Jin[§], Wei Wu[§], Sheldon X.-D. Tan[†], and Jun Yang[§]
[†]Department of Electrical Engineering, University of California, Riverside, CA 92521
[‡]Micron Technology Inc., San Jose, CA 95131
[§]Department of Computer Science and Engineering, University of California, Riverside, CA 92521

ABSTRACT

As power density increases exponentially, run-time regulation of operating temperature by dynamic thermal management becomes imperative. This paper proposes a novel approach to real-time thermal estimation at chip level for efficient dynamic thermal management in lieu of the thermal sensors, which are erroneous and having longer delays. Our new approach is based on the observation that the average power consumption of architecture level modules in microprocessors running typical workloads determines the trend of temperature variations. Such a feature can be exploited by applying fast moment matching technique in frequency domain. To obtain the transient temperature changes due to initial condition and constant power input pattern, numerically stable moment matching approach is carried out to speed up on-line temperature tracking with high accuracy and low overhead. The resulting fast thermal analysis algorithm has linear time complexity in run-time setting and leads to about two orders of magnitude speed-up over traditional integration-based transient analysis. The average maximum error under running typical benchmarks is only about 0.37°C as compared to other well-accepted simulation tools.

1. INTRODUCTION

As current IC technology enters nanometer realm, extremely high package density and operating frequency will lead to drastically increase of power density. The exponential power density increase will in turn cause average chip temperature to raise rapidly [2]. Furthermore, local hot spots, which have much higher power densities than the average, make local temperature even higher.

Higher temperature has significant adverse impacts on chip performance and reliability. It is believed that prompt real-time regulation of on-chip temperature by dynamic thermal management (DTM) is required for today's high-performance microprocessor and embedded systems [3, 15].

The basic idea of DTM is to dynamically reduce the temperature of some hot units (spots) in a chip via a suite of techniques such as activity migration, local toggling, dynamic voltage/frequency scaling [3, 15]. Performing DTM at architecture level is advantageous in that it can capture the run-time behavior of the program, and quickly adapt to different features within or across different programs.

One of the most critical aspects of thermal modeling and simulation for DTM is to efficiently capture the temperature changes due to the variations of the power consumption caused by the DTM techniques at chip architecture level. DTM performed at run-time requires accurate real-time sensing of temperature for each functional blocks. Previous research relies on CMOS-based sensor for on-line temperature tracking, which renders imprecision, delay and space overhead for hardware implementation [3, 7, 10]. These sensor noises could degrade DTM performance significantly due to conservative triggering of DTM [15]. One viable alternative soluton to this problem is to use fast on-chip thermal estimation technique in software form to replace the thermal sensors for effective DTM application.

*This work is funded by NSF CAREER Award CCF-0448534 and UC Senate Research Fund 05-06.

Although many efficient algorithms have been proposed for gate and circuit level thermal analysis [4, 17–20], less attention has been paid to thermal modeling and simulation at chip-architecture level. An architecture level thermal modeling and simulation tool called HotSpot [15] was developed to exploit and study different DTM techniques in regulating microprocessor operating temperature for representative benchmark programs. HotSpot provides an accurate architecture level thermal modeling based on equivalent circuit of thermal resistances and capacitances that correspond to the micro-architecture blocks and essential aspects of packaging. Component-wise temperatures are derived from the power consumptions generated by power simulations.

However, the efficiency of HotSpot for evaluating different DTM techniques depends on the execution time of transient thermal simulation throughout the program execution. HotSpot uses conventional integration-based transient simulation conducted at each execution interval in order to get the whole temperature profile. To obtain the temperature at certain running point of the program, all the previous temperature points should be generated since every point depends on its previous points. For a modern benchmark program which has tens to hundreds of billions of instructions, this method can lead to very long simulation time.

In this paper, we propose a fast run-time thermal simulation algorithm at architecture level for effective dynamic thermal management. Our idea is inspired by the observation that the average power in a certain amount of time determines the temperature variation trend. And this fact favors the application of frequency domain moment matching method in transient analysis, by which the time domain performance can be characterized by a few dominant poles. Since the analysis is performed in pure frequency domain and the resulting system transient response is in an analytical closed-form expression in terms of time, the execution time of temperature calculation can been improved significantly. In addition, the poles of the thermal circuits can be pre-computed off-line or at the initial stage, only the changing moments are computed in the run time, which leads to very fast linear time thermal analysis method.

