

## HotSpot: Techniques for Modeling Thermal Effects at the Processor-Architecture Level

Kevin Skadron<sup>†</sup>, Mircea Stan<sup>‡</sup>, Marco Barcella<sup>‡</sup>, Amar Dwarka<sup>‡</sup>, Wei Huang<sup>‡</sup>,  
Yingmin Li<sup>†</sup>, Yong Ma<sup>‡</sup>, Amit Naidu<sup>†</sup>, Dharmesh Parikh<sup>†</sup>, Paolo Re<sup>‡</sup>,  
Garrett Rose<sup>‡</sup>, Karthik Sankaranarayanan<sup>†</sup>, Ram Suryanarayan<sup>‡</sup>,  
Sivakumar Velusamy<sup>†</sup>, Hao Zhang<sup>‡</sup>, Yan Zhang<sup>‡</sup>

<sup>†</sup>Dept. of Computer Science, <sup>‡</sup>Dept. of Electrical and Computer Engineering  
University of Virginia  
Charlottesville, VA 22904  
*skadron@cs.virginia.edu* or *mircea@virginia.edu*

### Abstract

This paper describes a thermal-modeling approach that is easy to use and computationally efficient for modeling thermal effects and thermal-management techniques at the processor architecture level. Our approach models thermal behavior of the die and its package as a circuit of thermal resistances and capacitances that correspond to functional blocks at the architecture level. This yields a simple model that still accounts for heating in individual architecture-level blocks, while also capturing heat flow among blocks and through the package. A model like this that can be integrated with cycle-level microarchitecture simulators is needed, because the architecture community has demonstrated growing interest in thermal management, but currently lacks any way to model on-chip temperatures in a tractable way.

### 1. Introduction

Many analysts suggest that increasing power density and resulting difficulties in managing on-chip temperatures are some of the most urgent obstacles to continued scaling of VLSI systems within the next five to ten years. Just as has been done before for power-aware computing, “temperature-aware” computing must be approached not just from the packaging community, but also the VLSI and processor-architecture communities. In particular the solutions developed by the VLSI and architecture communities are often synergistic and typically require cooperation. Circuit techniques can reduce heat dissipation for all circuits of a particular style, while processor-architecture solutions can often use global, runtime knowledge to change the behavior of large portions of the processor. There is growing interest in architecture-level solutions, as evidenced by recent work on fetch throttling and dynamic voltage scaling in response to thermal stress [2, 7, 12]. *Yet the architecture community is currently completely lacking a way to model temperature at any level of granularity other than low-level circuits!* As earlier work [12] using even a very simple thermal model showed, the accuracy of thermal modeling has a substantial effect on the accuracy of thermal-management studies at the processor-architecture level and the conclusions they draw. Without this essential modeling capability, architecture researchers are limited to crude and inaccurate estimation techniques and are unable to effectively develop and evaluate techniques for thermal management.

We describe work in progress to develop a thermal modeling framework called *HotSpot* for use by processor architects. It con-

sists of portable modules to be incorporated into popular cycle-accurate processor power/performance simulators like *Wattch* [3]. The key capabilities of *HotSpot* are

- *Floorplanning*: The ability to use the specified architecture-level processor configuration and user-supplied area estimates to derive a high-level floorplan that gives adjacencies of the functional blocks of interest.
- *Lumped-RC Modeling*: The ability to model dynamic, localized heating in and among architecture-level functional blocks. This is accomplished by deriving an equivalent circuit model based on lumped thermal resistances and capacitances for the blocks and their connections to each other and to the package. The circuit is simple, making it reasonable to use in a cycle-level microarchitecture simulator.

These have been integrated into a portable software package that can easily be incorporated into architecture-level simulators and has already been integrated into *Wattch* [3].

