Title

FAST STATIC AND DYNAMIC GRID LEVEL THERMAL SIMULATION CONSIDERING TEMPERATURE DEPENDENT THERMAL CONDUCTIVITY OF SILICON

Author

Ziabari, Amirkoushyar

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

FAST STATIC AND DYNAMIC GRID LEVEL THERMAL SIMULATION CONSIDERING TEMPERATURE DEPENDENT THERMAL CONDUCTIVITY OF SILICON

A dissertation submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Amirkoushyar Ziabari

June 2012

The Dissertation of Amirkoushyar Ziabari is approved:

Professor Ali Shakouri, Chair

Professor Jose Renau

Professor Joel Kubby

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Content

List of Figures v

List of Table vii

Abstract viii

Acknowledgement x

1 Introduction 1

2 Power Blurring Method 5

2.1 Related Works………..………………………………………………… 5
2.2 Steady-State Power Blurring.…………………………………………..... 10
2.2.1 Methodology……………………………………………………..... 10
2.2.2 Full Chip Package Model………………………………………….. 12
2.2.3 The Thermal Mask………………………………………………..... 13
2.3 Steady-State Case Study……………………………………………….... 21
2.3.1 Error Metric………………………………………………………... 21
2.3.2 Case Study……………………………………………………….…. 22
2.4 Transient Power Blurring……………………………………………...... 25
2.4.1 Old Algorithm for Transient Methodology……………………….... 25
2.4.2 New Algorithm for Transient Methodology………………….…..... 28
2.5 Transient Analysis Case Studies………………………………………... 32
2.6 Summary………………………………………………………………... 35

3 Architecture Level Thermal Simulators 38

3.1 HotSpot……………………………………………………………………... 39

3.2 SESCTherm…………………………………………………………….... 39

3.3 PB vs. HotSpot vs. SESCTherm……………………………………………. 41

3.3.1 Error Metrics……………………………………………………………. 41

3.3.2 Methodology Validations……………………………………….... 41

3.3.3 A Static Case Study……………………………………………… 43

3.3.4 Transient Case Study I…………………………………………… 48

3.3.5 Transient Case Study II (A real workload)……...………………... 52

3.4 Summary…………………………………………………………………… 57

4 Adaptive Power Blurring 59

4.1 PB and APB Methods……………………………………………….... 59

4.2 Comparison Results and Discussions………………………………... 63

4.2.1 The Thermal Masks Table……………………………………... 63

4.2.2 Case Studies………………………………………………….…..... 64

4.3 Validation of ANSYS Nonlinear Simulations………………………….. 69

4.4 Summary…………………………………………………………….... 70

5 Conclusion 71

2.1 Contributions……………………………………………………………….. 71

2.2 Future Works……………………………………………………………….. 73

A List of Publications 74

Bibliography 75

List of Figures

2.1 The effect of thermal mount geometry on the shape and the average value of the temperature profile…………………………………………………….... 9
2.2 Schematic of a typical flip chip package assembly……………………….. 13
2.3 The thermal mask…………………………………………………….. 16
2.4 Method of Image…………………………………………………....... 17
2.5 Intrinsic Error………………………………………………………… 21
2.6 A typical power map and its corresponding temperature profile……….... 23
2.7 Static Comparison between PB and ANSYS (40×40 grid)………………... 24
2.8 Static Comparison between PB and ANSYS (80×80 grid)………………... 24
2.9 Transient thermal basis……………………………………………….. 27
2.10 Schematic overview of transient power blurring method (old algorithm)… 28
2.11 The new algorithm for transient power blurring…………………………… 30
2.12 Input power for transient case study 1……………………………………… 32
2.13 Thermal profiles obtained for case study 1………………………………… 35
2.14 Evolution of transient hot spot over the chip for case study 1..…………… 36
2.15 Evolution of relative error over time for case study 1....…………………... 36
2.16 Input power for transient case study 2……………………………………… 37
2.17 Evolution of transient hot spot over the chip for case study 2..…………… 37

3.1 The floorplan and power map for steady-state case study……………..... 44
3.2 A Static comparison between ANSYS, PB, HotSpot, and SESCTherm…… 46
3.3 A Static comparison between ANSYS, PB, HotSpot, and SESCTherm with nominal value of heat transfer coefficient………………………………….. 48
3.4 Transient case study comparison between the four methods………………. 49
3.5 Evolution of absolute temperature error in the hot spot over the chip……... 52
3.6 Effect of averaging power trace on evolution of hot spot temperature…….. 54
3.7 Evolution of hot spot and average temperature over time: A comparison between HotSpot and PB for gcc workload.………………………………… 55

3.8 Evolution of maximum and average dissipated power in blocks over time... 55
3.9 Effect of truncation of thermal mask……………………………………….. 56

4.1 Silicon thermal conductivity vs. temperature……………………………… 60
4.2 Adaptive Power Blurring (APB_I)…………………………………………. 62
4.3 Adaptive Power Blurring (APB_II)…...……………………………………. 62
4.4 3D view of four sample power dissipation maps…………………………… 65
4.5 Temperature profile (ANSYS)……………………………………………… 68
4.6 Cross sectional comparison between temperature profiles………………… 68

List of Tables

2.1 Analogy between image processing and power blurring ………………..... 12
2.2 Material properties and dimensions of the package model……………….. 14
2.3 A comparison between power blurring and ANSYS results (Static)……… 25

3.1 Material properties and dimensions of the package model……………….. 43
3.2 Static comparison between PB, HotSpot, and SESCTherm………………. 46
3.3 Transient comparison between PB, HotSpot, and SESCTherm for Case1... 51
3.4 Execution time of HotSpot and PB for real workload……………............ 57

