Fast Thermal Simulation for Architecture Level Dynamic Thermal Management [*]

Pu Liu[†], Zhenyu Qi[†], Hang Li[†], Lingling Jin[‡], Wei Wu[‡], Sheldon X.-D. Tan[†], Jun Yang[‡]

[†]Department of Electrical Engineering, University of California, Riverside, CA 92521
[‡]Department of Computer Science, University of California, Riverside, CA 92521

ABSTRACT

As power density increases exponentially, runtime regulation of operating temperature by dynamic thermal managements becomes necessary. This paper proposes a novel approach to the thermal analysis at chip architecture level for efficient dynamic thermal management. Our new approach is based on the observation that the power consumption of architecture level modules in microprocessors running typical workloads presents strong nature of periodicity. Such a feature can be exploited by fast spectrum analysis in frequency domain for computing steady state response. To obtain the transient temperature changes due to initial condition and constant power inputs, numerically stable moment matching approach is carried out. The total transient responses is the addition of the two simulation results. The resulting fast thermal analysis algorithm leads to at least 10x-100x speedup over traditional integration-based transient analysis with small accuracy loss.

1. INTRODUCTION

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

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. It is believed that prompt regulation of on-chip temperature by dynamic thermal management (DTM) is required for today's high-performance microprocessor and embedded systems [3, 12].

The basic idea of the 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, 12]. One of the most critical aspects of thermal modeling and simulation for DTM is to efficiently compute the temperature changes due to the variations of the power consumption caused by the DTM techniques at chip architecture level.

Previously many research works have concentrated on thermal modeling and simulation at circuit or gate level [4]. Due to the large volume of thermal components and power sources at full circuit or gate-level, different schemes were proposed to increase the efficiency of thermal circuit simulation, which can be roughly classified into two categories.

The first method is based on the discretization on differential operators (finite deference method) or the field quality (finite element method). Examples are [16, 15], where the entire chip is discretized, and the heat transfer equation in partial differential form is solved by finite difference or finite element method. The main draw back of those methods is the huge sizes of the resulting thermal circuits due to volume meshing. Different techniques were proposed to solve with the extremely large thermal circuit, such as ADI in [16], and model order reduction in [15]. The second method is based on Green function method [14, 17], which provides a fast yet less accurate thermal simulation as compared to above methods due to the simplified two dimensional modeling of the thermal problem.

Although many efficient algorithms have been proposed for circuit or gate level thermal analysis, less attention has been paid to thermal modeling and simulation at chip-architecture level. Architecture level thermal behavior can not be seen at circuit or gate level at design time since different workloads generate different thermal profiles. 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. Further, recent studies show that architecture level thermal managements at small performance degradation cost can significantly reduce the packaging costs designed for worst cases [12, 13, 5]. An architecture level thermal modeling and simulation tool called HotSpot [12] 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 thermal circuit of thermal resistances and capacitances that correspond to the microarchitecture 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 models the thermal behaviors based on the equivalent thermal circuit which consists of thermal resistors and capacitors. It uses conventional integration-based transient simulation conducted at each execution interval in order to get the whole temperature profile. When a program is loaded into HotSpot, its power consumption is first obtained at regular intervals. Then, temperature at every interval is calculated taking the temperature at the last interval and power values at the past few intervals. 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 (weeks). As a result, a much faster yet accurate transient simulation technique is required to efficiently design and optimize the temperature-aware chip architecture.

In this paper, we propose a fast transient thermal simulation algorithm at architecture level for fast dynamic thermal management. Our idea is inspired by recent discoveries in the runtime behavior of programs over long periods of time. It has been shown that most program behavior is not random and actually presents strong periodic patterns due to the existence of loops or *phases* [9, 11, 10].

