Thermal Modeling for Processors and Systems-on-Chip

Article · August 2010

DOI: 10.1007/978-1-4419-6175-4_15

CITATION
1

READS
423

3 authors, including:

Kevin Skadron
University of Virginia
315 PUBLICATIONS **15,538** CITATIONS

SEE PROFILE

Mircea R Stan
University of Virginia
299 PUBLICATIONS **10,686** CITATIONS

SEE PROFILE

Chapter 15
Thermal Modeling for Processors and Systems-on-Chip

Kevin Skadron, Mircea Stan, and Wei Huang

Abstract Chip power density and consequently on-chip hot spot temperature have been increasing steadily as a result of non-ideal technology scaling, leading to severely thermally constrained designs. In this chapter, we review a chip- and package-level thermal modeling and simulation approach, HotSpot, that is unique because it is compact, correct by construction, flexible, and parameterized. HotSpot is important for temperature-aware design, especially during early pre-RTL stages of SoC and processor designs. Several case studies further illustrate the necessity of thermal simulations and the usefulness of HotSpot.

15.1 Temperature-Aware Design

An unfortunate side effect of miniaturization and the continued scaling of CMOS technology is a steady increase in power densities. The resulting difficulties in managing temperature, especially local hot spots, have become one of the major challenges for designers. High temperatures have several detrimental effects on VLSI systems such as microprocessors and systems-on-chip (SoC). First, the device carrier mobility is degraded at higher temperatures, resulting in slower devices. Second, leakage power is escalated due to the exponential increase of sub-threshold current with temperature. Third, the interconnect resistivity increases with temperature, leading to more severe power grid IR drops and longer interconnect RC delays, causing performance loss and complicating timing and noise analysis. Finally, elevated temperatures accelerate interconnect and device aging, while package reliability can be severely affected by local hot spots and higher temperature gradients. For all these reasons, in order to fully account for the thermal effects, it is important to model chip temperature in an accurate but also efficient way at all stages of the design. In particular, it is crucial to take thermal effects into account as early as possible in the design cycle, since early high-level thermal-aware design decisions can significantly improve design efficiency and reduce design cost.

K. Skadron (✉)
Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA
e-mail: kadron@cs.virginia.edu

R. Leupers, O. Temam (eds.), *Processor and System-on-Chip Simulation*,
DOI 10.1007/978-1-4419-6175-4_15,

Figure 15.1 shows a basic ASIC/SoC design flow adapted to become *temperature-aware*. Temperature profiles are needed at both functional-block level and standard-cell level during the design flow. Similar arguments also apply to microprocessor design flows. From the flow it is clear 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 temperatures can then be used to perform power, performance, and reliability analyses, together with placement, packaging design, etc. As a result, all the design 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 essential because the operating temperatures are properly considered during the *entire* design flow instead of being determined only after the fact and at the end of the design flow.

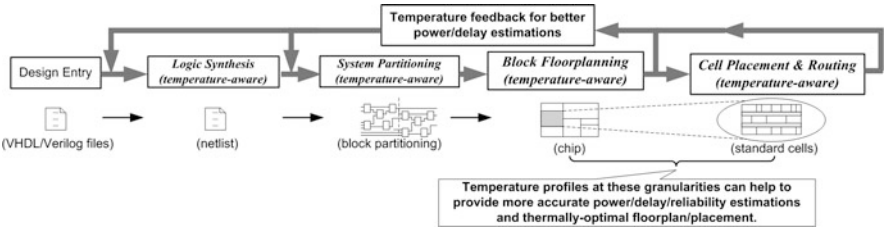


Fig. 15.1 An example of temperature-aware ASIC/SoC design flow [1]

It is important to recognize that power-aware design is insufficient for temperature management, because temperature also depends on the pattern of power dissipation in space and time, packaging and cooling choices, and the behavior of other system components. Furthermore, sometimes lower energy or low-power solutions can actually lead to higher temperatures (e.g., solutions that have higher local power densities), while a temperature-aware design can allow cheaper packages, quieter fans, and higher reliability. Also, low-power techniques tend to conserve power when activity is low, while thermal effects are a problem when activity is high. Thermal management techniques must therefore be designed differently from energy management techniques, even if they use some of the same underlying design "knobs" to control power. In recent years, research on temperature-aware design in the computer architecture and circuits communities has rapidly expanded, and more often we see chip architects and circuit designers branching out to collaborate with the design automation and thermal engineering communities.

15.2 Compact Thermal Modeling

The thermal model used to estimate temperatures is the key element for a temperature-aware design methodology. Figure 15.2 shows how the thermal model helps to close the loop for accurate power, performance, and reliability estimations.

