# Compact Thermal Modeling for Temperature-Aware Design

UNIVERSITY OF VIRGINIA DEPT. OF COMPUTER SCIENCE TECH. REPORT CS-2004-13*

APRIL 2004

Wei Huang†, Mircea R. Stan†, Kevin Skadron‡, Karthik Sankaranarayanan‡
Shougata Ghosh†‡, Sivakumar Velusamy‡

† *Dept. of Electrical and Computer Engineering,* ‡*Dept. of Computer Science*
*University of Virginia, Charlottesville, VA 22904*
{*wh6p, mircea, sg7w*}*@virginia.edu,* {*skadron, karthik, sv7d*}*@cs.virginia.edu*

## Abstract

*Thermal design in sub-100nm technologies has become one of the major challenges to the CAD community. Thermal effects on design aspects such as performance, power and reliability have to be thoroughly and seriously considered during the entire design flow in order to achieve faster design convergence and optimal design.*

*This paper expands the discussion in our conference paper [8]. We first introduce the idea of* temperature-aware *design. We then propose a compact thermal model which can be integrated with modern CAD tools to achieve a temperature-aware design methodology. Finally, we use the compact thermal model in a case study of microprocessor design to show the importance of using temperature as a guideline for the design. The results from our thermal model show that a temperature-aware design approach can provide more accurate estimations, and therefore better decisions and faster design convergence.*

## 1. Introduction

As CMOS technology is scaled into the sub-100nm region, the power density of microelectronic designs increases steadily. For example, the power density of high-performance microprocessors has already reached 50W/cm$^2$ at the 100nm technology node, and it will soon reach 100W/cm$^2$ at technologies below 50nm[1]. As a result, the average temperature of the die also increases rapidly. Furthermore, local hot spots on the die usually have significantly higher power densities than the average, making the local temperatures even higher.

Temperature has significant impacts on microelectronic designs. First, transistor speed is slower at higher temperature due to the degradation of carrier mobility. A back-of-the-envelope calculation shows that a single inverter is about 35% slower at 110°C than at 60°C. Second, the temperature dependance of leakage power is significant. Leakage power can be orders of magnitude greater at higher temperatures[15]. Third, the interconnect metal resistivity is also dependent on temperature. For example, the resistivity of copper increases by 39% from 1.72$\mu\Omega$-cm at 20°C to 2.39$\mu\Omega$-cm at 120°C. Higher resistivity causes longer interconnect $RC$ delay, and hence performance degradation. Last, but not least, reliability is strongly related to temperature. A first order model for the impact of temperature on reliability is the Arrhenius equation:

$$MTF = MTF_0 \; exp(E_a / k_b T)$$

where $T$ is operating temperature, $MTF_0$ is mean time to failure at a specified reference temperature, $E_a$ is the activation energy of the failure, $k_b$ is the Boltzmann constant. A well-known example of microelectronics reliability problems is the interconnect electromigration phenomenon. It is obvious from the Arrhenius equation that increasing the temperature

*This paper is an updated and extended version of a paper appearing in the 41st Design Automation Conference (DAC), San Diego, CA, June 2004.

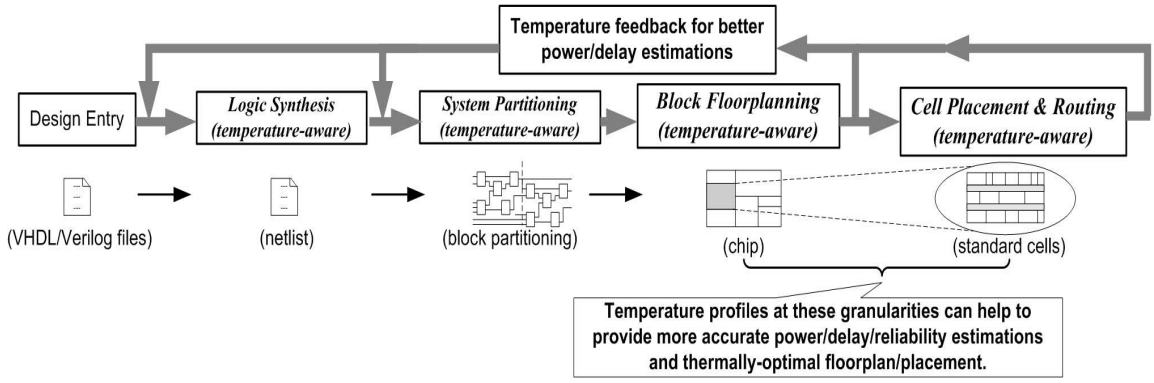



**Figure 1. An example of temperature-aware ASIC design flow.**

will exponentially decrease the mean time to failure, hence the life time. In summary: for future designs, higher operating temperature will have significant negative impacts on design aspects such as performance, power consumption, and reliability.

Based on the above facts, thermal design has become one of the major challenges for the CAD community in sub-100nm designs such as microprocessors, ASICs or Systems-on-a-Chip (SoC). Existing design methodologies typically use worst-case or room temperature when needed. This can lead to significant estimation errors and hence incorrect design decisions and longer design convergence time, as can be seen from the case study in Section 5. Therefore, it is crucial to find a way to properly address the temperature-related aspects of the design flow, and use temperature upfront as a guideline for design.

This paper is organized as follows. Section 2 introduces the idea of *temperature-aware* design. Section 3 proposes a compact thermal model that can be integrated into CAD tools to achieve a temperature-aware design flow. Validation of the model is presented in Section 4. In Section 5, a microprocessor design case study using the compact thermal model shows the importance of using temperature as a guideline for design. Section 6 concludes the paper.

## 2. Temperature-Aware Design

In sub-100nm technologies, early accurate design estimation is key to high-level design convergence and should ensure careful consideration of deep submicron effects (including power, performance, reliability, etc.) [9]. Temperature plays an important role in early accurate estimations of power, performance and reliability. In addition, thermal effects are influenced by placement and routing; for example, putting two hot blocks adjacent to each other will exacerbate the hot spots, while surrounding a hot block by several colder blocks will actually help in cooling down the hot spot. Temperature should thus be included in the cost function in order to achieve optimal placement and routing in sub-100nm designs. Temperature can also affect manufacturability in terms of packaging and choices of process if the design is thermally limited. Fig. 1 shows a simplified ASIC design flow adapted to become temperature-aware. Temperature profiles are needed at both functional-block level and standard-cell level during the ASIC design flow. Similar arguments also apply to microprocessor and SoC design flows.

From above, we see that it is very important to be able to estimate temperature at different granularities and at different design stages, especially early in the design flow. The estimated temperature can then be used to perform power, performance and reliability analyses, together with placement, packaging design, etc. As a result, all the decisions use temperature as a guideline and the design is intrinsically thermally optimized and free from thermal limitations. We call this type of design methodology *temperature-aware* design. The idea of temperature-aware design is unique because operating temperature is properly considered during the *entire* design flow instead of being determined only after the fact at the end of the design flow. There are a few examples of previous work about temperature-related design—for example, in [21], the authors present a design flow from digital simulations to a thermal map at the end of the design. This work is useful, but the design flow therein cannot be termed as a proper temperature-aware design since none of the intermediate design stages have closely considered temperature-related issues such as power or performance estimations, placement, thermal analysis, etc. Thus the design decisions of these stages are not optimized, and the design has to restart from the beginning if it turns out to be thermally limited.

## 3. A Compact Thermal Model

The key element for a temperature-ware design methodology is a thermal model to estimate operating temperatures. Fig. 2 shows how a thermal model helps to close the loop for accurate power, performance and reliability estimations. For example, the power model first provides estimated power to the thermal model. The thermal model in turn provides estimated temperature to the power model, and so on. After a few iterations, both power and temperature estimations converge, at that point, temperature-aware power estimation is achieved. Similarly, temperature-aware performance and reliability estimations can also be achieved.