Many programs execute as a series of *phases* which characterizes certain program behavior at different times. Each phase may be very different from the others, while having a fairly homogeneous and periodic behavior within itself. This also reflects a periodic nature of each computation unit's output power consumption. Fig. 1 shows a snapshot of a typical power trace of an integer register running under program `Lucas` over 14 time periods in SPEC CPU 2000 suite [1]. Each period contains about 20 power values collected over 0.2 million-instruction interval. This trace was obtained from simulating a 3GHz processor and thus, each period translates to ~0.26ms and the entire trace represents ~3.6ms of program execution. As we can see, there is very strong periodicity even in this short amount of time.
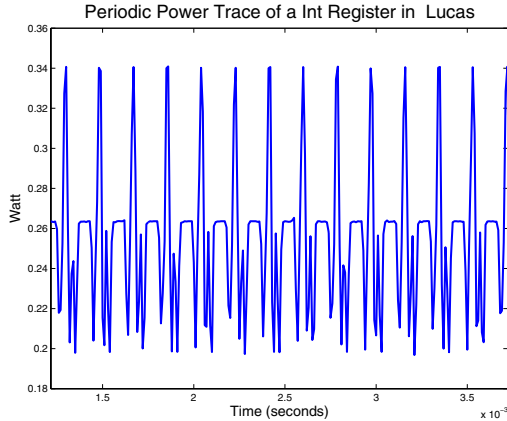

Figure 1: The power trace of an integer register in Lucas program

In this paper we exploit such periodic power traces of many hot modules at architecture level to speed up the transient thermal simulation. For instance, the integer register file is typically the hottest module for most benchmarks [13]. Our proposed new algorithm is based on the fact that the transient behavior of the linear thermal system is the sum of the zero-input natural response and zero-state forced response. We applied two efficient algorithms to compute the two responses: (1) For periodic power trace input, frequency domain spectrum analysis is performed to calculate the steady-state response. (2) For transient behaviors of temperature, moment matching technique is used with the consideration of the initial state and DC input value computed in the first step. Since the analysis is performed in pure frequent domain and the resulting system transient response is in an analytical closed-form expression in terms of time, the run time has been improved significantly.

The rest of the paper is organized as follows: Section 2 briefly mentions the architecture modeling in [13], and our modification of the architecture as opposed to [13]. In section 3, we describe our new algorithm, with an overview of our algorithm infrastructure, followed by the two main parts described above, as well as the theoretical analysis regarding to time efficiency. The experimental results are summarized and compared to SPICE-like simulator in section 4 to validate our method, with conclusions and future works in section 5.

2. ARCHITECTURE LEVEL THERMAL MODELING

Generally speaking, the heat transfer phenomena is governed by the following differential equation [4]:

$$\rho C_p \frac{\partial T(\vec{r},t)}{\partial t} = \bigtriangledown \cdot [\kappa(\vec{r},T) \cdot \bigtriangledown T(\vec{r},t)] + g(\vec{r},t) \qquad (1)$$

where $T(K)$ is the temperature, $\rho(Kg/m^3)$ is the density of the material, $C_p(J/m^3 \cdot K)$ is the specific heat, $\kappa(W/m \cdot K)$ is the thermal conductivity, and $g(W/m^3)$ is the heat energy generation rate. The heat flow described by this differential equation has the similar format as that for electrical current, and there is a well-known duality between them. The heat flow passing through a thermal resistor is equivalent to the electrical current, and the temperature difference corresponds to voltage difference. There is also the thermal-equivalent capacitance where the heat is absorbed. Based on these observations, an equivalent thermal RC circuit will be derived and solved in dealing with thermal issues.

In circuit level thermal RC circuit modeling, volume meshing is used to discretize the entire circuit structure, and finite difference or finite element method is used to discretize equations(1). 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.
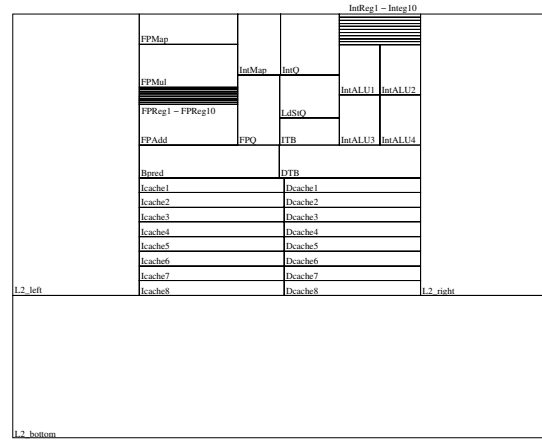