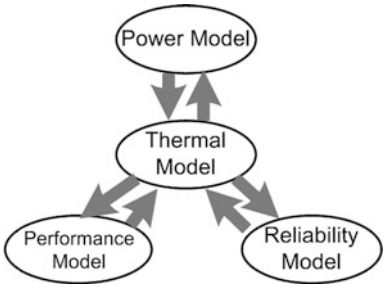


Fig. 15.2 Interactions among thermal model and power, performance, and reliability models [1]

For example, the power model first provides estimated power to the thermal model. The thermal model in turn provides estimated temperatures 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.

Due to the huge computational requirements, it is almost impossible to model temperature and analyze the thermal effects of a system together with the environment in their full details. Using detailed numerical analysis methods, such as finite-element models (FEM), is time-consuming and costly, hence appropriate to model temperatures only in special cases. The best trade-offs are offered by compact thermal models (CTMs) with reasonably accurate temperature predictions at different levels, e.g., circuit level, die level, package level [2].

A top-down hierarchy of compact thermal models can help designers at different levels of abstraction. As listed below, there are several desirable features that increase the usefulness of such a compact thermal model at a particular level.

15.2.1 Correct by Construction

A fully by-construction and parameterized compact thermal model allows chip designers and computer architects to explore new design alternatives and evaluate different thermally related design trade-offs at their corresponding design levels before the actual physical design becomes available. More importantly, with the aid of the parameterized compact thermal models, designers at different design levels can have more productive interactions and collaborations during early design stages. This leads to early discovery and considerations of potential thermal hazards of the system. True *physical* parameterization requires that the models be constructed solely on design geometries and material properties.

15.2.2 Boundary-Condition Independence (BCI)

A crucial feature of compact thermal models is boundary-condition independence (BCI). By achieving BCI, assumptions about the environment (e.g., ambient

temperature) do not affect the construction of the actual model. Prior *package-level* compact thermal models in [3, 4] achieve BCI by finding a thermal resistance network with minimum overall error when used with different boundary conditions [4–6]. At the chip and block level, we also need to find a way to construct a thermal model that is BCI.

15.2.3 Analogy of Electrical and Thermal Phenomena

The compact model must be easy to construct. One possibility is to utilize a well-known analogy of heat transfer and electrical phenomena, as shown in Table 15.1. In this analogy, 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 analogy 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). A compact thermal model is essentially a thermal RC circuit. Each node in the circuit corresponds to a block at the desired level of granularity. The 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.

Table 15.1 Analogy of thermal and electrical quantities

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	Ω
C_{th}, Thermal capacitance	J/K	C, Electrical capacitance	F

15.2.4 Detailed Temperature Information

A compact thermal model should also provide thermal information at the desired level of abstraction. For example, for package-level compact thermal models, previous studies [7, 8] have shown that the information of temperature distribution across the package is required. Using only a single junction-to-case thermal resistance leads to an inferior package design; instead, multiple nodes are needed on the package surfaces. Similarly, a compact thermal model at the silicon level should consist of enough nodes for detailed temperature distribution information across the die. In addition, both static and transient temperatures need to be modeled.

15.2.5 Granularity

A compact thermal model needs to match the spatial and temporal granularities to the level of abstraction (lower granularities at higher levels, and higher granularities at lower levels) in order to hide the details of the lower levels and make sure that the compact thermal model itself is no more complex than necessary. Modeling at finer granularity introduces unnecessary details and makes the computation slower. For example, system-level package compact thermal models, such as the DELPHI models [3, 4], hide the lower level details of the package structures, including the die, the thermal attach, and the solder balls, mainly because these details are intellectual properties of the vendors, but also because they would increase the complexity of the model without significantly improving the simulation accuracy. Similarly, a compact thermal model at the die level should hide the lower level details of the die, such as the actual circuit structures and physical layout.

15.2.6 Pitfalls

In order to construct a reasonably accurate yet compact thermal model for processor and SoC designs, there are several points worth considering:

- *Aspect ratio*. We found that a block with high aspect ratio cannot be modeled as one node. This is because the heat spreading in one direction dominates the other directions, resulting in a significantly non-uniform temperature distribution within the block, and hence a single temperature is no longer representative to the entire block. The solution is to divide the block into several sub-blocks with aspect ratios close to unity [9]. This also applies in the depth dimension, i.e., if a block occupies a tiny chip area (e.g., $100 \times 100\,\mu\text{m}$), whereas the chip is $700\,\mu\text{m}$ thick, it is better to further divide the chip into multiple sub-layers or to combine the tiny block with adjacent blocks to make the aspect ratio in the depth dimension closer to unity.
- *Package components*. It is tempting to simplify the thermal package (including thermal interface material (TIM), heat spreader, heat sink, and the secondary heat transfer path) into only one or a few nodes and try to have a simple thermal boundary condition for the silicon die. The result, however, may be inaccurate and yield misleading silicon temperature estimations. The right thing to do is to analyze the geometries and thermal properties of each package components and understand their impact on the die temperature before simplifying them [10].
- *Granularity*. In most cases, it is not necessary to model temperature at granularities that are so fine as to be computationally infeasible, such as nanometers and nanoseconds. This is because, in time, the thermal time constants (milliseconds or more) are much longer than electrical time constants; and the lateral heat spreading makes a small heat source (size less than a few hundred microns on each side) relatively cool [11]. We show some analysis and examples related to this issue in later sections of the chapter.