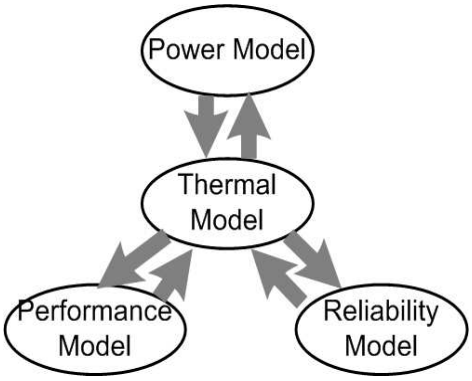



**Figure 2. Interactions among thermal model and power, performance and reliability models.**

There are a number of existing thermal models for different parts of a microelectronic design. For example, our previous work [18] [17] presents a dynamic compact thermal model, *HotSpot*, only at the microarchitecture level. [24] presents a chip-level thermal model based on full-chip layout. In [10], the authors present package thermal models. In [3], the authors present a thermal modeling approach based on analytical solutions of heat transfer equations, and the model is mainly focused at device level. None of these thermal models have the flexibility to model temperature at arbitrary granularities. Some of them are also computationally intensive. Thus, they are not completely suitable for temperature-aware design. To fulfill the requirements of a temperature-aware design, the thermal model has to be able to provide temperatures at different granularities (circuit structures, standard cells, functional unit blocks, etc.), and at different levels (silicon surface, interconnect, package, etc.). The model also needs to be computationally efficient to avoid time-consuming calculations during high-level, prior-layout design stages. In some cases, the model should be able to also model transient temperature changes. Of course, the model needs to be reasonably accurate to provide useful temperature estimates.

In this paper, we propose a *compact* thermal model that meets all the above requirements and can be integrated into existing CAD tools to achieve temperature-aware design. This compact thermal model is an extended version of *HotSpot*, which was proposed in [18] and [17]. Before moving into modeling details, it is useful to notice that this compact thermal model is a general model and therefore can be applied to different contexts. For example, dynamic thermal management (DTM) is an active research area in computer architecture community[4]. In this context, although at run time, only thermal sensors and DTM methods are needed in order to implement DTM techniques, the thermal model is still very attractive to computer architects, because it helps to simulate and explore different DTM methods. As another example, in the design automation context, we have already argued in Section 1 and Section 2 that a thermal model is needed to guide CAD, choices in process, circuit style, packaging, etc. People from CAD community and industry may consider the thermal model attractive. In this paper, we are interested in the latter context.

### 3.1. Model Overview

There is a well-known duality between heat transfer and electrical phenomena, as shown in Table 1. In this duality, heat flow that passes through a thermal resistance is analogous to electrical current; temperature difference is analogous to voltage. Similar to an electrical capacitor that accumulates electrical charges, thermal capacitance defines the capability of a structure to absorb heat. The rationale behind this duality is that electrical current and heat flow can be described by a similar set of differential equations (there is no thermal equivalent of electrical inductance though). The compact thermal model we

| Thermal quantity | unit | Electrical quantity | unit |
|---|---|---|---|
| $P$, Heat flow, power | $W$ | $I$, Current | $A$ |
| $T$, Temperature difference | $K$ | $V$, Voltage | $V$ |
| $R_{th}$, Thermal resistance | $K/W$ | $R$, Electrical resistance | $\Omega$ |
| $C_{th}$, Thermal capacitance | $J/K$ | $C$, Electrical capacitance | $F$ |

**Table 1. Duality between thermal and electrical quantities.**

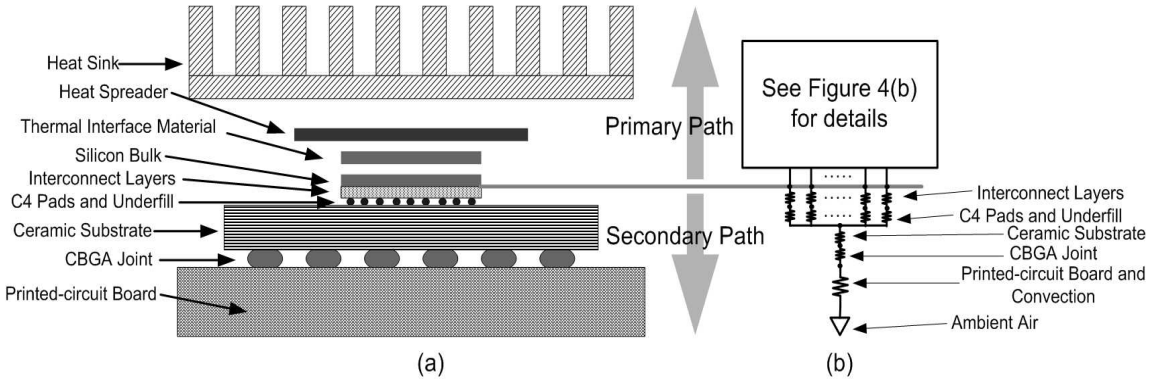



**Figure 3. (a) A typical flip-chip, CBGA package with heat sink (adapted from [13]). (b) Corresponding thermal circuit in our thermal model. Thermal capacitors connecting each node to ambient are not shown for clarity.**

propose is essentially a thermal RC circuit. Each node in the circuit corresponds to a block at the desired level of granularity. Heat dissipation of each block is modeled as a current source connected to the corresponding node. Solving this thermal RC circuit gives the temperatures of each node.

Fig. 3(a) shows a modern single-chip CBGA package [13]. Heat generated from the active silicon device layer is conducted through the silicon die to the thermal interface material, heat spreader and heat sink, then convectively removed to the ambient air. In addition to this primary heat transfer path, a secondary heat flow path exists from conduction through the interconnect layer, I/O pads, ceramic substrate, leads/balls to the printed-circuit board. Our compact thermal model models all these layers in both heat flow paths, with special emphasis on the primary path and the on-chip interconnect layer. This is because detailed temperature profiles of these parts are very important for temperature-aware design. In the model, we have also considered lateral heat flow within a layer by adding lateral thermal resistances to achieve greater accuracy of temperature estimation. Fig. 3(b) shows the thermal RC circuit structure that corresponds to Fig. 3(a). Next, we present the modeling details of each layers along both heat flow paths.

### 3.2. Primary Heat Flow Path

Fig. 4(a) shows an example thermal circuit of a silicon die with only three microarchitecture blocks from our previous work [18]. One significant improvement of the compact thermal presented in this paper is that we extend the thermal model for the primary heat flow path in [18] by making the model grid-like, thus being able to model temperatures at arbitrary granularities. Fig. 4(b) shows our modeling approach with an example granularity of 3x3 grid cells. Each silicon grid cell can be of arbitrary aspect ratio and size, which are determined by the desired level of granularity. We also add to the model a layer of thermal interface material that is absent in [18] but does exist in real packages. Adding the thermal interface material brings the temperature gradient across silicon closer to real design measurements, because the thermal interface material usually has a lower thermal conductivity compared to silicon and metal heat spreader and heat sink. As another small change compared to previous work, the part of the heat spreader that is right under the interface material, as well as the interface material itself, are divided into the same number of grid cells as the silicon die in order to improve accuracy. Other parts in the primary heat flow path are modeled in a similar way as in [18]— the remaining part of the heat spreader is divided into four trapezoidal blocks. The heat sink is divided into five blocks: one corresponding to the area right under the heat