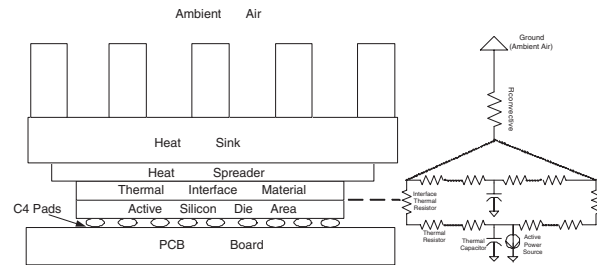**Figure 2: Modified architecture of Compaq Alpha 21364**

Figure 3: Modern Chip Packaging Structure and Equivalent RC Circuit Modeling

In this paper, we follow thermal modeling method at the architectural level thermal in [5], where a fairly accurate equivalent RC model, 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 as shown in Fig 3, 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 2. The difference between our floor-plan model and the one in [13] is that

we divide some critical computing components, such as Integer Register (IntReg) and Floating Point Register(FPReg), into more detailed pieces so that more accurate temperature variation could be obtained in these temperature-critical components used during DTM. As shown in Fig. 2, 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 [7]. Each unit has a thermal capacitance to the thermal ground, which is determined by individual unit's area. Also 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 lie 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 package bottom. Calibration of this resulting model parameters is done as in [13], to provide a convergent results as compared to other commercial tools, as well as a good distribution of benchmarks behaviors in the final experimental results. The thermal circuit netlist is stamped into matrix representation, and processed in the circuit simulation phase discussed in Section 3.

3. NEW ALGORITHM FOR FAST TRANSIENT THERMAL SIMULATION

We start with equivalent thermal circuit model developed at architecture level as discussed in the previous section.

3.1 Algorithm Preliminary

For a general dynamic system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{2}$$

where $\mathbf{A}$ is the system matrix and $\mathbf{B}$ is the input position matrix. $\mathbf{x}$ and $\mathbf{u}$ are the state variable and input vectors. The complete response is the sum of the zero-input response and the zero-state response starting from time t_0 as shown in Eq.(5).

$$\mathbf{x}(\mathbf{t}) = \mathbf{e}^{\mathbf{A}(\mathbf{t}-\mathbf{t_0})}\mathbf{x_0} + \int_{\mathbf{t_0}}^{\mathbf{t}} \mathbf{e}^{\mathbf{A}(\mathbf{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)\mathbf{d}\tau. \tag{3}$$

The first term on the right-hand side is the zero-input response due to the initial condition and the second term on the right-hand size is the zero-state response due to input sources $\mathbf{u(t)}$ only.
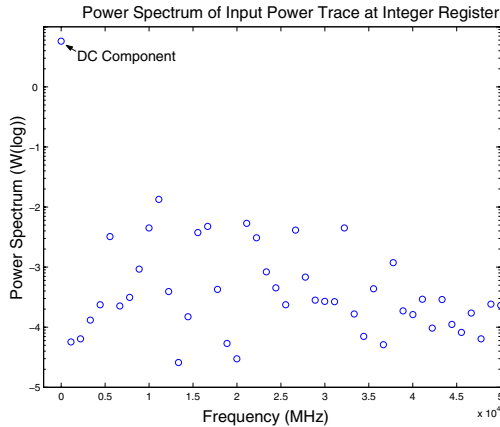
Figure 4: The power spectrum of an integer register file in Lucas program