15.3 The *HotSpot* Thermal Modeling Approach

In this section, we present a compact thermal modeling method, named *HotSpot*, which takes a structured assembly approach of constructing a physical compact thermal model by first modeling the silicon die and other packaging components as a collection of simple 3D shapes, and then assembling them into more complex compact thermal models according to the overall structure. This modeling approach is fully parameterized and satisfies most of the above desirable features as shown later in the chapter.

15.3.1 Overview

When constructing a compact thermal model with HotSpot, one needs to first identify the different layers of the design that are made of different materials. This requires the designer have some prior detailed knowledge of the (intended) design structure. These layers are then stacked on top of each other as shown in Fig. 15.3. The layers can, for instance, represent heat sink, heat spreader, silicon substrate, on-chip interconnect layer, C4 pads, ceramic packaging substrate, solder balls, etc. Usually, the surface that generates heat is the surface of the silicon substrate layer.

Each layer is then divided into a number of blocks. For example, in Fig. 15.4c, the silicon substrate layer can be divided according to architecture-level blocks (only three blocks are shown for simplicity) or finer granularity, depending on what the die-level design requires. For other layers that may require less detailed thermal information, one can simply divide that layer as illustrated in Fig. 15.4a. The center shaded part in a layer shown by Fig. 15.4a is the area covered by another adjacent layer such as the one shown in Fig. 15.4c. This center part can have the same number of nodes as its smaller neighbor layer or collapse those nodes into a single node, depending on the accuracy and computation overhead requirements. The remaining part at the periphery in Fig. 15.4a is then divided into four trapezoidal blocks, and each is assigned one thermal node. Each block in each layer has a vertical (depth) thermal resistance and several lateral resistances, which model vertical heat transfer to its neighbor layers and lateral heat spreading/constriction within the layer itself, respectively. Figure 15.4b shows a side view of one block with both the lateral and the vertical resistances. Vertical (depth) resistances can

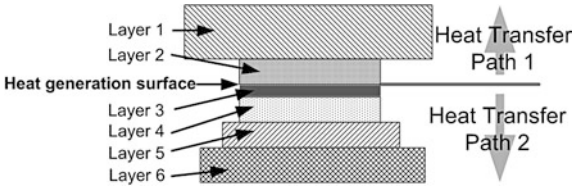


Fig. 15.3 The stacked layers of different materials in the HotSpot modeling approach. Heat generating surface and major heat transfer paths are also shown [12]

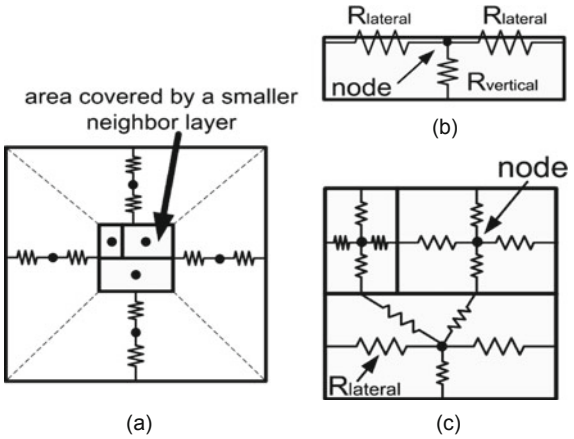


Fig. 15.4 (**a**) Area division of larger layers (*top view*). (**b**) Side view of one block with its lateral and vertical (depth) thermal resistances. (**c**) A layer, for example, the silicon die, can be divided into arbitrary number of blocks if detail thermal information is needed (*top view*) [13]

be calculated by $R_{\text{vertical}} = t/(k \cdot A)$, where t is the thickness of that layer, k is the thermal conductivity of the material of that layer, and A is the cross-sectional area of the block. Calculating lateral thermal resistances is not as straightforward as the depth resistances. This is because of the complex nature of modeling heat spreading and constriction. One can consider the lateral thermal resistance of one block as the spreading/constriction thermal resistance of the other parts within a layer to that specific block.