The rest of the paper is organized as follows: Section 2 briefly mentions the architecture level thermal modeling in [16]. Section 3 demonstrates the relationship between average power and temperature variation during thermal circuit simulation. In section 4, we describe the entire flow of our fast algorithm with theoretical analysis regarding to time efficiency. The experimental results are summarized and compared to HotSpot and SPICE results in section 5 to validate our method, with conclusions and future works in section 6.

2. ARCHITECTURE LEVEL THERMAL MODELING

In circuit level thermal RC modeling, volume meshing is used to descretize the entire circuit structure, and finite difference or finite element method is used [18, 19] to descretize partial differential equation depicting thermal conducting phenomena. The resulting RC circuit is typically huge. At architecture level, however, due to the limited components at floor-plan and unknown details of physical implementation, the corresponding RC model is compact, and the accurate extraction of thermal resistance and capacitance is critical to the application of thermal analysis.

In this paper, we follow thermal modeling method at the architectural level in [8], where a fairly accurate equivalent RC model,

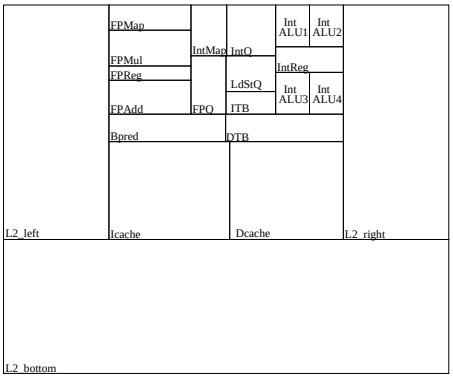

Figure 1: Floor-plan of Compaq Alpha 21364

which is verified by other commercial tools, is developed from floor-plan information. For a modern chip with CBGA packaging, heat sinks and cooling systems, there exists two main heat conduction paths, where the heat generated by active silicon die area can flow either through the convective ambient air, or the printed-circuit board. The primary RC circuit lies in the silicon die area, where the floor-plan information is provided to obtain the equivalent thermal resistance and capacitance. The floor-plan example we used in this paper is depicted in Fig 1. As shown in Fig. 1, the thermal resistance between two adjacent modules are determined by the common border length shared by them. Spreading/constriction resistances are also considered as in [11]. Each unit has a thermal capacitance to the thermal ground, which is determined by individual unit's area. And a scaling factor is needed to bridge the gap between this single-lumped capacitance and a distributed one. Besides the active die area, there are two additional heat spreader and heat sink layers lying underneath it. More component units are developed in the model corresponding to these two layers, but without active power sources for each component as appeared in the active die area.

Finally, in modeling the package to air interface, an equivalent convection resistance $R_{convection}$ is assigned, and a sustained power source is attached between the thermal ground (temperature of ambient air) and the bottom of the package. Calibration of this resulting model parameters is done as in [16], to provide a convergent results as compared to other commercial tools, as well as a good distribution of benchmark behaviors in the final experimental results. The thermal circuit netlist is stamped into matrix representation, and processed in circuit simulation phase discussed in section 4.

3. AVERAGE POWER AND TEMPERATURE VARIATION

Functional block's temperature variation during run time is caused by the irregular power trace generated by each unit in the floor-plan. This power input is composed of the DC signal which determines the global variation trend, and small AC disturbation added around that DC value for the oscillation of that trend. By performing FFT on a typical input power trace pattern, our spectrum analysis shows that most of the energy in the power trace concentrates on the DC component. High frequency components are normally at least two orders of magnitude smaller than the DC component.

Since most of the power lies in DC inputs, we can calculate the average power within a period of time to represent the power variation during that period. This averaged power value will act as a constant DC input to thermal circuit, and the transient response computed should be close to the exact response. In addition, current industry practices use counter register to compute on-line average power [7], which, as well, favors our scheme of using average power in predicting the temperature variation while on-the-fly of program execution.