### 2. Thermal Modeling at the Architecture Level

Prior work on thermal issues in the architecture field has not modeled temperature at all, using instead a mere average of power dissipation values as a proxy. Yet a simple average is insufficient to capture transient thermal behavior. *HotSpot* provides a direct model of temperature by computing thermal R/C values and constructing a circuit model for the heat dissipation by and among the different architecture-level blocks within a microprocessor. This functionality is encapsulated in a portable software module that can be integrated into architecture-level power-performance simulators like *Wattch*. The power-performance simulator determines cycle-by-cycle power dissipations, which *HotSpot* then uses to compute heating and heat flow over time.

In order to derive a lumped circuit model we need to decide on the level of granularity for the lumped elements. For pipeline-level modeling, we use a natural partitioning where functional blocks on the chip are the nodes in the lumped circuit model. Additional R/C elements in the circuit account for packaging components. This granularity has the advantage that there is a one-to-one correspondence between the model in a power/performance simulator like *Wattch* and the model in the thermal simulator, which leads to a straightforward coupling between the two. Naturally, it has the disadvantage that it assumes uniform power density within a block, when in fact hot spots may occur at finer granularities. Since

HotSpot is intended for use in early pipeline-level microarchitecture planning and research, when no lower-level descriptions are available, we argue that this is a necessary abstraction. Safe operating temperatures at the granularity of blocks must be chosen in a way that more localized hot spots will also operate at safe temperatures.

The circuit model we derive is solved by writing a set of ordinary differential equations using Kirchoff's current law and solving it using fourth-order Runge-Kutta. Each call to the circuit solver takes about 100 microsec., so we average power dissipations over 10 clock cycles and compute temperatures at this slightly reduced granularity (*i.e.*, 5–10 nanosec., much shorter than the thermal time constants). The thermal modeling slows down simulation speeds in Wattch by less than 50–60%.

This paper focuses on techniques for deriving a high-level floorplan and for deriving lumped R and C values for the architectural blocks of interest. The circuit model is dependent on the floorplan and packaging of the processor being simulated, while the floorplan and the calculation of values for thermal resistance and capacitance are dependent on the areas of the functional blocks of interest. This is sufficient to model actual on-chip operating temperatures, but a complete modeling and evaluation tool for thermal management must account for the behavior of temperature *sensors*. Before describing our floorplan and lumped-RC derivations, we give a brief overview of how HotSpot will treat these other issues.

### 2.1. Package

Because we wish to model thermal behavior using an equivalent circuit, the package must also be represented with thermal R/C values. Fortunately, there is a rich body of work describing how to derive these values—we primarily follow [9]—and R/C values for many packages can also be found online. Eventually, HotSpot will determine the thermal resistance of any specified cooling package consisting of a heat sink, heat spreader, and some bonding material; and will account for conduction through a bulk material, thermal spreading from a small area to a larger one (including bulk material resistance [9]), and convection from air that flows around the fins.

### 2.2. Area

Efficient architecture-level thermal modeling requires estimation of the area of various functional units based on architectural parameters. The area estimation itself can also allow the designers to make more comprehensive design space comparisons in the early phase of the design process. Unfortunately, area estimation is fraught with inaccuracy due to custom sizing and other design decisions. HotSpot therefore currently uses externally defined areas of architectural functional blocks. These can be obtained from the literature or published floorplans. Eventually, HotSpot will use analytic models to scale these externally-supplied measurements for exploring alternative architectures.

### 2.3. Sensors

An RC network that matches the desired microarchitecture is sufficient to simulate the actual evolution of temperatures in different architectural blocks. This can be used to determine how thermal stress is correlated to the architecture, and how design decisions influence thermal behavior and related effects like leakage currents. After work on HotSpot is completed, the next step is to develop a model for sensor accuracy using sensors like those proposed by Székely *et al.* [14] and Syal *et al.* [13]. This will then permit the realistic exploration of thermal-management techniques that respond to high on-chip temperatures by reducing power dissipation while minimizing performance loss.