For layers that have surfaces interfacing with the ambient, i.e., the boundaries, we assume that each surface has a constant heat transfer coefficient h. The corresponding thermal resistance can then be calculated as $R_{\text{convection}} = 1/(h \cdot A)$, where A is the surface area. Strictly speaking, these convection thermal resistances are not part of the compact thermal model, because they include the information of the environment. If the environment changes, i.e., the boundary conditions change, the value of these convection resistances also change. On the other hand, for a particular design, the values of all the other thermal resistances shown in Fig. 15.4a–c should not change if the compact thermal model is BCI.

The HotSpot modeling approach first described in [14] has been successfully used in many academic as well as industry research activities.

15.3.2 Primary and Secondary Heat Transfer Paths

In HotSpot, we also consider lateral heat flow within a layer by adding lateral thermal resistances to achieve greater accuracy of temperature estimation. Figure 15.5 shows a modern single-chip CBGA package [15]. Heat generated from the active silicon device layer is conducted through the silicon die to the thermal interface material, heat spreader, and heat sink and then convectively removed to the ambient.

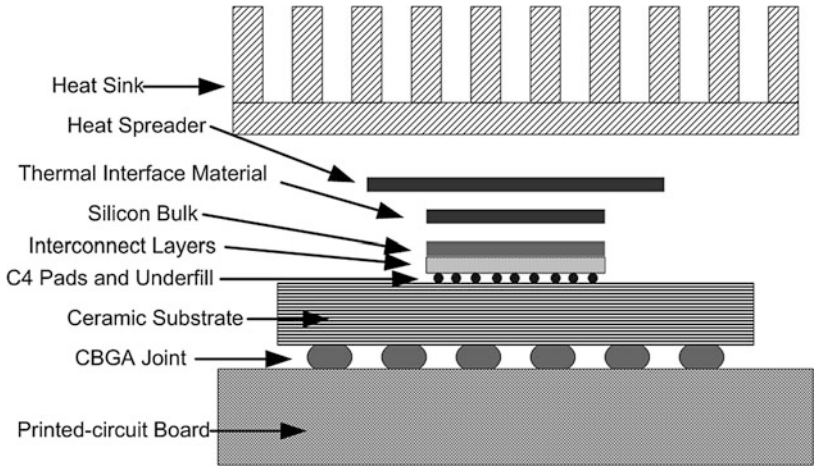


Fig. 15.5 Packaging components in a typical CBGA package, adapted from [15]

In addition to this primary heat transfer path, a secondary heat flow path exists due to conduction through the interconnect layer, I/O pads, ceramic substrate, leads/balls to the printed-circuit board. HotSpot models all these layers in both heat flow paths. Although the primary path removes most heat (more than 90%) in advanced cooling packages, the secondary path can also be essential in cooling solutions where forced convection or heat sinks are not used, as in most low-power SoCs.

15.3.3 Functional Units Versus Regular Grids

Layers such as the silicon die, the interface material, and the center part of the heat spreader can either be divided naturally according to functional unit shapes or be divided into regular grid cells, depending on the needs of the designer. For example, if a chip designer needs to estimate average temperatures for each functional block a thermal model at the functional unit granularity is best in that case. However, for a package designer, the temperature gradients across silicon die and other package layers are important metrics to evaluate the reliability of the package. In this case, a grid-like thermal model is more suitable, since it provides more detailed estimations of maximum and minimum temperatures, whereas the functional block-level thermal model does not. HotSpot offers a choice between a possibly irregular functional unit partitioning and a regular grid of variable granularity.

15.3.4 Model Validation

We performed a number of validations for the HotSpot thermal model—with detailed finite-element modeling software, thermal test chip [1], infrared thermal

imaging [16], and actual on-chip temperature sensors [17]. We have also validated that the HotSpot thermal modeling approach is boundary-condition independent [13].

15.4 Case Studies

In this section, we provide several brief case studies illustrating how to use a compact thermal modeling approach such as HotSpot in various aspects of thermal analysis and temperature-aware design for processors and SoCs.

15.4.1 A SoC Thermal Analysis Example

To illustrate the importance of temperature-aware design early in the design process, we show the thermal analysis together with the temperature-leakage closed loop for a SoC design [9]. We use InCyte®, a commercial early design estimation tool[1] to reconstruct an SoC design based on the published 180 nm design data in [18]. We use HotSpot 4.0 for the thermally self-consistent leakage analysis of this SoC design.

We picked logic and memory modules similar to those in [18] from InCyte's incorporated IP libraries and came up with an early SoC design whose total power was almost identical to data reported in [18]. InCyte also outputs a preliminary floorplan for the design. Notice that InCyte estimates leakage power of each block at a constant temperature. The estimated temperature map of this 180 nm design is shown in Fig. 15.6.