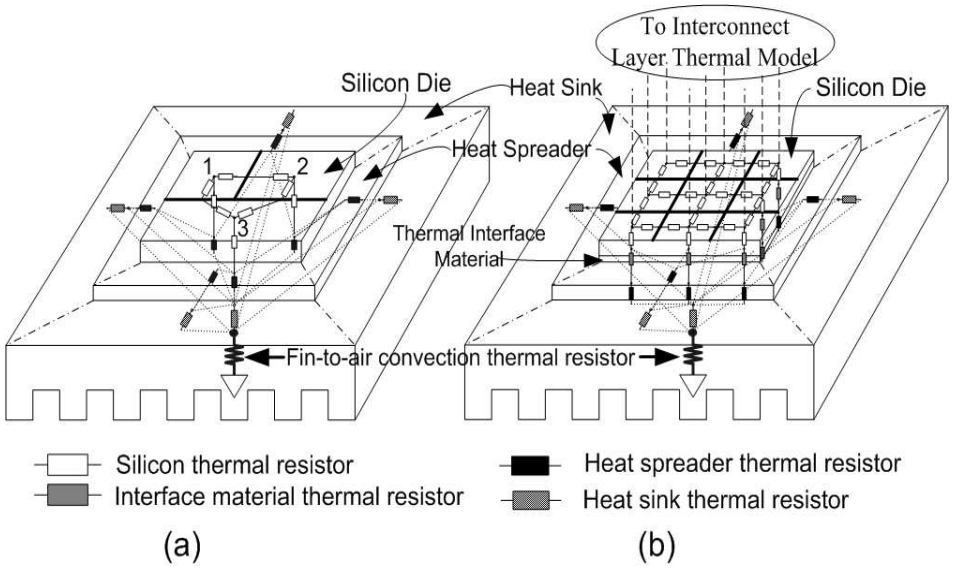



**Figure 4. (a) Thermal circuit of a silicon die with 3 microarchitecture blocks, adapted from [18]. (b) Thermal circuit of a silicon die with 3x3 grid cells, with thermal interface material, heat spreader and heat sink. (Thermal capacitors and heat sources are not shown for clarity.)**

spreader and four trapezoids for the periphery. Each grid cell maps to a node in the thermal circuit, and there are vertical and lateral thermal resistors connecting the nodes. Each node also has a thermal capacitor connected to the ambient. The power dissipated in each silicon grid cell is modeled as a "current source" connected to the corresponding node. The package-to-air thermal resistor is calculated from specific heat-sink configurations and ambient conditions [14]. [1]

The derivation of thermal resistors and capacitors is mainly based on the fact that vertical thermal resistors are proportional to the thickness of the material and inversely proportional to the cross-sectional areas across which the heat is being transferred: $R_{vertical} = t/(k{\cdot}A)$, where $k$ is thermal conductivity of the material. Lateral thermal resistors are essentially the constriction or spreading thermal resistances for heat to diffuse laterally from one block into other parts of the material, and are calculated by a method described in [11] and [19]. Thermal capacitors, on the other hand, are proportional to both thickness and area: $C = \alpha{\cdot}c_p{\cdot}\rho{\cdot}t{\cdot}A$, where $c_p$ and $\rho$ are the specific heat and density of the material, respectively. Notice that the thermal capacitor used here is a single-lumped model instead of a more detailed distributed model. Therefore, a scaling factor $\alpha \simeq 0.5$ for thermal capacitances is used to correct this, similar to what was derived analytically in [2] for single-lumped vs. distributed electrical RC circuits. It is useful to note that the derivation methods of thermal Rs and Cs for the primary heat flow path allows us to use the same modeling approach at different levels of granularity.

### 3.3. Secondary Heat Flow Path

The secondary heat transfer path helps to remove a non-negligible amount of total generated heat (up to 30%). Neglecting this heat transfer path will lead to inaccurate temperature predictions. In addition, in order to model temperature-affected on-chip interconnect delay and life time, the thermal model of the interconnect metal layers is needed, which is part of the secondary heat transfer path. In this paper, the thermal model for the secondary heat flow path is divided into two parts: one corresponding to the interconnect layers, and the other for the path from the I/O pads to the printed-circuit board (see Fig. 3(a) and (b)).

[1] We have developed a stand-alone tool to do this job, and it will be integrated with the thermal model in the near future, see [14] for details.

| Megacell's Name | $k_r$ | $N$ | $p_r$ | Megacell's Name | $k_r$ | $N$ | $p_r$ |
|---|---|---|---|---|---|---|---|
| Instruction Cache | 4.12 | 380000 | 0.20 | Instr. Fetch Address | 3.20 | 16500 | 0.60 |
| Instruction Cache Tags | 3.80 | 18000 | 0.47 | Instr. Fetch Datapath | 3.20 | 13800 | 0.60 |
| Data Cache | 4.12 | 350000 | 0.20 | Instr. Fetch Control | 3.20 | 9500 | 0.60 |
| Data Cache Tags | 3.80 | 25500 | 0.47 | Address Queue | 3.20 | 22000 | 0.60 |
| TLB | 3.80 | 22400 | 0.35 | Instr. Decode and Reg. Ren. | 3.20 | 45300 | 0.60 |
| Secondary Cache Control | 3.20 | 15700 | 0.60 | Integer Datapath | 3.20 | 43800 | 0.60 |
| External Interface | 3.20 | 18400 | 0.60 | Integer Queue | 3.20 | 19700 | 0.60 |
| System Interface Buffers | 3.20 | 22600 | 0.60 | Floating Point Datapath | 3.20 | 32600 | 0.60 |
| Free List | 3.20 | 9800 | 0.60 | Floating Point Queue | 3.20 | 51000 | 0.60 |
| Graduation Unit | 3.20 | 26300 | 0.60 | Floating Point Multiplier | 3.20 | 19300 | 0.60 |
| Die-level Equivalent | 3.79 | 1162200 | 0.34 | Logic Core Equivalent | 3.23 | 388700 | 0.58 |

**Table 2. Rent's Rule parameters for a RISC microprocessor [23], data are extracted from [26]. The last row shows the equivalent Rent's Rule parameters of the whole die and the logic core (excluding the cache memories).**

#### 3.3.1 Interconnect Thermal Model

There are two aspects considered in the interconnect thermal model: First, the self-heating power of an individual metal wire, which is

$$P_{self} = I^2 \cdot R = I^2 \cdot \rho_m \cdot l / A_m$$

where $I$ is the current flowing through the wire, $R = \rho_m \cdot l / A_m$ is the electrical resistance, $\rho_m$ is the metal resistivity (which is temperature dependent), $l$ and $A_m$ are the length and cross-sectional area of the individual wire. Because the interconnect thermal model needs to *predict* wire temperatures before physical layout is available, this means the model has to be able to predict the average wire length and average self-heating current (RMS current) for wires in each metal layer. It is also important to notice that because the routing methodology and self-heating mechanism are significantly different for the signal interconnects and the power distribution network, the ways of predicting average wire length and self-heating current are also not the same for signal interconnects and power supply distribution network. Therefore, we treat signal wires and power supply wires separately in this section. The second aspect is to find the equivalent *thermal* resistance for each metal wire and its surrounding inter-layer dielectric. Vias also play an important role in heat transfer among different metal layers, and therefore should also be included in the model.

**Average interconnect length in each metal layer.** Here, signal interconnects are considered first. We predict the average signal interconnect length in each metal layer by adopting and extending the statistical *a priori* wire-length distribution model presented in [6], which improves an earlier wire-length distribution model [7]. It is important to note that an interconnect thermal model at high levels of abstraction strongly depends on the *a priori* wire-length distribution model, and hence is limited by the accuracy and efficiency of the wire-length distribution model.

The model in [6] is based on Rent's Rule:

$$T = k_r N^{p_r}$$

where $k_r$ and $p_r$ are Rent's Rule parameters, $N$ is the number of gates in a circuit, $T$ is the predicted number of I/O terminal in the circuit. Table 2, which is extracted from [26], shows typical values of $N$, $k_r$ and $p_r$ for a RISC microprocessor [23]. For a given microprocessor family, the Rent's Rule parameters of each circuit block (mega-cell) tend to remain the same over generations due to the recursive application of Rent's Rule throughout the entire monolithic circuit block [6]. Therefore, we can assume the same parameters, $k_r$ and $p_r$, can be used in a design at future technology nodes for the same microprocessor family, although the number of gates $N$ increases. Sometimes, wire-length distribution is needed at a higher abstraction level, e.g. at die level, thus equivalent Rent's Rule parameters need to be calculated at that level. Using the method in [25], one can calculate the equivalent Rent's Rule parameters for the whole processor and the core of the processor (excluding the on-chip cache memories), as also shown in Table 2.