Our spectral analysis of the typical power trace of many architecture modules shows that most of energy in the power trace concentrates on the DC as shown in Fig. 4 (Y axis is plotted in logarithm). As a result, we partition the power trace $\mathbf{u(t)}$ into two parts $\mathbf{u_1}$ and $\mathbf{u_2(t)}$. $\mathbf{u_1}$ is the DC component of the power input and $\mathbf{u_2(t)}$ is the periodic component of the power input where

$$\mathbf{u_2(t+T)} = \mathbf{u_2(t)}, \tag{4}$$

and T is the time of the period in power inputs. We stress here that the practice power traces are not exact periodic over long time. But typically in a given phase, changes between different periods are so small that approximating them as periodic inputs will not cause significant errors. As a result, Eq.(5) can be written as

$$\mathbf{x}(\mathbf{t}) = \mathbf{e}^{\mathbf{A}(\mathbf{t}-\mathbf{t_0})}\mathbf{x_0} + \int_{\mathbf{t_0}}^{\mathbf{t}} \mathbf{e}^{\mathbf{A}(\mathbf{t}-\tau)}\mathbf{d}\tau\mathbf{B}\mathbf{u_1} + \int_{\mathbf{t_0}}^{\mathbf{t}} \mathbf{e}^{\mathbf{A}(\mathbf{t}-\tau)}\mathbf{B}\mathbf{u_2}(\tau)\mathbf{d}\tau. \tag{5}$$

Now the first two terms on the right hand side are only function of the initial condition and DC inputs of power trace. The third response of the circuit is stimulated by periodic power inputs.

The main idea of this work is to efficiently compute the first two responses by using moment matching method and the third periodic response by using fast spectral analysis method in frequency domain to exploit the periodic patterns of power inputs. In the next two subsections, we present these two algorithms.

3.2 Moment Matching Considering 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:

$$\mathbf{Gx} + \mathbf{C\dot{x}} = \mathbf{Bu_1}, \tag{6}$$

here we only consider the DC component of power trace $\mathbf{u_1(t)}$. $\mathbf{C}$ 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. In frequency domain, the Laplace transformation of the state equation (6) can be rewritten as

$$\mathbf{GX}(s) + s\mathbf{CX}(s) - \mathbf{x}(0) = \frac{1}{s}\mathbf{Bu_1}. \tag{7}$$

In the traditional AWE based moment matching method [8], 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 which are considered in the moment matching directly. As a result, the computation costs of the proposed method is not related to the number of sources.

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

$$\mathbf{G}\tilde{\mathbf{X}}(s) + s\mathbf{C}\tilde{\mathbf{X}}(s) = s\mathbf{CX}(0) + \mathbf{Bu_1} \tag{8}$$

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

$$\mathbf{G}(\mathbf{m_0} + \mathbf{m_1}\mathbf{s} + \mathbf{m_2}\mathbf{s}^2 + ...) + s\mathbf{C}(\mathbf{m_0} + \mathbf{m_1}\mathbf{s} + \mathbf{m_2}\mathbf{s}^2 + ...) = s\mathbf{Cx(0)} + \mathbf{Bu_1} \tag{9}$$

We then obtain the recursive moment computation formula as follows:

$$\begin{aligned}
\mathbf{m}_0 &= \mathbf{G}^{-1}\mathbf{Bu_1} \\
\mathbf{m}_1 &= -\mathbf{G}^{-1}\mathbf{C}(\mathbf{m_0} - \mathbf{x}(0)) \\
\mathbf{m}_2 &= -\mathbf{G}^{-1}\mathbf{Cm_1} \\
&\vdots \\
\mathbf{m}_{2q} &= -\mathbf{G}^{-1}\mathbf{Cm}_{2q-1}
\end{aligned} \tag{10}$$

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

$$\mathbf{X}(s) = \frac{1}{s}\mathbf{m}_0 + \mathbf{m}_1 + s\mathbf{m}_2 + s^2\mathbf{m}_3 + \cdots + s^{2q-1}\mathbf{m}_{2q} + \dots \quad (11)$$