Because InCyte does not yet include the temperature dependence of leakage, whereas sub-threshold leakage is exponentially dependent on temperature, we double-check to see whether the thermally self-consistent leakage power causes thermal problems to this 180 nm SoC design. Using HotSpot and the simple leakage model in [10] to iterate the temperature-leakage loop as shown in Fig. 15.7, after convergence we find that the final total leakage is only a negligible 546 μW for this design with the selected package. Therefore, the above temperature estimation is quite accurate without considering the temperature-leakage loop.

However, if we re-design this SoC design in a 90 nm technology, there are two design possibilities: (1) We can scale both the area and active power of each individual blocks and thus maintain the same function and complexity. This means that the total power of the entire design is also scaled accordingly, and the power density remains roughly the same due to area scaling. Therefore, we can use a cheaper thermal package for less overall power consumption and keep the chip below the 85°C thermal constraint. (2) Since the ITRS [19] projects that the die size and power remain the same, if not increase, across different technology nodes, we can

[1] http://www.chipestimate.com/

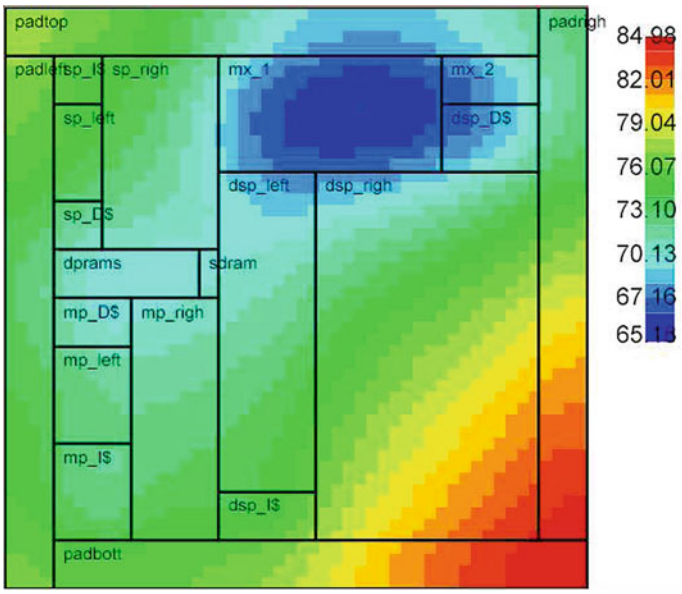


Fig. 15.6 Estimated temperature map of an SoC design at 180 nm technology, based on data in [18] and InCyte. Temperatures are in degree celsius [9]

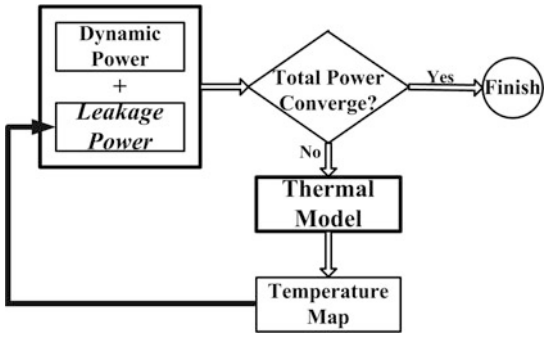


Fig. 15.7 A thermal model closes the loop for leakage power calculation [10]

alternatively assume that the total active power and chip area remain the same as those in the 180 nm design. Assuming a floorplan similar to that in 180 nm technology, this is equivalent to adding more parallelism (such as more processing cores and higher memory bandwidth) to the die and designing the chip for higher throughput by burning more power. In this case, with the same thermal package, after iterating with the leakage-temperature loop, the hottest on-chip temperature exceeds the thermal threshold and eventually causes thermal runaway! The reason is twofold: (1) at 90 nm, a greater fraction of total power consumption is caused by leakage [19], and (2) the sub-threshold leakage power's dependency on temperature is stronger at

90 nm than at 180 nm (using leakage model coefficients in [20, 21]). The results are listed in Table 15.2.

Table 15.2 As technology scales, temperature dependence of sub-threshold leakage power becomes more problematic. Without early-stage thermally optimized design flow, thermal runaway can happen even for low-power SoC designs

	Active power	Avg. temp rise	Hottest temp rise	Actual leakage	Leakage error at const temp
180 nm, orig. design	1.7 W	30.77°C	59.98°C	546 μW	116%
90 nm, scaled power area	316 mW	6.19°C	25.42°C	24 mW	66%
90 nm, same power area	1.7 W	>35.08°C	Runaway	>277 mW	>680%

The above SoC design example shows that it is crucial to incorporate thermal estimations (such as leakage-temperature dependence) early in the design process in order to locate potential thermal hazards that are too costly to fix in later design stages. At this early design stage, possible solutions to the SoC design at 90 nm can be as follows: (1) circuit designers can choose IPs that have high-Vt transistors and use reverse body-bias or sleep transistors for non-critical paths to reduce leakage; (2) architects can consider using dynamic voltage and frequency scaling (DVFS), migrating computation [20, 22], more parallelism, and temperature-aware floorplanning techniques [23], etc., to reduce hot spot temperatures. Alternatively, (3) package designers need to consider the possibility of adding a heatsink or a fan. Trade-offs among portability, cost, performance, and temperature have to be made.