Three wire-length regions are considered in [6]—local, semi-global and global. The model predicts the number of wires of any specific length, which is called the interconnect density function $i(l)$, where $l$ is the wire length in gate pitches. Fig. 5 shows an example wire-length distribution based on ITRS data [1] for high-performance designs at the 45nm technology

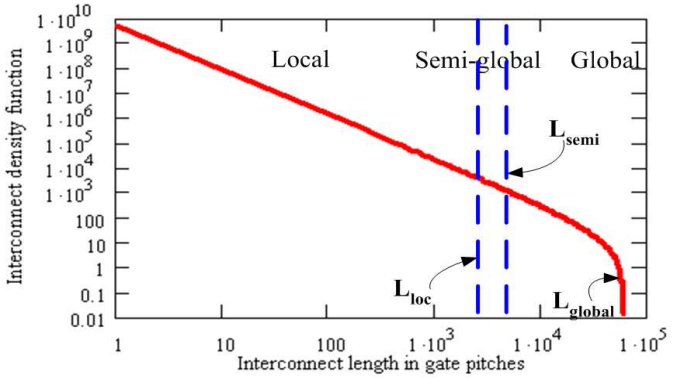



**Figure 5. An example of wire-length distribution at 45nm technology node**

node, where $L_{loc}$, $L_{semi}$, $L_{glob}$ are maximum local, semi-global and global wire lengths, respectively. Using the interconnect density function $i(l)$, one can calculate the average length and number of wiring nets for each region. For example, for the semi-global region:

$$l_{semi} = \chi \text{ f.o.} \frac{\int_{L_{loc}}^{L_{semi}} i(l) \cdot l \, dl}{\int_{L_{loc}}^{L_{semi}} i(l) \, dl}$$

$$n_{semi} = \frac{1}{\text{f.o.}} \int_{L_{loc}}^{L_{semi}} i(l) \, dl$$

where $\chi$ is the correction factor that converts the point-to-point interconnect length to wiring net length (using a linear net model $\chi = 4/(\text{f.o.} + 3)$ ), f.o. is the average number of fan-outs per wiring net. More details can be found in [6].

However, there is no wire-length distribution information regarding each metal layer when using this three-region division method in [6]. For the interconnect compact thermal model, we need the wire-length distribution predictions of every metal layer. Because of the predominant usage of Manhattan routing method, at least two metal layers are needed to route one wiring net—one layer for horizontal routing, the other for vertical routing. In our paper, we determine which pair of metal layers at which each wiring net resides by filling every two metal layers with wiring nets, starting from the shortest wiring nets. That is, the shortest wiring nets of the wire-length distribution in Fig. 5 are assigned to Metal 1 and 2. Once the first two metal layers are filled, we proceed to Metal 3 and Metal 4, and so on, until all the wiring nets are assigned to their corresponding pair of metal layers. In this way, we are also able to estimate the number of metal layers needed for a design. As illustrated in Fig. 6, assume the length of the shortest and longest point-to-point interconnects that can be assigned to a pair of metal layers are $L_{min}$ and $L_{max}$ in gate pitches, we can find the average length and total number of wiring nets by

$$l_{avg} = \chi \text{ f.o.} \frac{\int_{L_{min}}^{L_{max}} i(l) \cdot l \, dl}{\int_{L_{min}}^{L_{max}} i(l) \, dl}$$

$$n_{total} = \frac{1}{\text{f.o.}} \int_{L_{min}}^{L_{max}} i(l) \, dl$$

Furthermore, by assuming the routing structure of Fig. 6, where $M$ is the number of signal wires between two power rails and $Sp$ is ratio of the space between every two signal wires to $l_{avg}$ (both $M$ and $Sp$ are design parameters and are tunable by the user), we have the following relation:

$$n_{total} \cdot (Sp + 1) \cdot l_{avg} \cdot \frac{M + 1}{M} \cdot p = 2 \cdot Area$$

where $p$ is the wire pitch of a metal layer, and $2 \cdot Area$ is the available routing area for the interested pair of metal layers. Using this relation, and starting at Metal 1 and Metal 2 with $L_{min} = 1$, we are able to solve for $L_{max}$ and $L_{min}$ for each pair of metal layers. An example metal layer assignment for the interconnect distribution of Fig. 5 is shown in Fig. 7(a). Another way to assign signal wiring nets to different layers is to calculate the number of metal layers needed for each of

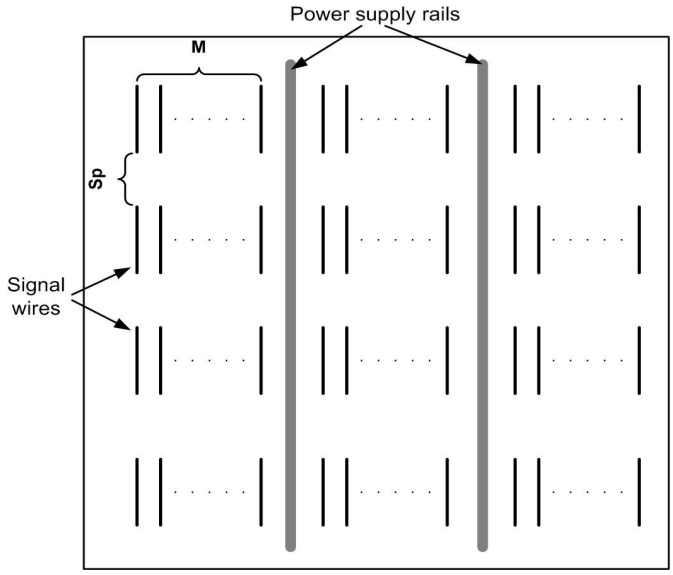



**Figure 6. A scheme to assign signal interconnects to metal layers. $M$ is the number of signal wires between two power rails and $Sp$ is ratio of the space between every two signal wires to average signal wire length of that metal layer.**

the three regions, namely, local, semi-global and global region in [6]. The resultant metal layer assignment is shown in Fig. 7(b). As can be seen, the results in Fig. 7(a) and (b) are similar, but Fig. 7(a) provides detailed metal layer assignment estimations for every two metal layers without considering the three regions, while information provided in Fig. 7(b) is more coarse. Therefore, we prefer the approach used in Fig. 7(a). On the other hand, if the total number of metal layers are fixed, parameters $Sp$ and $M$ can be adjusted accordingly to fit all the signal interconnects into the metal layers.

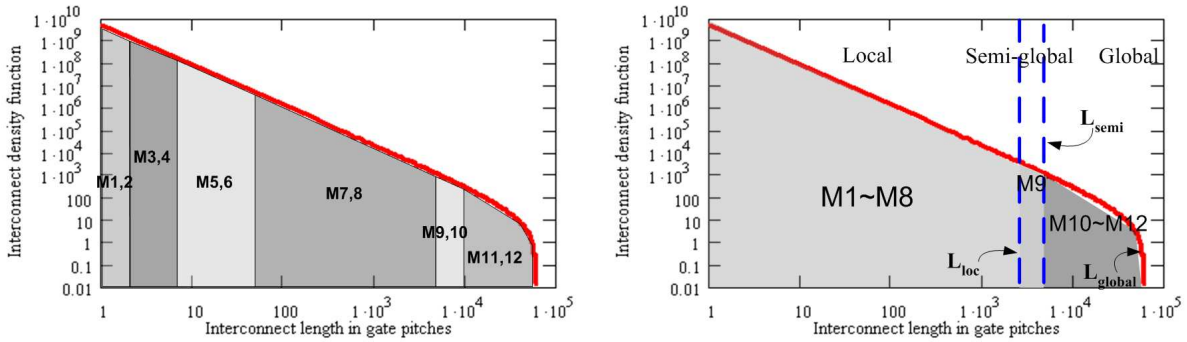
So far, we have considered average signal interconnect length in each metal layer. We also need to find average wire length for power supply network, which is usually grid-like. This is relatively simple—we only need to find the length of the power grid section in each metal layer. The assumption here is that the power power grid for each metal layer is uniformly distributed. This is a reasonable assumption during earlier high-level design stages.