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 qth order Padé approximation, the first $2q$ moments are needed. Then we obtain $2q$ moment matching equations:

$$\begin{aligned}
-(k_1 + k_2 + \cdots + k_q) &= \mathbf{m}_0 - \mathbf{x}(0) \\
-(\tfrac{k_1}{p_1} + \tfrac{k_2}{p_2} + \cdots + \tfrac{k_q}{p_q}) &= \mathbf{m}_1 \\
&\vdots \\
-(\tfrac{k_1}{p_1^{2q-1}} + \tfrac{k_2}{p_2^{2q-1}} + \cdots + \tfrac{k_q}{p_q^{2q-1}}) &= \mathbf{m}_{2q-1}
\end{aligned} \quad (12)$$

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

$$x_k(s) = \frac{1}{s}m_0 + \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \dots \quad (13)$$

The poles are first computed by the improved multi-node moment matching method as show in next subsection [6]. After this, all the residues k_i are computed using q equations from Eq.(12). 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 explicitly be enforced as shown in the first equation in Eq.(12).

Since most of the power energy is in the DC inputs, the response computed by moment matching can be close to the exact response, and the spectrum analysis solution typically adds small transient changes in the node temperature as observed by our experiments.

3.3 Multi-Node Moment Matching For Numerically Stable Estimation of Poles

One issue with direct moment match is that the high order moments will become less accurate as high order moments are power function of poles as shown in Eq.(12). High order pole information will soon be lost numerically. To resolve this issue, we apply the multi-node moment match (MMM) technique to get more numerical stable estimation of the system poles [6].

The main idea of MMM method is to estimate the poles by using the moment information from different nodes or from different input stimulus instead of single node with single stimulus as done in traditional moment matching method [8]. The rational behind this method is that poles from different node and from different input stimulus are same. By using MMM, we only need less orders of moments to compute the same number of poles. For instance, if moments from different node with a single input are used only, we can get the p poles by using just $p + 1$ moments. While if k inputs are allowed, then we can get the p poles by just using $p/k + 1$ moments for each input with about the same computing cost. Since the pole information are better preserved numerically in small-order moments as shown in Eq.(12), we can get much better and numerically stable estimation of the poles.

Although we compute the direct responses $X(s)$ in frequency domain instead of the transfer functions in our work, MMM can still be applied. The reason is that the initial condition and DC power inputs do not bring any additional poles into the response. Specifically, we assume there exist reduced system of order q to be determined

$$\mathbf{A}_{q \times q}\dot{\mathbf{x}} = \mathbf{x} + \mathbf{r_q}u.$$

We select q nodes and their corresponding moments become the moment vectors $\tilde{\mathbf{m}}_0, \tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_{q+1}$. According to Eq.(10), we have

$$[\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_q] = \mathbf{A}_{q \times q}[\tilde{\mathbf{m}}_0 - \tilde{\mathbf{x}}(0), \tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_{q-1}] \quad (14)$$

$$\mathbf{A}_{q \times q} = [\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_q][\tilde{\mathbf{m}}_0 - \tilde{\mathbf{x}}(0), \tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_{q-1}]^{-1} \quad (15)$$

where $\tilde{\mathbf{x}}(0)$ are the initial condition vector on the selected q nodes. Then the eigenvalues of $\mathbf{A}_{q \times q}$ are the reciprocals of q dominant poles of the original system in Eq.(6). Notice that only $q + 1$ moments are required to compute the q dominant poles.

3.4 Spectrum Analysis in Frequency Domain

In this subsection, we discuss spectrum analysis method for periodic zero state steady response of the thermal circuits under periodic input power traces. The basic idea is to transform the input signals into frequency domains via discrete Fourier transformation (DFT) and then compute the responses of dominant Fourier coefficients or harmonics. The resulting response Fourier coefficients are then transformed to time-domain to get the steady-state time responses.