4. FAST THERMAL ANALYSIS METHOD

In HotSpot, the transient temperature changes are computed by solving thermal differential equations using numerical integration method based on the *fourth-order Runge-Kutta method*, which is similar to the method used by SPICE simulator. In this paper, we propose a new approach to compute the transient temperature changes based on frequency domain moment matching concept. Moment matching concept was proposed for efficiently computing the transfer functions of RLC circuits for fast timing analysis of VLSI interconnects [14]. The main idea of moment matching is that the transient behaviors of a dynamic system can be accurately described by a few dominant poles of the systems. Those poles can be efficiently computed by finding the leading moments of the variable response $X(s)$ at node x excited by impulse input at the input node in frequency domain.

In our problem, we apply moment matching algorithm for computing the transient temperature responses under initial temperature conditions and average power inputs for a given time interval. The main cost of computing one moment is the same as solving temperature at one time step in traditional integral-based method.

4.1 Thermal Moment Matching (TMM) with Initial Condition and DC Inputs

For the equivalent thermal circuits with thermal resistor and capacitors and power trace input, we use can Modified Nodal Analysis to formulate the thermal circuit equation as:

$$\mathbf{Gx} + \mathbf{C}\dot{\mathbf{x}} = \mathbf{Bu} \tag{1}$$

Here we only consider the DC component of power trace $\mathbf{u(t)}$. $\mathbf{C}$ and $\mathbf{G}$ are capacitive and conductive circuit matrices, $\mathbf{x}$ is the vector of node temperature. $\mathbf{u}$ is the vector of independent power sources, and $\mathbf{B}$ is the input selector matrix. $\mathbf{X}(0)$ is the initial temperature value at each node. In frequency domain, the Laplace transformation of the state equation (1) can be rewritten as

$$\mathbf{GX}(s) + \mathbf{C}(s\mathbf{X}(s) - \mathbf{X}(0)) = \frac{1}{s}\mathbf{Bu} \tag{2}$$

In traditional AWE based moment matching method [13], transfer functions between designated sources or ports are computed. As a result, n moment series have to be computed for each node for n input sources as each source stimulates a moment vector at each node. In our problem, we only compute one moment series at each node as the response from the initial condition and constant DC power inputs are considered in the moment matching directly. As a result, the computation costs of the proposed method is not related to the number of input sources.

Specifically, let $\tilde{\mathbf{X}}(\mathbf{s}) = \mathbf{sX(s)}$, then the above equation becomes:

$$\mathbf{G}\tilde{\mathbf{X}}(s) + s\tilde{\mathbf{X}}(s) = s\mathbf{CX}(0) + \mathbf{Bu} \tag{3}$$

We then expand the $\tilde{\mathbf{X}}(s)$ using Taylor's series at $s = 0$, to have

$$\mathbf{G}(\mathbf{m_0} + \mathbf{m_1 s} + \mathbf{m_2 s^2} + ...) + \mathbf{sC}(\mathbf{m_0} + \mathbf{m_1 s} + \mathbf{m_2 s^2} + ...) = s\mathbf{CX}(\mathbf{0}) + \mathbf{Bu} \tag{4}$$

We then obtain the recursive moment computation formula as follows:

$$\begin{aligned} \mathbf{m}_0 &= \mathbf{G}^{-1}\mathbf{Bu} \\ \mathbf{m}_1 &= -\mathbf{G}^{-1}\mathbf{C}(\mathbf{m}_0 - \mathbf{X}(0)) \\ \mathbf{m}_2 &= -\mathbf{G}^{-1}\mathbf{C}\mathbf{m}_1 \\ &\vdots \\ \mathbf{m}_{2q} &= -\mathbf{G}^{-1}\mathbf{C}\mathbf{m}_{2q-1} \end{aligned} \tag{5}$$

After all the moments are computed, the response at each node can be written as

$$\mathbf{X}(s) = \frac{1}{s}\mathbf{m}_0 + \mathbf{m}_1 + s\mathbf{m}_2 + s^2\mathbf{m}_3 + \cdots + s^{2q-1}\mathbf{m}_{2q} + \ldots \tag{6}$$