15.4.2 Exploring Design Choices with Parameterization in HotSpot

As mentioned before, HotSpot is parameterized in terms of geometries and material properties of silicon, package components, and cooling solutions. Achieving full parameterization of compact thermal models is important in many ways. It allows designers at all design stages to freely explore all the possible thermally related design spaces. For example, the heat spreader and heat sink are two important package components for high-performance VLSI designs. While a large heat spreader and heat sink made from high thermal conductivity materials can reduce the temperature of silicon die, they also significantly increase the total price of the system. Therefore, exploring design trade-offs between hot spot temperatures and package cost is crucial. With parameterized compact thermal models, this exploration can be done easily and efficiently by simply sweeping the size of the heat spreader and heat sink with different material properties to achieve the desired package design point. On the other hand, building package prototypes or detailed thermal models greatly slows the design process and increases the design cost.

For illustration, we first present an example analysis to show the strength of using parameterized compact thermal models to efficiently investigate the impact of thermal interface material on across-silicon temperature differences. Figure 15.8 shows the relationship between the thickness of the thermal interface material (TIM), which glues the silicon die to the heat spreader. We plot the temperature readings from a compact thermal model with 40 × 40 grid cells on silicon. This analysis is based on an Alpha 21364-like floorplan shown in Fig. 15.9a. Average silicon die temperature is also plotted in Fig. 15.8 for reference. The total heat generated from the silicon surface is 40.2 W, the die size is 15.9 × 15.9 × 0.5mm, and the thermal conductivity of the thermal interface material is 1.33 W/(m-K).

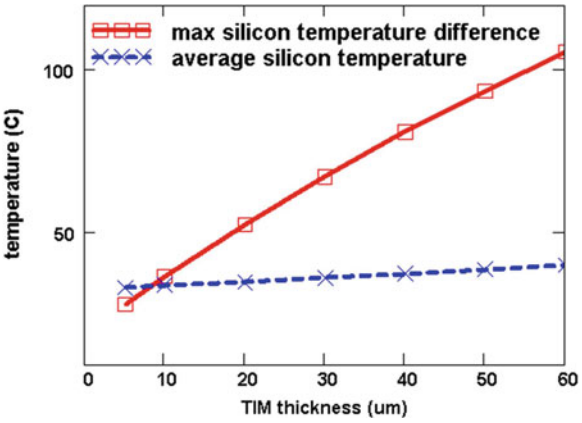
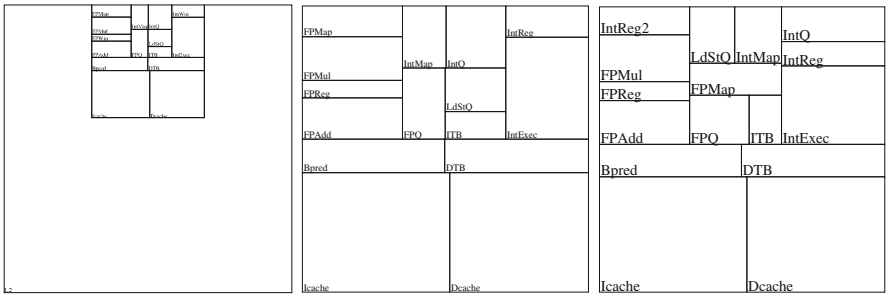


Fig. 15.8 The impact of thermal interface material (TIM) thickness to silicon die temperature difference. Average silicon die temperature is also plotted as reference [10]

As can be observed from Fig. 15.8, although the TIM thickness does not have an obvious impact on the average die temperature, thicker TIM results in poor heat spreading which leads to large temperature differences across the die. Such large temperature differences may be disastrous to circuit performance and die/package reliability. Using a better heat sink will only lower the average silicon temperature but will not help to reduce the temperature difference. From this analysis, which has been easily performed by our parameterized model, we can reach the conclusion that using as thin as possible a thermal interface material is one of the key opportunities for package designers to consider and more important than, for example, using a larger heatsink. In some recent work [4, 24], the importance of measuring and modeling thermal interface's impacts on the entire package has been also discussed, although not at the same silicon die level as the above example shows.