In discrete Fourier transformation, if we sample k points in the time domain, we have k harmonics in the frequency domain. Then we solve for responses of each harmonics on the thermal circuits as show below:

$$\begin{aligned}
\mathbf{A}(\omega_1)\mathbf{X}(\omega_1) &= \mathbf{P}(\omega_1) \\
\mathbf{A}(\omega_2)\mathbf{X}(\omega_2) &= \mathbf{P}(\omega_2) \\
&\vdots \\
\mathbf{A}(\omega_k)\mathbf{X}(\omega_k) &= \mathbf{P}(\omega_k)
\end{aligned} \quad (16)$$

where $\omega_i, i = 1 \dots k$ are the harmonic frequencies, $\mathbf{A}(\omega)$ is the MNA matrix stamped with these thermal elements at frequency ω, $\mathbf{X}(\omega)$ is the temperature vector at each component at frequency ω, and $\mathbf{P}(\omega)$ is the computed harmonics of the power input vector from DFT.

By this transformation, the computational cost can be reduced significantly especially for the long cycle simulation. The reason is that we only compute the stead state using one period and the number of MNA equations depends on only the number of sampling points. If the faster simulation is required, we can reduce the sampling frequency to speed up the calculation at the cost of accuracy loss. But at the same time we make sure that the sampling rate should satisfy the Shannon's sampling theorem. Finally we convert the frequency response to time domain responses by inverse discrete Fourier transformation.

Note that the final steady state temperature will be the temperature we computed using spectrum analysis and environmental temperature, which is treated as the ground voltage in our equivalent thermal circuits. The actual temperature response is the sum of the zero-input response and the zero-state response.

Since the solution from the spectrum analysis in frequency domain is the steady state solution, which means it will happen at infinity time when system response become stable, it is less accurate for response at $t = 0$. But the difference is very small as energy in the periodic inputs are typically very small compared to the DC components. This has been observed and verified by our experimental results.

3.5 Application to Dynamic Thermal Management

Our techniques have great benefits to perform DTM at architecture level. For instance, given an initial thermal setting and the fact that a phase of the program is periodic, our technique can predict if the current phase will reach a critical temperature and if yes, how soon it will happen. Thus, before such a critical temperature is reached, effective DTMs such as dynamic voltage/frequency scaling, local toggling, and activity migration can be carried to cool down the soon-to-be hot modules preventing them from entering the temperature critical stage. This can be easily computed using

our proposed two-step algorithm. In some situations, the DC component striped periodic power trace only adds small distributions on the general transients of the temperature. The moment matching responses can be used to give fairly accurate time estimation.

Another important question of DTM is whether a program having long time of periodic behavior will ever reach a critical temperature, i.e., what is the steady thermal state. If the steady state is well below a temperature threshold, then there is no need to perform DTM as long as the same periodicity holds. Our moment matching algorithm can directly compute such an asymptotic steady state from the very beginning of the period, producing fast and accurate prediction. On the other hand, if the steady state is very close to some critical temperature, only the pure spectrum analysis is needed to find out the perturbations around the steady temperature. In that case, DTM is necessary whenever the temperature pulses surpass the threshold.

If there are several phases involved in the input power traces, we can compute the responses for each phase using the proposed methods and combine them to report the temperature profile.

3.6 Time Complexity Analysis

For a thermal circuit with n nodes, if only a few moments ($q \ll n$) are required, the time complexity for moment matching is approximately $O(n^{1.5})$, where $n^{1.5}$ is the approximately cost of one LU decomposition of the sparse matrix G. The time complexity for spectrum analysis is approximately $O(kn^{1.5})$, where k is the number of sampling points in Eq.(16) as we need to solve k times of the equation sets in frequency domain. But the sampling only happen in one period. So k is not related to the time interval.

Notice that the time complexities of both analysis are independent of time intervals or number of time steps used in the traditional integration based transient simulation, which is the major advantage and speedup over traditional methods.

4. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented in Matlab. We use the modified Compaq Alpha 21364 microarchitecture in Fig 2 for generating the power traces and equivalent thermal models similar to that in [13]. To perform fairly comparison, we also implements traditional integration based thermal simulation method used in HotSpot in Matlab in order to compare the accuracy and the speed difference between the two approaches.