### 2.4. Related Work

A great deal of work has explored thermal electro-thermal simulation at the transistor and logic levels, for example the SISSI package by Székely *et al.* [15]. Other work has looked at thermal modeling for entire packages and/or boards, for example the SUNRED and THERMAN tools developed by Rencz *et al.* [11], and MONSTR developed by Koval and Farmaga [8].

But we are unaware of any prior work to perform thermal modeling at the level of the processor architecture and integrate it with a cycle-accurate power-performance simulator like Wattch. The closest work we have been able to find is the iTAS environment by Cheng and Kang [4], which models steady-state heating at the functional-block level, but does not perform dynamic modeling nor consider how to accommodate blocks at the architecture level.

## 3. Floorplanning: Modeling Thermal Adjacency

Current microprocessor simulators at the architecture level do not model floorplan information, yet this knowledge is becoming essential for architecture-level modeling of performance, power, and heat. This necessitates a simple floorplanning tool that does not require lower-level information, because architecture research and product planning often wishes to study architectures that may never be realized in lower-level descriptions that traditional floorplanning algorithms require. The HotFloorplan tool incorporates a new algorithm to solve this problem.

We have observed that published floorplans of processor chips reflect pipeline order: units involved in adjacent pipeline stages are typically also adjacent in the floorplan. This suggests that the ordering of the pipeline stages is a good way to express floorplans at the architectural level. The user pictures the floorplan as a set of “concentric semicircular strips”. Each strip corresponds to a sequence of functional blocks and their areas. The strips are fit onto the chip by adjusting their widths, assuming the blocks' aspect ratio is variable. Apart from the ease of specification, another advantage of these semicircular strips is the ease of modification. Future work will allow the user to completely specify a floorplan using vertical and horizontal adjacency matrices. This is desirable when simulating a system with a known floorplan.

A sample floorplan generated by our tool appears as the right-most item in Figure 1. In order to compare this default floorplan with the floorplans of contemporary microprocessors, we took the die photographs and floorplans of a variety of processors, two of which are also shown in Figure 1: the Alpha 21264 [10] and the MIPS R10000 [5]. For comparison, we merged the manufacturer-specified blocks in those photographs so that they approximately match the blocks of the default floorplan. We then flipped and rotated those layouts to get similar orientations. It can be seen from these figures and other floorplans which we examined that the blocks adjacent to each other in the floorplan are indeed typically the blocks that are adjacent in the ordering of the pipeline stages. Moreover, the default floorplan, which is generated using the ordering of pipeline stages, qualitatively resembles the real floorplans.

Since a great deal of customization goes into most floorplans, our goal is to develop an algorithm that can produce floorplans with reasonable adjacencies for high-level modeling of thermal, power, and delay effects. Results so far suggest that our high-level floorplanning tool does a good job of producing reasonable layouts and can be used to provide sufficiently realistic adjacency information for high-level performance, power, and thermal studies.

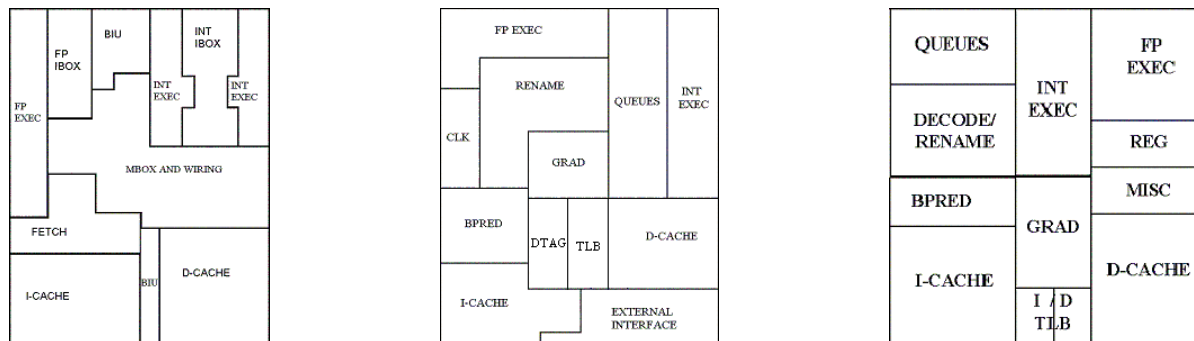


Figure 1. “Normalized” floorplan for the Alpha 21264 (left) and MIPS R-10000 (middle) processors; and the “generic” floorplan generated by the HotFloorplan algorithm (right).