Another example of utilizing our parameterized compact thermal model is the investigation of a dynamic thermal management technique (DTM) at the micro-architecture level called migrating computation (MC) [22]. It is obvious that two silicon-level functional units that run hot by themselves will tend to run even hotter when adjacent. On the other hand, separating them will introduce additional

communication latency that is incurred regardless of operating temperature. This suggests the use of spare units located in cold areas of the chip, to which computation can migrate only when the primary units overheat. In this study, the floorplan of the Alpha 21364 core is carefully changed to include an extra copy of integer register file (see Fig. 15.9), which is usually the hottest spot on the silicon die for this design. We also model the fact that accessing the secondary integer register file entails extra power and extra access time due to the longer distance. With this new floorplan, we can shift the workload back and forth between the two register files when the one in use overheats, with a little performance overhead (11.2% slower). The changes in silicon floorplan can be easily adapted into corresponding parameterized compact thermal models, and thus the temperatures of functional units can be analyzed efficiently to investigate the usefulness of the migrating computation DTM technique. By doing this, packaging complexity and cost can be significantly reduced and still yield almost the same operating temperature and performance as with much more expensive and complicated thermal package.



(a) (b) (c)

Fig. 15.9 (**a**) Approximated floorplan of Alpha 21364 microprocessor. (**b**) Close-up look of the 21364 core, with only one register file on the *top-right corner*. (**c**) 21364 core with an extra copy of register file at the *top-left corner* [14]

15.4.3 Other Examples of HotSpot Usage by the Temperature-Aware Research Community

There are many other applications of the HotSpot thermal model. For example, temperature-aware floorplanning for both 2D [23] and 3D chips [25]; temperature considerations for reliability-aware design [21]; exploration of the thermal constraint on the design space for chip multiprocessor architectures [26, 27]; dynamic thermal management at the architecture level [14]; on-chip temperature sensor placement [28, 29]; temperature-aware high-level synthesis [30]; temperature-aware supply voltage planning in SoC [31]; thermal-aware SoC test scheduling [32], etc. We encourage interested readers to explore the plethora of research topics related to temperature-aware design.

15.5 Conclusion

In this chapter, we reviewed the important concept of temperature-aware design, especially during early stages of SoC and processor designs. We then introduced an accurate yet fast thermal modeling approach—HotSpot. HotSpot is unique and useful because it is compact, correct by construction, flexible, and parameterized. Since its introduction, HotSpot has been adopted and applied in many research areas in processor, SoC, and package design.

References

1. Huang, W., Stan, M.R., Skadron, K., Sankaranarayanan, K., Ghosh, S., Velusamy, S.: Compact thermal modeling for temperature-aware design. In: *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 878–883, San Diego, CA, USA, June (2004).
2. Sabry, M.-N.: Compact thermal models for electronic systems. *IEEE Trans Components Packaging Technol* **26**(1), 179–185, March (2003).
3. Rosten, H.I., Lasance, C.J.M., Parry, J.D.: The world of thermal charaterization according to DELPHI—part I: Background to DELPHI. *IEEE Trans Components Packaging Technol* **20**(4), 384–391, December (1997).
4. Lasance, C.J.M., Rosten, H.I., Parry, J.D.: The world of thermal charaterization according to DELPHI—part II: Experimental and numerical methods. *IEEE Trans Components Packaging Technol* **20**(4), 392–398, December (1997).
5. Lasance, C.J.M.: The influence of various common assumptions on the boundary-condition-independence of compact thermal models. *IEEE Trans Components, Packaging, Manufacturing Technol–Part A* **27**(3), 523–529, September (2004).
6. Vinke, H., Lasance, C.J.M.: Compact models for accurate thermal charaterization of electronic parts. *IEEE Trans Components, Packaging, Manufacturing Technol–Part A* **20**(4), 411–419, December (1997).
7. Rosten, H., Lasance, C.: Delphi: The development of libraries of physical models of electronic components for an integrated design environment. In: *Proceedings on Conference of International Electronics Packaging Society (CIEPS)*, (1994).
8. Bar-Cohen, A., Elperin, T., Eliasi, R.: θ_{jc} charaterization of chip packages—justification, limitations and future. *IEEE Trans Components, Hybrids, Manufacturing Technol* 12, 724–731, December (1989).
9. Huang, W., Skadron, K., Sankaranarayanan, K., Ribando, R.J., Stan, M.R.: Accurate, pre-RTL temperature-aware processor design using a parameterized, Geometric thermal model. *IEEE Trans Comput* **57**(9), 1277–1288, September (2008).
10. Huang, W., Humenay, E., Skadron, K., Stan, M.: The need for a full-chip and package thermal model for thermally optimized IC designs. In: *Proceedings of the ACM/IEEE International Symposium on Low Power Electronic Design (ISLPED)*, pp. 245–250, San Diego, CA, USA, August (2005).
11. Huang, W., Stan, M., Sankaranarayanan, K., Ribando, R., Skadron, K.: Many-core design from a thermal perspective. In: *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, Anaheim, CA (2008).
12. Huang, W., Stan, M.R., Skadron, K., Ghosh, S., Velusamy, S., Sankaranarayanan, K.: Hotspot: A compact thermal modeling methodology for early-stage vlsi design. *IEEE Trans Very Large Scale Integration (VLSI) Syst* **14**(5):501–513, May (2006).
13. Huang, W., Stan, M.R., Skadron, K.: Parameterized physical compact thermal modeling. *IEEE Trans Components Packaging Technol* **28**(4), 615–622, December (2005).