The first term on the right-hand side is a step response in time domain and the rest of the moments then are used to find the rational approximation via Padé approximation. In order to find a q^{th} order Padé approximation, the first $2q$ moments are needed. Then we

obtain $2q$ moment matching equations:

$$\begin{aligned} &-(k_1 + k_2 + \cdots + k_q) = \mathbf{m}_0 - \mathbf{X}(0) \\ &-(\tfrac{k_1}{p_1} + \tfrac{k_2}{p_2} + \cdots + \tfrac{k_q}{p_q}) = \mathbf{m}_1 \\ &\vdots \\ &-(\tfrac{k_1}{p_1^{2q-1}} + \tfrac{k_2}{p_2^{2q-1}} + \cdots + \tfrac{k_q}{p_q^{2q-1}}) = \mathbf{m}_{2q-1} \end{aligned} \tag{7}$$

where p_i and k_i are the ith pole and residue in the partial fraction form of the response at node k

$$x_k(s) = \frac{1}{s}m_0 + \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + ... \tag{8}$$

For the entire moment matching flow, the poles are first computed by the projection-based method as shown in the next subsection. After this, all the residues k_i are computed using the first q equations from (7). The time domain responses are trivially obtained by taking inverse Laplace transformation of $x_k(s)$. Note also that since the transient responses start with an initial condition, the initial conditions need to be explicitly enforced as shown in the first equation in (7).

4.2 Finding Numerically Stable Poles and Residues

Traditional moment matching method [14] may produce unreliable poles (positive poles). A better way of finding poles is by projection based model order reduction, where moments are orthonormalized and are used to build a projection matrix. The projection matrix then is used to reduce the original circuit matrix by congruence transformation, which can ensure that the reduced system is passive (thus stable) [5]. Also by using this method, we only require q moments to find q poles.

Specifically, we obtain the first q moment vectors through (5). Then we form the following $N \times q$ matrix where each moment vector is a column.

$$M = [\mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{q-1}]_{N \times q} \tag{9}$$

where $q \ll N$, and N is the number of temperature variables (nodes) in the thermal circuit and also the dimension of the moment vectors. Then we orthonormalize M into a $N \times q$ projection matrix V such that columns in V are mutually orthogonal, i.e. $v_i^T v_j = 0, i \neq j$. Such an orthogonalization process can be easily carried out by using *Gram-Schmidt* orthonormalization algorithm [6]. Once we obtain the projection matrix V, the original circuit matrix $\mathbf{G}$ and $\mathbf{C}$ in (1) can be reduced to two $q \times q$ order reduced matrices by the *congruence transformation*:

$$\hat{\mathbf{G}} = V^T \mathbf{G} V, \quad \hat{\mathbf{C}} = V^T \mathbf{C} V \tag{10}$$

After this reduction process, the eigenvalues of matrix $\hat{\mathbf{G}}^{-1}\hat{\mathbf{C}}$ will be related to the dominant poles we are looking for as:

$$p_i = -\frac{1}{\lambda_i} \tag{11}$$

where p_i and λ_i are the ith pole and eigenvalue. This can be easily obtained by performing the eigen-decomposition of $\hat{\mathbf{G}}^{-1}\hat{\mathbf{C}}$. Once all the poles are computed, we then compute the residues at node x using equations in (7).

The proposed method guarantees stability of the responses as all the poles computed are stable pole (less than zero in their real part) [9, 12] due to the nature of congruence transformation.

4.3 Temperature Computation by Segmentation of Input Power Trace

It can be foreseen that if the interval over which an average power is obtained is increased, the temperature curve computed by TMM may have large deviations from the true temperature curve. The main reason is that average power is calculated over an interval that is too coarse-grained, losing local variation of the power. As a result, the DC component we extracted may not accurately reflect the temperature trend.

To resolve this problem, we can further partition the interval into several segments and each segment is simulated sequentially based on its start and end time. The end time will be the initial condition for the next segment TMM computation. The goal here is to make sure that the the drifts from the DC components to the average power is within a desired accuracy so that the moment matching method is accurate enough. Segment-by-segment simulation will be more accurate at only a small computing cost because the number of segment is limited, and the computation time of TMM for each segment is constant. We will illustrate the experimental results of this segmentation scheme on extremely long power traces (thus instruction cycles) in section 5.

5. EXPERIMENTAL RESULTS

The proposed algorithm TMM has been implemented in Matlab. We use the floor-plan of Compaq Alpha 21364 micro-architecture in Fig 1 for generating the power traces and equivalent thermal models similar to that in [16]. In order to perform a fair comparison, we implement traditional SPICE-like integration based thermal simulation method in Matlab for CPU time comparison. Since SPICE and HotSpot share the same computation mechanism, their execution time should be comparable. The accuracy comparison, however, is conducted in terms of the transient temperature results from HotSpot and TMM for a better reference.