We evaluate our results using benchmarks from the SPEC CPU 2000 suite [1]. We select 3 programs Art, Lucas, and Wupwise and run for 10 billion instructions and simulated the periodical portions. Since the hottest unit is the integer register file, we compute the temperature changes of one typical integer register file as example.

Table 1: CPU time comparison for running SPEC CPU2000 programs

Program	#II	#S	CPU (HotSpot)	CPU (TMMSpectrum)
Lucas	20000	18	183.74	0.78
Lucas	30000	18	219.43	0.77
Art	10000	614	72.23	15.67
Art	30000	614	242.49	15.55
Wupwise	10000	24	66.31	1.11
Wupwise	30000	24	515.02	1.10

Table 1 summarizes the statistics of the three programs and experimental parameters. Column #II is the number of instruction intervals, which is about 10μs, Column *#S* is the number of samplings used in the spectrum analysis. Columns *CPU (HotSpot)* and *CPU (TMMSpectrum)* are the CPU times for the traditional simulation method used in HotSpot and our proposed method. The CPU time is in unit of second. The simulated thermal circuit consists of about 166 nodes. Although the thermal circuit is small, given the

very long power input trace (billions of instruction cycles), the simulation time of HotSpot will still be long as shown in Table 1. In the test cases, 6 poles are computed from multi-node moment matching to compute the transient response. We find 6 poles typically give fairly well results.

From Table 1, we can see that the proposed method have 10x to 100x speedup over the traditional simulation method.
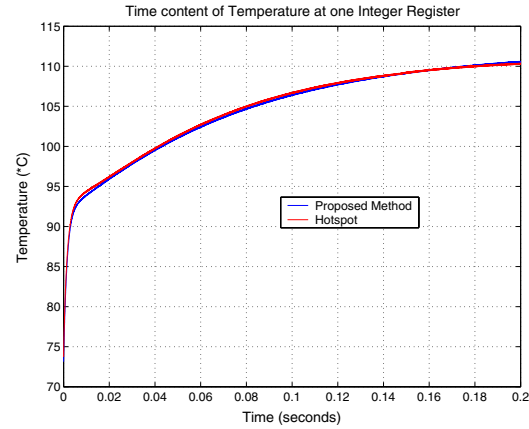

Figure 5: The temperature change comparison for Lucas program

Fig. 5 shows the transient temperature change comparison for program Lucas for about 20000 simulation points where each point is 3.3μs. Fig. 6 shows the transient temperature change comparison for program Art for about 10000 points. Fig. 7 shows the transient temperature change for Wupwise program. It can be seen that the results from our new algorithm match pretty well with the HotSpot simulation results but with at least 10X shorter CPU time as shown in Table 1. Note that given longer simulation interval, the speedup will be further increased as the new method dose not depend on the number of simulation intervals.
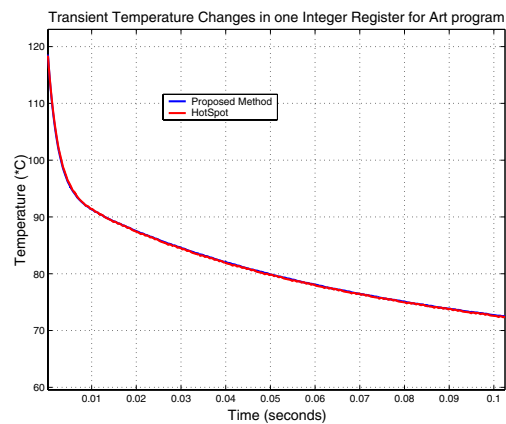