14. Skadron, K., Stan, M.R., Huang, W., Velusamy, S., Sankaranarayanan, K., Tarjan, D.: Temperature-aware microarchitecture. In: *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pp. 2–13, San Diego, CA, USA, June (2003).
15. Parry, J., Rosten, H., Kromann, G.B.: The development of component-level thermal compact models of a C4/CBGA interconnect technology: The motorola PowerPC 603 and PowerPC 604 RISC microproceesors. *IEEE Trans Components Packaging Manufacturing Technol–Part A* **21**(1), 104–112, March (1998).
16. Huang, W., Skadron, K., Gurumurthi, S., Ribando, R.J., Stan, M.R.: Differentiating the roles of IR measurement and simulation for power and temperature-aware design. In: *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Boston, MA, USA, April (2009).
17. Velusamy, S., Huang, W., Lach, J., Stan, M.R., Skadron, K.: Monitoring temperature in fpga based socs. In: *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, pp. 634–637, San Jose, CA, USA, October 2005.
18. Stolberg, H., Moch, S., Friebe, L., Dehnhardt, A., Kulaczewski, M., Berekovic, M., Pirsch, P.: An soc with two multimedia dsps and a risc core for video compression applications. In: *Digest of Papers, IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, February (2004).
19. The International Technology Roadmap for Semiconductors (ITRS), (2007).
20. Heo, S., Barr, K., Asanovic, K.: Reducing power density through activity migration. In: *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 217–222, Seoul, Korea, August (2003).
21. Srinivasan, J., Adve, S.V., Bose, P., Rivers, J.A.: The impact of technology scaling on lifetime reliability. In: *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, Florence, Italy, June (2004).
22. Skadron, K., Sankaranarayanan, K., Velusamy, S., Tarjan, D., Stan, M.R., Huang, W.: Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans Arch Code Optim* **1**(1), 94–125, March (2004).
23. Sankaranarayanan, K., Velusamy, S., Stan, M.R., Skadron, K.: A case for thermal-aware floorplanning at the microarchitectural level. *J Instr-Level Parallelism* **7**, October (2005).
24. Lasance, C.J.M.: The urgent need for widely-accepted test methods for thermal interface materials. In: *Proceedings of the 19th IEEE SEMI-THERM Symposium*, pp. 123–128, San Jose, CA, USA, (2003).
25. Han, Y., Koren, I., Moritz, C.A.: Temperature-aware floorplanning. In: *Proceedings of Workshop on Temperature-Aware Computer Systems (TACS)*, Madison, WI, USA, (2005).
26. Li, Y., Lee, B., Brooks, D., Hu, Z., Skadron, K.: CMP design space exploration subject to physical constraints. In: *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Austin, TX, USA, (2006).
27. Chaparro, P., Gonzalez, J., Magklis, G., Cai, Q., Gonzalez, A.: Understanding the thermal implications of multicore architectures. *IEEE Trans Parallel Distributed Syst* **18**(8), 1055–1065, (2007).
28. Memik, S.O., Mukherjee, R., Ni, M., Long, J.: Optimizing thermal sensor allocation for microprocessors. *IEEE Trans Comput Aided Design* **27**(3), 516–527, March (2008).
29. Sharifi, S., Simunic Rosing, T.: Accurate temperature sensing for efficient thermal management. In: *Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED)*, San Jose, CA, USA, (2008).
30. Memik, S.O., Mukherjee, R.: An integrated approach to thermal management in high-level synthesis. *IEEE Trans VLSI* **14**(11), 1165–1174, November (2006).
31. Yang, S., Wolf, W., Vijaykrishnan, N., Xie, Y.: Reliability-aware SOC voltage islands partition and floorplan. In: *Proceedings of IEEE Annual Symposium on VLSI (ISVLSI)*, Montpellier, France, (2006).
32. He, Z., Peng, Z., Eles, P., Rosinger, P., Al-Hashimi, B.M.: Thermal-aware SoC test scheduling with test set partitioning and interleaving. *J Electron Testing Theory Appl* **24**(1–3), 247–257, (2008).