#### 4. Deriving Lumped Thermal R/C Values

Given a floorplan with areas and the thermal resistance and capacitance values for the package, the next step is to derive a circuit to model dynamic heat flow in the chip, and compute the values for the remaining thermal Rs and Cs. For every time step, the thermal model receives the power dissipated in each block, and determines the average temperatures of each of those blocks at the end of that time step. The circuit model of a simple chip with several blocks is shown in Figure 2.

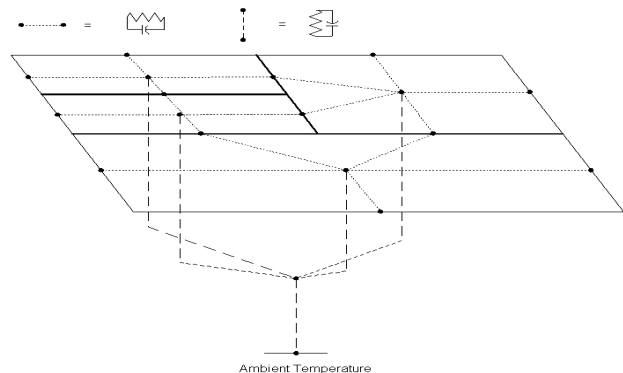


Figure 2. Sample circuit model for a simple floorplan of four blocks.

(Dotted lines in the plane of the die represent circuit elements that capture lateral thermal behavior; dashed lines represent vertical elements that capture thermal behavior in the package.)

The HotBlocks tool computes a thermal resistance and a capacitance from each of the blocks to all its neighbors. Further RC pairs from the center of each block to a common node represent the flow of heat from each block into the package. This captures spatial non-uniformity in temperature in the package. Finally, a fixed thermal R and C models the heat-sink. These Rs and Cs depend on the physical parameters and on the geometry of the blocks and of the die. For validating the HotBlocks technique, we use simple floorplans and power-density values and solved the RC circuits using Spice, but the HotBlocks tool and associated Runge-Kutta circuit solver have been completed and successfully integrated into Wattch.

Because the vertical thermal Rs and Cs dominate heat transfer in silicon, it is straightforward to think about a simpler model without lateral thermal Rs and Cs. Our comparison tests a simpler model, which does not contain lateral thermal Rs and Cs, with one that in-

cludes them. The results show that inclusion of lateral components provide much better results.

We evaluated the accuracy of our approach using different heat sinks, chip thicknesses, floorplans, and distributions of power densities. For reasons of space, we present evaluation using a simple abstract floorplan, shown in Figure 3, and two different sets of power densities, also shown in Figure 3. We used a die thickness of  $0.5\text{mm}^2$ , and a thermal resistance of  $0.7\text{ K/W}$  for the heat spreader and heat sink combined [16]. Results for other configurations, including for a more realistic floorplan with ten blocks that approximates the block sizes and power densities of an actual processor, can be found in [1].