**Average interconnect self-heating current in each metal layer.** Again, we first consider the self-heating current of signal interconnects in each metal layer. For each switching event, half of the energy drawn from the power supply is dissipated in the form of heat on the charging/discharging transistor and on the output signal interconnect. The average current flow through the interconnect during a switching event can be solved from the following equation:

$$I_{avg}^2(R_{tr} + R_{wire})t_d = \frac{1}{2}\alpha\, C_L V_{dd}^2$$

where $I_{avg}$ is the average self-heating current per wire in each metal layer. $R_{tr}$ is the on-resistance of the transistor, $R_{wire}$ is the wire resistance, $\alpha$ is the switching activity factor, $C_L$ is the load capacitance, and $t_d$ is the delay of the switching event. There is an issue here needed to be taken into account—for long interconnects, repeaters are inserted in order to achieve minimum delay. The critical wire-length between repeaters ($L_{crit}$), the delay for one section of buffered interconnect ($\tau_{crit}$), the optimal number of repeaters ($Nr_{crit}$) and the optimal size of repeaters ($s_{crit}$) for interconnects in each region can be found using the repeater insertion model proposed in [12]. The calculations of $R_{tr}$, $R_{wire}$, $C_L$ and $t_d$ are different for wires without or with repeaters inserted—the wire length is either the total wiring net length or the length of a wire section between repeaters; the driving and load gates are either gates with average transistor size or repeaters with size of $s_{crit}$. Finally, the delay of the switching event, $t_d$, can be approximated as as $\tau_{crit}$ for interconnects with repeaters, and $clock\_cycle\_time/logic\_depth$ for interconnects without repeaters.

To calculate average currents for power supply grid sections, there are two methods. One method is to build a grid-like resistive network model for $V_{dd}$ and $GND$, somewhat resembling the thermal circuit used for modeling the primary heat flow path in Section 3.2. Each resistor connecting two nodes in the same metal layer is now the electrical resistance of one




(a) (b)

**Figure 7. (a)—Metal layer assignment by filling every two metal layers with signal wires, starting from Metal 1 and Metal 2; (b)—Metal layer assignment by calculating number of metal layers needed for each of the three regions (local, semi-global and global). The former method is superior to the latter one by providing more detailed metal layer assignment information.**

power supply grid section. Resistors connecting power grid nodes of different metal layers represent the vias. The topology of the network is obtained by knowing the pitch between power rails in each metal layer, average length and number of power grid sections between power grid. Next, by applying currents to the top-layer nodes that are at the C4 pads sites, the resistive network is solved to find the average self-heating current of the power grid in each metal layer. The other method to calculate average self-heating current of power grid section in a metal layer is quite straightforward—we can simply divide the total current delivered to a metal layer by the number of power grid sections. This method is suitable for high-level design stages, but it is not as accurate as the first method.

**Total interconnect self-heating power in each metal layer.** With all the above information of average interconnect length and average current in each layer (for both signal interconnects and power grid sections), we calculate the average self-heating power per interconnect in each metal layer:

$$P_{self} = I_{avg}^2 \cdot R_{wire} = I_{avg}^2 \cdot \rho_m \frac{l_{wire}}{A_{wire}}$$

where $A_{wire}$ and $l_{wire}$ are the cross-sectional area and the average length of signal interconnects or power grid sections in each metal layer, respectively.

Last, we calculate the self-heating power for each metal layer of the circuit. "Circuit" here means a circuit block at the desired level of granularity. If, for example, we calculate the self-heating power of metal 10 as:

$$P_{self\_m10} = P_{self\_glob\_sig} \cdot n_{sig\_m10} + P_{self\_glob\_pwr\_net} \cdot n_{pwr\_net\_m10}$$

So far, we are done with the first aspect of interconnect thermal modeling—self-heating power calculation. Next, we calculate the equivalent thermal resistance of wires and the surrounding dielectric, together with the thermal resistance of vias.

**Equivalent thermal resistance of wires/dielectric and vias.** We first start from a simplistic case. Fig. 8(a) shows a single interconnect surrounded by inter-layer dielectric. On top of and below it are interconnects in neighboring layers. $d$ is the thickness of the inter-layer dielectric, $W$ and $H$ are width and height of the interconnect cross section. We try to find the thermal resistor associated with each wire $R_0$; The rectangular cross section of the wire can be approximated by a circle of the same area. Heat is spreading from the wire into the dielectric, the isothermal surface is a cylindrical surface marked by the dashed circle. The equivalent resistance $R_0$ has to take into account the top half volume of the shaded cylinder.

We first calculate the thermal resistance of the dark shaded slice of inter-layer dielectric shown in Fig. 8(a). It can be written in the form of the integral

$$dR_0 = \int_0^{d/2} \frac{1}{k_{ins}} \frac{dx}{(r+x)d\phi \cdot l} = \frac{1}{k_{ins} \cdot l \cdot d\phi} \ln(\frac{d+2r}{2r})$$

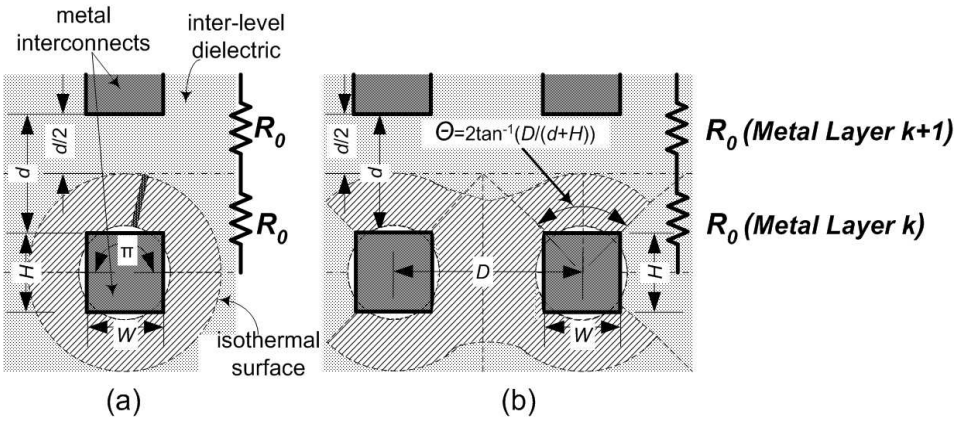



**Figure 8. Interconnect structures for calculating equivalent thermal resistance of wires and surrounding dielectric—(a) stacked single wires (b) real wire structure with multiple wires in each layer.**

where $x$ is the integral variable, $k_{ins}$ is the thermal conductivity of the inter-layer dielectric, $\phi$ is the angle of the slice, $r = \sqrt{WH/\pi}$ is the equivalent radius of the wire, and $l$ is the length of the wire,

If we define thermal conductance $G_0$ as the reciprocal of thermal resistance $R_0$, we have

$$dG_0 = k_{ins} \cdot l \cdot d\phi \cdot \frac{1}{\ln(\frac{d+2r}{2r})} \Rightarrow G_0 = \int_0^{\pi} dG_0 = \pi \cdot k_{ins} \cdot l \cdot \frac{1}{\ln(\frac{d+2r}{2r})}$$

so the total equivalent thermal resistance is

$$R_0 = \frac{1}{G_0} = \ln(\frac{d+2r}{2r})/(\pi \cdot k_{ins} \cdot l).$$

Fig. 8(b) shows the real case: multiple wires are in the same layer. The wire pitch is denoted by $D$. A phenomenon called thermal coupling happens when neighboring wires dissipate power at the same time. Thermal coupling leads to less effective heat conducting area and change the shape of the isothermal surface. The actual isothermal surface is shown by the dashed area in the figure. In this case, each wire's effective heat spreading angle is approximately $\theta = 2 \cdot \arctan(D/(d+H))$, and the corresponding equivalent thermal resistance for each wire becomes:

$$R_0 = \ln(\frac{d+2r}{2r})/(\theta \cdot k_{ins} \cdot l)$$

Inter-layer heat transfer also happens through vias. A simplistic approximation of the number of vias for signal interconnect would be to assume that each wiring net has two vias, one connected to the upper metal layer, and another one connected to the lower metal layer. A more accurate approximation is to assume each wiring net has $(2 \cdot \text{f.o.} + 2)$ vias, where f.o. is the average fan out number of each gate. As illustrated in Fig. 9, $(\text{f.o.} + 2)$ vias are at the ends of the wiring net and connecting the wiring net to lower metal layers and eventually to the device layer at the silicon surface. The other f.o. vias are used to aid the routing of the wiring net within the pair of metal layers in which the wiring net resides. For power supply grid, because the wires are usually wider than signal wires, designers usually put an array of vias at the intersection of two power rails at different metal layers. As illustrated in Fig. 10, the number of vias at on intersection of power rails can be estimated by

$$\frac{1}{4}\left(\frac{W_{wire}}{W_{via}} - 1\right)^2$$

where $W_{wire}$ and $W_{via}$ are the widths of the power wire and the via, respectively. The thermal resistance of each via can be calculated by $R_{via} = t_v/(k_v A_v)$, where $k_v$ is thermal conductivity of via-filling material. $t_v$ and $A_v$ are thickness and cross-sectional area of the via.