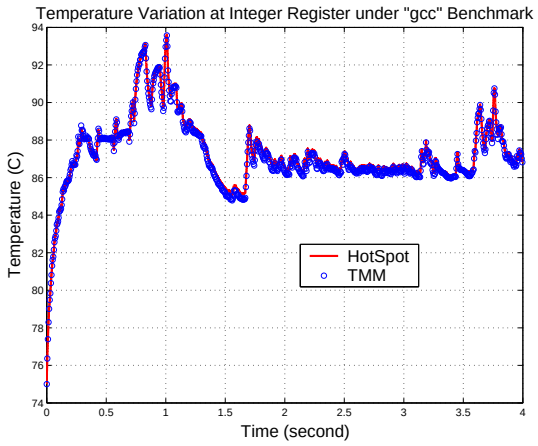

Figure 2: Temperature Comparison between TMM and HotSpot under gcc Benchmark

We evaluate our results by running 10 benchmark programs from SPEC CPU 2000 suite [1]. In our experiments, we run all the benchmarks on a 3GHz processor, which yields $0.33ns$ per cycle period. We sample each component's output power at 10K-cycle intervals, corresponding to $3.3\mu s$ in each time interval. As mentioned earlier, proper segmentation should be performed for extremely long power trace to avoid the error caused by average power drifting. We first arbitrarily set the segment window size to be 1500 intervals to evaluate the performance of TMM. On the other hand, SPICE and HotSpot will run the entire power trace interval-by-interval to derive the whole temperature profile.

Table 1 summarizes the statistics of these programs and experimental results. The simulated thermal circuit consists of 73 nodes. Although the circuit size is small, given the very long input power trace (tens of billions of instruction cycles), the simulation time of HotSpot/SPICE will still be long as shown in Table 1. We notice that our algorithm achieves almost 100X speed-up over SPICE. Considering the faster run time provided by Linux workstation, the actual run time speed-up of TMM over HotSpot will be larger. For accuracy evaluation, the average and maximum errors compared to HotSpot among all these 10 benchmarks are only 0.13°C and 0.37°C, which provides a highly accurate temperature prediction for on-line DTM application. In all the test cases, 7 poles are computed for the transient response analysis.

To show the detailed waveforms comparison of TMM with HotSpot results, we pick a run-time window of temperature variation at Integer Register File under program `gcc`. As illustrated in Fig 2, each point calculated by TMM matches very well with the corresponding temperature point in HotSpot curve.

We also study the effect of segment window sizes on the performance of TMM. Fig 3 depicts the run-time, average and maximum

Table 1: Performance Evaluation of TMM under Different Benchmarks

Program	Ins. Cycles(billion)	Exec. Time(s)	CPU(TMM)(s)	CPU(SPICE)(s)	Speed-up	Avg. Err.(°C)	Max. Err.(°C)
gcc	28.5	9.49	55	4357	79	0.09	0.37
wupwise	56.3	18.76	126	8615	68	0.10	0.42
eon	27.3	9.07	55	4166	75	0.11	0.27
gzip	24	7.99	55	3669	66	0.11	0.30
bzip	55.7	18.55	116	8514	73	0.05	0.34
lucas	46.7	15.53	90	7131	79	0.16	0.53
mesa	20.0	6.68	42	3065	72	0.17	0.34
parser	30.7	10.24	67	4700	70	0.13	0.43
swim	11	3.68	23.6	1690	72	0.3	0.49
vortex	74.3	24.75	147	11360	77	0.074	0.27
Average	-	-	-	-	-	**0.13**	**0.37**

errors for each TMM run under different window sizes. We select benchmark program `wupwise` as an example. From the figure we can see, that the execution time will decrease rapidly with the increasing window size. This is because of the decreased segment number to be calculated by TMM. The maximum error from TMM as compared to HotSpot will increase with a larger window size, which is due to a less accurate average power estimation within that period of time. However, the error is less than 0.7°C for the maximum window size of 6000. And the average error almost stays the same for all window sizes around only 0.1°C. Compared to a real temperature deviation of 2°C offered by temperature sensor in previous DTM scheme [16], our new algorithm provides a viable and reliable on-line temperature estimation for DTM applications.