Figure 6: The temperature change comparison for Art program

We also notice that when the temperature increases over the time the temperature computed by the proposed method may have noticeable differences from the ones given by HotSpot. The main reason is that the periodic power inputs may change slightly over the time. As a result, the DC components may drift from the value we use for the moment matching at $t = 0$ as shown Fig. 5. To resolve this problem, we can partition the simulation intervals into several segments and each segment is simulated sequentially based

on its start and end times. The goal here is to make sure that the DC components from periodic power trace drifts are very small so that the moment matching method is accurate enough. Segment-by-segment simulation will be more accurate at a small computing cost.
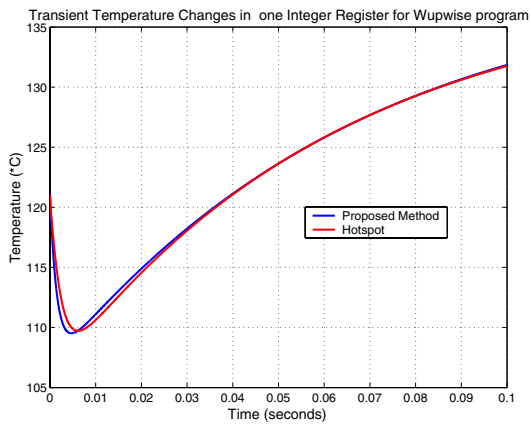

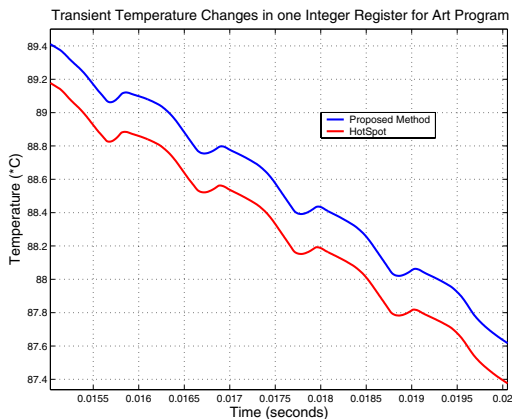**Figure 7: The temperature change comparison for Wupwise program**

Figure 8: The detailed periodic temperature change comparison for Art program

To show the detail waveforms due to periodic power trace, we enlarge Fig. 6, which is shown in Fig. 8. It can be shown that the temperature indeed changes periodically over the time. Both HotSpot and the proposed new method match each other very well.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed an efficient thermal analysis technique for architecture level dynamic thermal management. The new approach exploits the periodic patterns in power consumptions of the architecture modules in microprocessors and embedded high-performance systems. We proposed to use spectrum analysis in frequency domain to compute the periodic responses of temperatures and numerically stable moment matching method, which considers the initial conditions and DC power inputs, for computing transient temperature changes. The total transient response is the addition of the two simulation results. The resulting fast thermal analysis algorithm leads to at least 10x speedup over traditional integration-based SPICE-like HotSpot transient simulation with small accuracy loss. In the future, we will integrate our thermal simulation engine

with the DTM optimizer described in section 3.5 to complete the architecture level DTM framework.

6. 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*, 2001, pp. 171–182.

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

[5] 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.

[6] Y. Ismail, "Efficient model order reduction via multi-node moment matching," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2002, pp. 767–774.

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

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

[9] T. Sherwood, E. Perelman, and B. Calder, "Basic block distribution analysis to find periodic behavior and simulation points in applications," in *the International Conference on Parallel Architectures and Compilation Techniques(PACT2001)*, 2001, pp. 3–14.

[10] T. Sherwood, E. Perelman, G. Hamerly, S. Sair, and B. Calder, "Discovering and exploiting program phases," in *IEEE Micro:Micro's Top Picks from Computer Architecture Conferences*, 2003, pp. 84–93.

[11] T. Sherwood, S. Sair, and B. Calder, "Phase tracking and prediction," in *Proc. IEEE International Symposium on Computer Architecture (ISCA)*, 2003, pp. 45–57.

[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.

[13] ——, "Temperature aware microarchitecture: Extended discussion and results," in *University of Virginia, Dept. of Computer Science, Technicial Report CS-2003-08*, Apr. 2003.

[14] 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.

[15] 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*, vol. 21, no. 12, pp. 1434–1445, Dec. 2002.

[16] ——, "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)*, 2004, pp. 357–362.

[17] 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)*, Jan. 2005, pp. 87–92.