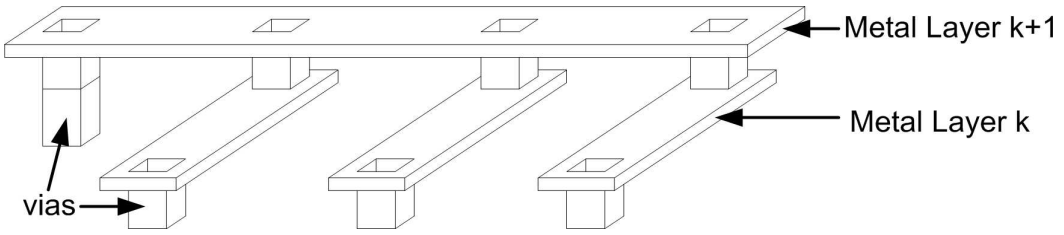



**Figure 9. Estimating number of vias for signal interconnects — A wiring net with fan out 3 is shown in this figure. The number of vias is $(2 \cdot$f.o. $+ 2)$.**

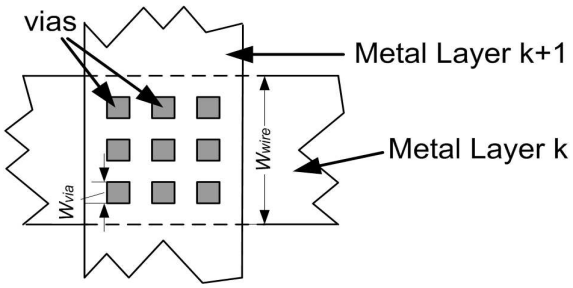



**Figure 10. Estimating number of vias for power supply wires — An array of vias are put in the intersection of power wires at two metal layers. $W_{wire}$ and $W_{via}$ are the widths of the power wire and the via, respectively.**

All thermal resistors of wires and vias between two metal layers can be considered parallel to each other. Thus, combining all the thermal resistors between two metal layers of the circuit, we obtain: (use Metal 4 and Metal 5 as an example)

$$R_{m4m5} = \left( \frac{R_{0\_sig\_m4}}{n_{m4\_sig}} || \frac{R_{0\_pwr\_net\_m4}}{n_{m4\_pwr\_net}} + \frac{R_{0\_sig\_m5}}{n_{m5\_sig}} || \frac{R_{0\_pwr\_net\_m5}}{n_{m5\_pwr\_net}} \right) || \frac{R_{via}}{n_{m4,5\_sig\_via} + n_{m4,5\_pwr\_net\_via}}$$

We are almost done with the interconnect thermal modeling. One last step is to stack the thermal resistances for each layer to construct the whole thermal circuit for all interconnect layers. Currently, the interconnect thermal model doesn't include thermal capacitors, but these will be added using the methods presented in Section 3.2 and in this section. Designers are usually more interested in steady-state interconnect temperatures for electromigration and power-grid $IR$ drop analysis.

### 3.3.2 Thermal Model from I/O Pads to PCB

Our model for the heat flow path from I/O pads to PCB consists of a series of thermal RC pairs, each of which represents the thermal resistance and capacitance of pad-bumps/underfill, ceramic substrate, ball/lead array, and PCB convection (see Fig. 3(b)). Thermal Rs and Cs are calculated in a similar way as in Section 3.2. The Rs and Cs for the pads/underfill level are modeled at the desired level of granularity. One end for each of these Rs for pads/underfill is connected to the interconnect-level thermal model, the other end is joined into one node, which is then connected to the RC pair representing ceramic substrate, and so on.

## 3.4 Simulation Speed for the Compact Thermal Model

So far, we have shown all the parts of the compact thermal model. The model is derived in a straightforward way and is computationally efficient. To make the calculations faster, the we use First Order Difference Equations for the RC network instead of differential equations and uses small time steps to calculate the transient change in temperature. A typical node is

shown in Fig. 11 and the equations related to it are:

$$P_{i,j} = \frac{T_{i,j} - T_{i-1,j}}{R_1} + \frac{T_{i,j} - T_{i,j+1}}{R_2} + \frac{T_{i,j} - T_{i+1,j}}{R_3} + \frac{T_{i,j} - T_{i,j-1}}{R_4} + \frac{T_{i,j} - T_{bottom}}{R_{bottom}} + \frac{C_{i,j}\Delta T_{i,j}}{\Delta t}$$

$$\Delta T_{i,j} = \frac{P_{i,j}\Delta t}{C_{i,j}} + \frac{\Delta t}{C_{i,j}}\left(\frac{T_{i-1,j}}{R_1} + \frac{T_{i,j+1}}{R_2} + \frac{T_{i+1,j}}{R_3} + \frac{T_{i,j-1}}{R_4} + \frac{T_{bottom}}{R_{bottom}}\right) - \frac{\Delta t}{C_{i,j}}\frac{T_{i,j}}{G_{T_{i,j}}}$$

$$G_{T_{i,j}} = \frac{1}{R_1} + \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_{bottom}}$$

The second equation is the main difference equation that is used to calculate the change in temperature after each run. After calculating the change in temperature, $\Delta T_{i,j}$ after time $\Delta t$, it is added back to $T_{i,j}$.

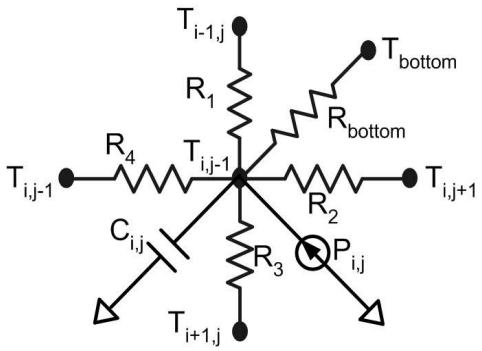



**Figure 11. A typical node in the thermal RC network. Its neighboring nodes are shown, together with thermal resistors, capacitor and heat source that are connected to that node.**

Now this equation uses the temperatures of the other nodes. Instead of solving them simultaneously, the temperatures of the other nodes are kept constant (this works since the time steps are really small and the error is negligible). This means, on every iteration, only those nodes that are not adjacent to each other are being calculated. This is implemented by a Red/Black tree, where each Red node is surrounded by 4 Black nodes. The vertical node $T_{Bottom}$ lies on next layer of the package. This approach is taken to minimize memory. Alternative approach would be to have two copies of the nodes and updating them after every run. The number of iterations necessary for every time step can be varied until the appropriate number is found which results in accurate values.

Computing steady-state temperatures, however, is different from computing transient temperatures. Usually, one can first find the inversion of the conductance matrix, then multiply it by the power vector and get the steady-state solutions. As the number of grids become more and more with increasing granularity, the operation of matrix inversion becomes computationally intensive. Another way to compute the steady-state solutions is to solve the above difference equations for a very long time interval, hence the results approach the steady-state values. Both methods are very time-consuming. Our solution is first using the matrix inversion method at a much coarser granularity to find approximate steady-state temperatures, and set these temperatures as the initial values for the difference equations, then solve the difference equations for a very short time interval to reach more accurate steady-state temperature estimations. The steady-state temperature computation times for different granularities shown in our DAC'04 paper [8] are based on this method.