There is currently no good way to take actual, localized temperature measurements from a real chip, so to obtain “reference” data and validate our model, we compared the computed results with those obtained by Floworks [6]. Floworks is a commercial, detailed, validated, finite-element simulator of 3D heat flow. It calculates both flow patterns and heat transfers for systems of various geometries, combinations of materials, fluid flows, boundary conditions, and heat sources. In our simulations in Floworks, we use the function of heat transfer in solids to get our reference temperatures of different blocks. Our model includes forced airflow in a confined box with a copper heat sink of 28 fins.

|   |   |   | Set 1  |         | Set 2 |         |       |
|---|---|---|--------|---------|-------|---------|-------|
|   |   |   | Pwr    | Density | Power | Density |       |
| 4 | 5 | 1 | Block1 | 30      | 1.071 | 10      | 0.357 |
|   |   |   | Block2 | 20      | 1.667 | 20      | 1.667 |
|   |   |   | Block3 | 10      | 0.417 | 10      | 0.417 |
|   |   |   | Block4 | 15      | 0.625 | 40      | 1.667 |
|   |   |   | Block5 | 25      | 1.25  | 20      | 1.667 |
| 3 |   | 2 |        |         |       |         |       |

Figure 3. Left: One of the floorplans used for testing; total area is  $100\text{ mm}^2$ . Right: The two different distributions of power dissipation we assumed.

(Power is in watts and density is in  $\text{W}/\text{mm}^2$ )

The results are shown in Figure 4, with temperatures in Kelvin and an ambient of  $303.2\text{ K}$ . The percentage error (relative to the ambient) with the circuit model proposed in Figure 2 is less than 2% for our proposed model. The simple model is much worse, indicating the importance of accounting for lateral thermal coupling.

Validation of HotBlocks is currently complete only for steady state due to the long computation times in Floworks. This means that the accuracy of the lumped thermal capacitances and their effect on transient behavior is not yet fully validated. Nevertheless, the evaluation so far shows the importance of lateral thermal flow

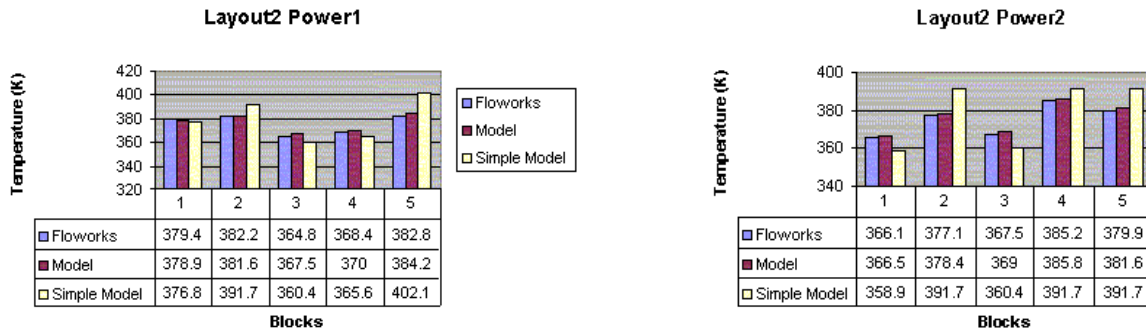


Figure 4. Comparison of lumped-RC and finite-element (Floworks) simulations of per-block temperatures. “Floworks” is the reference from Floworks, “Model” is our proposed model that includes lateral R and C, and “Simple Model” omits the lateral R and C.

and that the circuit network we propose provides a high degree of accuracy for steady state, certainly well within the precision requirements of architecture-level modeling. The tight agreement suggests that the circuit network we propose captures the important thermal effects. We are now in the process of conducting the much more computationally expensive validation of transient effects, again using Floworks results as a reference. Preliminary results in [1] suggest that our lumped circuit model closely tracks the reference results for transient behavior too.

## 5. Conclusions and Future Work

This paper described an approach to modeling thermal behavior in architecture-level power/performance simulators. Our technique is based on a simple network of thermal resistances and capacitances, which are derived using area and floorplan information for the major functional blocks at the architecture level. The floorplans that we generate match published floorplans well, and our algorithm requires no behavioral, RTL, or synthesis information. The circuit model that we propose has so far matched reference results found with Floworks within 2%.