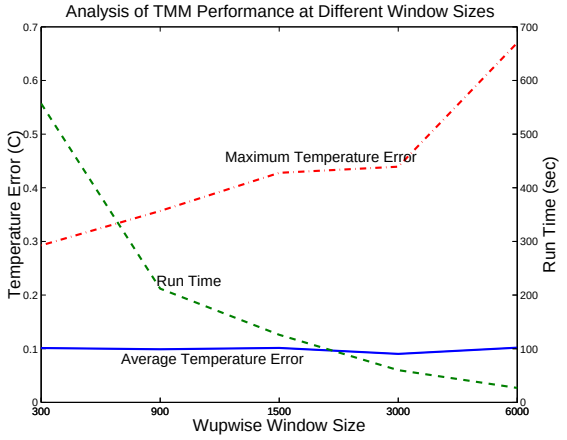

Figure 3: TMM Performance Evaluation under Different Window Sizes

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed an efficient thermal analysis technique suitable for run-time on-chip dynamic thermal management. The new approach exploits the effect of average input power patterns on the temperature variation of architecture level modules in microprocessors and embedded high-performance systems. We found that the average power determines the global temperature variation trend. And based on this observation, we proposed to use numerically stable moment matching method, which considers the initial conditions and DC power inputs, for computing transient temperature changes. The resulting fast thermal analysis algorithm has a linear time complexity in run-time setting and has about two orders of magnitude speed-up over traditional integration-based SPICE/HotSpot transient simulation methods with small accuracy loss. In the future, we will integrate our thermal simulation engine with the DTM controller to complete the architecture level DTM framework.

7. REFERENCES

[1] http://www.spec.org/cpu2000/CFP2000/.

[2] International technology roadmap for semiconductors(itrs), 2004 update, 2001. http://public.itrs.net.

[3] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proc. of Intl. Symp. on High-Performance Comp. Architecture*, pages 171–182, 2001.

[4] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang. *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.

[5] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by pade approximation via the lanczos process. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, May 1995.

[6] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3 edition, 1989.

[7] S. Gunther, F. Binns, D. Carmean, and J. Hall. Managing the impact of increasing microprocessor power consumption. In *Intel Technology Journal*, First Quarter 2001.

[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)*, pages 878–883, 2004.

[9] K. J. Kerns and A. T. Yang. Stable and efficient reduction of large, multiport RC network by pole analysis via congruence transformations. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 16(7):734–744, July 1998.

[10] K. Lee and K. Skadron. Using performance counters for runtime temperature sensing in high performance processors. In *the Workshop on High-Performance, Power-aware Computing(HP-PAC), in conjunction with the 2005 International Parallel and Distributed Processing Symposium*, Apr. 2005.

[11] S. Lee, S. Song, V. Au, and K. Moran. Constricting/spreading resistance model for electronics packaging. In *Proc. ASME/JSME Thermal Engineering Conference*, pages 199–206, Mar. 1995.

[12] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: Passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pages 645–654, 1998.

[13] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pages 352–366, April 1990.

[14] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. *Electronic Circuit and System Simulation Methods*. McGraw-Hill, New York, 1994.

[15] 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)*, pages 2–13, 2003.

[16] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature aware microarchitecture: Extended discussion and results. In *University of Virginia, Dept. of Computer Science, Technicial Report CS-2003-08*, Apr. 2003.

[17] B. Wang and P. Mazumder. Fast thermal analysis for vlsi circuits via semi-analytical green's function in multi-layer materials. In *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS)*, 2004.

[18] T. Y. Wang and C. C. Chen. 3-D thermal-ADI: a linear-time chip level transient thermal simulator. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 21(12):1434–1445, Dec. 2002.

[19] T. Y. Wang and C. C. Chen. Spice-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on model reduction. In *Proc. Int. Symposium. on Quality Electronic Design (ISQED)*, pages 357–362, 2004.

[20] Y. Zhan and S. Sapatnekar. Fast computation of the temperature distribution in vlsi chips using the discrete cosine transform and table look-up. In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pages 87–92, Jan. 2005.