Table 3 shows the computation times of our thermal model to obtain transient solutions for several different simulation time intervals at different granularities. The initial temperatures are all set to room temperature. For short transient simulation time intervals, the computational overheads are in the the order of mili-seconds. This computational efficiency means there is little computation overhead for existing design methodologies to incorporate the compact thermal model for temperature-aware design. For longer simulation time interval, we can use a method similar to the one mentioned above for calculating steady-state temperatures, i.e. setting initial temperatures of the solver to the ones obtained from a coarser granularity, then calculating actual transient temperatures at the right granularity. By doing this, the computation time for longer simulation time interval can also be in the order of mili-seconds.

| # of grid cells | 33.3$\mu$s simulation time interval | 3.33ms simulation time interval |
|---|---|---|
| 5x5 | 0.4ms | 0.4ms |
| 50x50 | 70ms | 80ms |
| 100x100 | 240ms | 590ms |
| 160x160 | 620ms | 3.2s |

**Table 3. Computation times of our model for different transient simulated time intervals.**

| | steady-state | transient |
|---|---|---|
| average absolute error | $1.46\%$ | $2.26\%$ |
| error range | $-3.35\%$—$+4.75\%$ | $-7.0\%$—$+6.7\%$ |

**Table 4. Percentage error values for primary heat flow path validations**

## 4. Model Validation

We validate the compact thermal model in the same sequence we derive it—primary heat flow path first, followed by the secondary heat flow path.

### 4.1 Primary Heat Flow Path

This part of the model is validated against a commercial thermal test chip [20]. The thermal test chip has a 9x9 grid of power dissipators, which can be turned on or off individually, with an embedded thermal sensor for each grid cell. The test chip can measure both steady-state and transient temperatures for each of the grid cells. We built the same 9x9 grid-like chip structure in our thermal model. In this experiment, we neglected the secondary heat flow path, because the test chip is wire bonded and plugged in a plastic socket that has very low thermal conductivity. We then turned on sets of power dissipators in the test chip and assigned the same power values at the same locations in our thermal model.

Fig. 12(I) shows the steady-state thermal plots using measurements from the test chip and results from our thermal model. Transient temperature data from the thermal model are also compared with the test chip transient measurements, as shown in Fig. 12(II). Table 4 shows the percentage error values, which are calculated by $(T_{model} - T_{chip})/(T_{chip} - T_{ambient})$. The power density in this experiment is $50\mathrm{W/cm}^2$ in the heat dissipating area (the 3x3 lower-right corner). As can be seen, our thermal model of the primary heat flow path is reasonably accurate, with the worst case error values for steady-state temperatures and transient temperatures less than 5% and 7%, respectively.

### 4.2 Secondary Heat Flow Path

For validation of the interconnect thermal model, we compare our model to the finite-element models (FEM) published in [16]. There the authors build two interconnect test structures in FEM analysis software: one with individual metal wires on top of each other (this corresponds to the case of Fig. 8(a)); and the other one with multiple metal wires within each layer(this corresponds to the case of Fig. 8(b)). Both test structures have four metal layers at 0.6$\mu$m technology. We use exactly the same settings for our interconnect thermal model as in [16], and perform the same two experiments—1) for the stacked single-wire test structure, apply different power for each wire and obtain the temperature rise with respect to ambient temperature; 2) for both test structures, apply different current density for each layer and obtain the temperature rise. Results are shown in Fig. 13(a) and (b). As can be seen, the results of our interconnect thermal model match FEM simulation results very well.

For validation of the thermal model from I/O pads to printed-circuit board, there is no straightforward existing data for comparison, but, based on the validation of other parts of the thermal model, we have enough confidence that our model for this part is reasonably accurate. A simple calculation using our model based on the thermal specifications of the PowerPC603 CBGA package [13] shows that about 17.5% of total heat is dissipated through the secondary heat flow path.

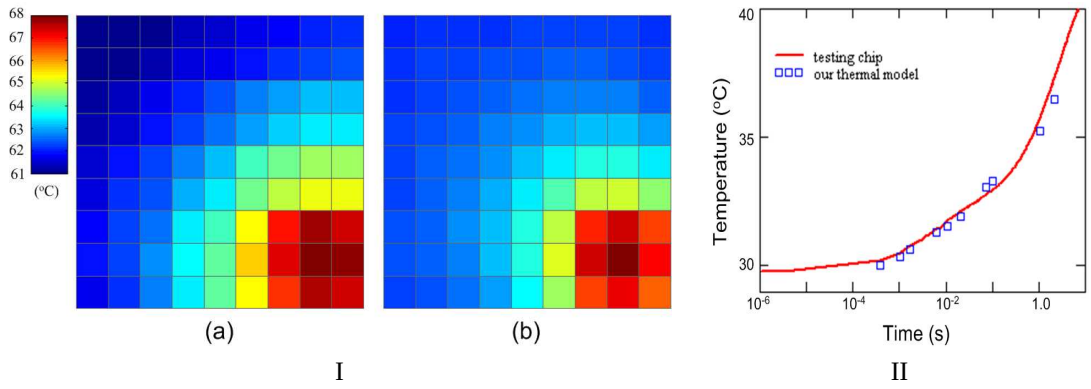



**Figure 12. (I)— Steady-state validation of the compact thermal model: (a) Test chip measurements (b) Results from the model with errors less than 5%. (II)— Transient validation of the compact thermal model. Percentage error is less than 7%. (Transient temperature response of one power dissipator is shown here.)**

| physical parameters | across die | L1 D-cache |
|---|---|---|
| number of transistors | 2200 million | 70 million |
| Rent's parameters | $p_r = 0.6$, $k_r = 4.0$ | $p_r = 0.6$, $k_r = 4.0$ |
| feature size | 45nm | 45nm |
| wiring levels | 12 | 12 |
| area | $3.10\text{cm}^2$ | $9.56\text{mm}^2$ |
| power dissipation | 218W | 60.9W |
| power density | $70.3\text{W/cm}^2$ | $637\text{W/cm}^2$ |

**Table 5. A microprocessor example—across-die vs. L1 D-cache (based on ITRS 45nm technology node [1]).**

## 5. A Case Study

In this section, we present a microprocessor design at a future 45nm technology node as a case study. This case study demonstrates the application of our compact thermal model and the importance of using temperature as a guideline during design. Technology specifications used in this case study are shown in Table 5, the second column of which is taken from [1]. We use an on-die level-one (L1) data cache approximating that of the Alpha 21364 processor scaled to 45nm technology node as an example of localized heating. The scaling process is a linear scaling from known data at 130nm technology, with proper considerations for leakage power and area. Power consumption values of functional units are extracted from a technology-scaled version of Wattch [5].

We first show that at the die level, using estimated temperature from our thermal model offers more accurate design estimations for power, delay and interconnect reliability than just using room temperature or worst-case temperature as can be seen from the results presented in Table 6. Simply using room temperature or worst-case temperature yields more errors, therefore leading to possibly incorrect design decisions and longer design convergence time.