The thermal model which we propose here, called HotSpot, fills a void in the architecture community’s modeling capabilities. It can be used to simulate how thermal stress is correlated to the architecture, and how design decisions influence thermal behavior and other important effects that are dependent on operating temperature like leakage currents. Future work consists of developing models for sensor accuracy and delay so that run-time techniques for thermal management can be explored as well. It is our hope that this paper will stimulate greater collaboration among the package, VLSI, and architecture communities to find better ways to model and manage heat at all levels of computer systems design.

## Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This work is supported in part by the National Science Foundation under grant nos. CCR-0133634 and MIP-9703440, a grant from Intel MRL, and an Excellence Award from the Univ. of Virginia Fund for Excellence in Science and Technology.

## References

- [1] M. Barcella, W. Huang, K. Skadron, and M. Stan. Architecture-level compact thermal R-C modeling. Tech. Report CS-2002-20, Univ. of Virginia Dept. of Computer Science, July 2002.
- [2] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proc. of the Seventh Int’l Symp. on High-Performance Computer Architecture*, pp. 171–82, Jan. 2001.
- [3] D. Brooks, V. Tiwari, and M. Martonosi. Watch: A framework for architectural-level power analysis and optimizations. In *Proc. of the 27th Ann. Int’l Symp. on Computer Architecture*, pp. 83–94, June 2000.
- [4] Y.-K. Cheng and S.-M. Kang. A temperature-aware simulation environment for reliable ULSI chip design. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(10):1211–20, Oct. 2000.
- [5] MIPS R10000 die photo. From website: CPU Info Center. [http://bwrc.eecs.berkeley.edu/CIC/die\\_photos/#mips](http://bwrc.eecs.berkeley.edu/CIC/die_photos/#mips).
- [6] Floworks: Fluid Flow Analysis for SolidWorks. Website. <http://www.floworks.com>.
- [7] W. Huang, J. Renau, S.-M. Yoo, and J. Torellas. A framework for dynamic energy efficiency and temperature management. In *Proc. of the 33rd Ann. IEEE/ACM Int’l Symp. on Microarchitecture*, pp. 202–13, Dec. 2000.
- [8] V. Koval and I. W. Farmaga. MONSTR: A complete thermal simulator of electronic systems. In *Proc. of the 31st Design Automation Conf.*, June 1994.
- [9] S. Lee, S. Song, V. Au, and K.P. Moran. Constricting/spreading resistance model for electronics packaging. In *Proc. of the ASME/JSME Thermal Eng. Conf.*, pp. 199–206, Mar. 1995.
- [10] M. Matson et al. Circuit implementation of a 600MHZ superscalar RISC microprocessor. *Computer Design: VLSI in Computers and Processors*, 26(2):104–110, Feb. 1998.
- [11] M. Rencz, V. Székely, A. Poppe, and B. Courtois. Friendly tools for the thermal simulation of power packages. In *Proc. of the Int’l Wkshp On Integrated Power Packaging*, pp. 51–54, July 2000.
- [12] K. Skadron, T. Abdelzaher, and M. R. Stan. Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management. In *Proc. of the Eighth Int’l Symp. on High-Performance Computer Architecture*, pp. 17–28, Feb. 2002.
- [13] A. Syal, V. Lee, A. Ivanov, and J. Altet. CMOS differential and absolute thermal sensors. In *Proc. of the Seventh Int’l On-Line Testing Wkshp.*, pp. 127–132, 2001.
- [14] V. Székely, C. Márta, Z. Kohári, and M. Rencz. CMOS sensors for on-line thermal monitoring of VLSI circuits. *IEEE Trans. on VLSI Systems*, 5(3):270–276, Sept. 1997.
- [15] V. Székely, A. Poppe, A. Páhi, A. Csendes, and G. Hajas. Electro-thermal and logi-thermal simulation of VLSI designs. *IEEE Trans. on VLSI Systems*, 5(3):258–69, Sept. 1997.
- [16] R. Viswanath, W. Vijay, A. Watwe, and V. Lebonheur. Thermal performance challenges from silicon to systems. *Intel Technology J.*, Q3, 2000.