The second experiment is to show the importance of being able to estimate temperatures at different granularities. This is because different stages of the design process need different granularities of power, delay or reliability estimations, hence different granularities of temperature estimations. By changing the number of grid cells, i.e. the level of granularity in our thermal model, we can calculate the average temperature across the die, average temperature of the L1 data cache, and max/min temperatures within the L1 D-cache. As can be seen in Table 7, a local hot spot like an L1 D-cache can have a significantly higher temperature than the average die temperature. Even within the L1 D-cache itself, there are also noticeable temperature gradients. Therefore, during the design of specific blocks like the L1 D-cache, using average die temperature yields inaccurate design estimates. From the last column of Table 7, we can also see the influence of number of grid cells

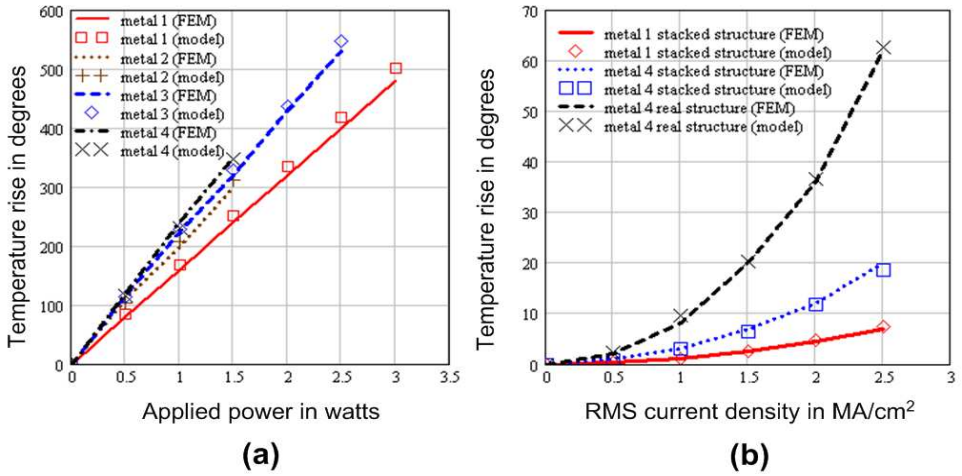



**Figure 13. Interconnect thermal model validation—FEM results (lines) from [16], and our thermal model results (markers): (a) stacked single wires—powers are applied to each wire (b) RMS current densities are applied to both test structures.**

| | model | room temp. | worst-case temp. |
|---|---|---|---|
| leakage power | 1.0 | 0.61 | 2.85 |
| delay | 1.0 | 0.83 | 1.25 |
| life time | 1.0 | 37.40 | 0.027 |

**Table 6. Temperature estimates using room temperature and worst-case temperature, normalized to the temperature estimates from the thermal model.**

on the accuracy of maximum L1 D-cache temperature predictions. Also we can see from Table 7 that at some point, further increasing the granularity can no longer improve the maximum temperature estimations in L1 D-cache. The minimum grid size needed in order to achieve the desired accuracy of temperature estimations depends on the applied power density to L1 D-cache and the equivalent thermal thickness, which is the thickness from the interested surface to the isothermal surface somewhere deep in the package and is usually on the order of hundreds of microns to millimeters for silicon. Analysis shows that the minimum grid size is on the order of the equivalent thermal thickness. For grid size smaller than that, the temperature gradient across the grid is "filtered" out and becomes negligible.

| # of grids (die) | die avg. T | D-cache avg. T | D-cache max T |
|---|---|---|---|
| 25x25 | 72.8 | 115.4 | 120.5 |
| 30x30 | 72.8 | 115.4 | 123.7 |
| 35x35 | 72.8 | 115.4 | 126.7 |
| 40x40 | 72.8 | 115.4 | 128.1 |
| 45x45 | 72.8 | 115.4 | 128.8 |
| 50x50 | 72.8 | 115.4 | 129.1 |
| 55x55 | 72.8 | 115.4 | 129.2 |

**Table 7. Temperatures at different levels of granularity (°C).**

As another example of how our compact thermal model can be applied, recent work on a leakage power simulator [22] uses our compact thermal model to predict operating temperature of the microprocessor and hence closes the loop of temperature and leakage power estimation.

## 6. Conclusion

We believe that thermal design will be one of the major challenges for the CAD community for sub-100nm designs. To address this challenge, we introduce the idea of *temperature-aware* design, which uses temperature as a guideline during the entire design flow. We also propose a compact thermal model for temperature-aware design. Results from our thermal model show that a temperature-aware methodology can provide more accurate design estimations, and therefore better design decisions and faster design convergence.

### Acknowledgments

This work is supported in part by the National Science Foundation under grant no. CCR-0133634, two grants from Intel MRL, and an Excellence Award from the Univ. of Virginia Fund for Excellence in Science and Technology.

## References

[1] The international technology roadmap for semiconductors (ITRS), 2003.

[2] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[3] W. Batty et al. Global coupled EM-electrical-thermal simulation and experimental validation for a spatial power combining MMIC array. *IEEE Transactions on Microwave Theory and Techniques*, pages 2820–33, Dec. 2002.

[4] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proc. HPCA-7*, pages 171–182, January 2001.

[5] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. ISCA-27*, pages 83–94, June 2000.

[6] J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration (GSI)—part I: Derivation and validation. *Electron Devices, IEEE Transactions on*, 45(3):580–589, March 1998.

[7] W. E. Donath. Wire-length distribution for placements of computer logic. *IBM Journal of Research and Development*, 2(3):152–155, May 1981.

[8] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S.Velusamy. Compact thermal modeling for temperature-aware design. In *Proc. 41st DAC*, pages –, June 2004.

[9] T. Kam, S. Rawat, D. Kirkpatrick, R. Roy, G. S. Spirakis, N. Sherwani, and C. Peterson. EDA challenges facing future microprocessor design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 19(12):1498–1506, December 2000.

[10] C. J. M. Lasance. Two benchmarks to facilitate the study of compact thermal modeling phenomena. *Components and Packaging Technologies, IEEE Transactions on*, 24(4):559–565, December 2001.

[11] S. Lee, S. Song, V. Au, and K. Moran. Constricting/spreading resistance model for electronics packaging. In *Proc. AJTEC*, pages 199–206, March 1995.

[12] R. H. J. M. Otten and R. K. Brayton. Planning for performance. In *Proc. DAC 98*, pages 122–127, June 1998.

[13] J. Parry, H. Rosten, and G. B. Kromann. The development of component-level thermal compact models of a C4/CBGA interconnect technology: The motorola PowerPC 603 and PowerPC 604 RISC microproceesors. *Components, Packaging, and Manufacturing Technology–Part A, IEEE Transactions on*, 21(1):104–112, March 1998.

[14] G. S. Rose. Modeling the thermal properties of ic packaging using thermal rc networks. Technical report, University of Virginia, Dept. of Electrical and Computer Engineering, 2002. Available at http://www.ece.virginia.edu/hplp.

[15] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. *Proceedings of the IEEE*, 91(2):305–327, February 2003.

[16] S. Rzepka, K. Banerjee, E. Meusel, and C. Hu. Characterization of self-heating in advanced VLSI interconnect lines based on thermal finite element simulation. *Components, Packaging, and Manufacturing Technology–Part A, IEEE Transactions on*, 21(3):406–411, September 1998.

[17] K. Skadron, K. Sankaranarayanan, S. Velusamy, D. Tarjan, M. R. Stan, and W. Huang. Temperature-aware microarchitecture: Modeling and implementation. *ACM Transactions on Architecture and Code Optimization*, 2004. To appear.

[18] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proc. ISCA-30*, pages 2–13, June 2003.

[19] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture: Extended discussion and results. Technical Report CS-2003-08, University of Virginia, Computer Science Department, 2003.

[20] V. Székely, C. Márta, M. Renze, G. Végh, Z. Benedek, and S. Török. A thermal benchmark chip: Design and applications. *Components, Packaging, and Manufacturing Technology–Part A, IEEE Transactions on*, 21(3):399–405, September 1998.

[21] K. Torki and F. Ciontu. IC thermal map from digital and thermal simulations. In *Proceedings of the 2002 International Workshop on THERMal Investigations of ICs and Systems (THERMINIC)*, pages 303–08, Oct. 2002.

[22] Y. F. Tsai. An architecture-level leakage power simulator. In *Ph.D. Forum at DATE 2004*, February 2004.
[23] N. Vasseghi, K. Yeager, and et al. 200-mhz superscalar risc microprocessor. *Solid State Circuits, IEEE Journal of*, 31(11):1675–1686, Novemebr 1996.
[24] T.-Y. Wang and C. C.-P. Chen. 3-D thermal-ADI: A linear-time chip level transient thermal simulator. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(12):1434–1445, December 2002.
[25] P. Zarkesh-Ha, J. A. Davis, W. Loh, and J. D. Meindl. On a pin versus gate relationship for heterogeneous systems: Heterogeneous rent's rule. In *Proc. IEEE CICC*, pages 93–96, May 1998.
[26] P. Zarkesh-Ha and J. D. Meindl. Stochastic net-length distribution for global interconnects in a heterogeneous system-on-a-chip. In *Proc. IEEE Symp. on VLSI Tech.*, pages 44–45, June 1998.