4.1 Comparison between PB, APB_I, and APB_II……………………………. 67

Abstract

Fast Static and Dynamic Grid Level Thermal Simulation Considering Temperature Dependent Thermal Conductivity of Silicon

by

Amirkoushyar Ziabari

Power blurring has been developed to calculate temperature profiles in VLSI ICs, in both steady-state and transient regimes. It is illustrated that PB is very fast and accurate in steady-state thermal analysis. However, the transient method was inefficient when it was handling the temperature analysis of real workloads. A new algorithm for transient power blurring has been developed to calculate temperature profiles of real workloads. It is shown that the method can be utilized for thermal simulation at both device and architecture level. A detailed comparison is performed between the power blurring method and two of the commonly used architecture level thermal simulators, HotSpot and SESCTherm simulators. We take into account both the accuracy of the thermal simulation and the computational speed. The results indicate that Power Blurring has the potential to be a promising architecture level thermal simulator for fast calculation of the temperature profile from the input power map in a realistic package which, in turn, is a key ingredient for full self-consistent simulations.

Additionally, power blurring is extended so that it considers the temperature dependent thermal conductivity of silicon chips in ICs. Temperature rises of 40-50°C on the chip will reduce the thermal conductivity of silicon by 10-20%. This could affect the hot spot temperature by 5-7°C. In this work the PB approach, which is based on the superposition principle, is extended to account for the temperature-dependent material properties. Two Adaptive Power Blurring (APB) methods based on iterative procedures are proposed. In both methods, PB provides an initial temperature distribution estimation using the room temperature Si thermal conductivity. Good convergence is achieved in only 2-3 iterations in both methods. It is demonstrated that these APB methods substantially improve the accuracy at high temperature rise values, in particular at hot spots, while still being much faster than traditional finite element modeling (FEM) computations.

Acknowledgement

First and Foremost, I would like to express my thankfulness to professor Shakouri for all his supports and guidance throughout the course of my work. I am not saying this because everybody does that. I am saying this because I hear his voice whenever I am doing a presentation, working on a paper, reading a paper, scheduling a work, executing a simulation, performing an experiment, and so on. I am thankful for all I have learned and experienced under his excellent supervision.

I am also grateful to fellow Quantum Electronic Group members and UCSC Colleagues who have been a pleasure to work with. In particular, Je-Hyoung Park, Xi Wang, Ehsan K. Ardestani, Zhixi Bian, Bjorn Vermeersch, and Gilles Pernot for all their help, feedback, and useful discussions.

I would like to express my gratitude to professor Renau and professor Kubby for reading my thesis and providing me with their feedback.

Last but not least, I would like to thank my loving fiancé, Maryam, who calms me with her encouraging words. I would also like to thank my family, without them I never could have come this far in my life.

Chapter 1

Introduction

Increasing power density due to the shrinking of transistor size and higher integration levels in the CMOS VLSI has resulted in elevated temperature in integrated circuit (IC) chips. Temperature non-uniformity across the chip leads to hot spots. This effect and the elevation in temperature are critical issues because of their impact on both the performance and the reliability of IC chips [1].

In addition, the increasing leakage power and its exponential dependence on the temperature require more attention to thermal-aware simulations and optimizations. Hence, a precise thermal profile is essential for an accurate analysis of performance, reliability, and power management.

Generally, thermal simulations and design optimizations are done under steady-state worst case conditions. Due to high computational cost, this results in the use of conservative margins in thermal designs. However, temperature non-uniformity evolves over time and hot spots can be transient. As the thermal budget becomes increasingly tight, the worst case approach becomes too costly and inefficient. Even with the state-of-the-art tools, chip-level transient thermal simulations with a realistic package configuration are too expensive for physical design optimization or performance verification in the packaged environment. Additionally, in the early stages of chip design, specific package information and thermal boundary conditions may not be available. In such cases, a fast thermal analysis method is highly desired [2]. Fast thermal simulations also allow optimization of the chip during architecture evaluation.

Our experiments show that in a fast power and thermal simulation for a processor, the thermal simulation takes up to 20% of the time budget for a single core system. A multicore system might need even more time for thermal simulation. Hence, accuracy as well as speed of the thermal model directly impact the productivity in the evaluation phase of processor design.

The Power Blurring method (PB) has been introduced as an ultrafast and accurate method to calculate the temperature profile in IC chips through convolution of the power map of the device under investigation and a thermal mask. This mask conforms to the impulse response of the system and represents how much of a temperature rise takes place on the surface of a die due to a unit point heat source. The significance of the method is not only because of its ability to calculate the temperature profiles by two orders of magnitude (100 times) faster than finite element methods (FEMs), but also due to its advantage over the other current Green's function based techniques by considering the real geometry of packaged VLSI chips [3]. The Power Blurring method is further improved to predict IC chip temperature with high spatial resolution, which has been prohibitively expensive with conventional methods [4]. Using the PB method, transistor level thermal maps (5×5 μm^2 grid) of a 5×5 mm^2 chip with a computation time of 20 seconds have been obtained. Chip-level transient thermal simulation is another area that has been tackled by the PB method. With a minor adjustment, the time evolution of the thermal mask resulting from a spatiotemporal impulse is employed for transient simulations. It has been shown that the PB method can estimate the temperature profiles with error less than 3% with a computation speed one hundred times faster than the industry standard finite element tools [5]. Despite all the advantages mentioned, there were some limitations for the power blurring method that are addressed and solved in this work.

First, for a long transient full power trace thermal analysis, power blurring needs lots of memory to perform the analysis and record the data while calculating temperature results. Therefore, it was not practically efficient to put PB in real applications such as transient thermal analysis with a real workload at the architecture level, and thermal aware design and floorplaning. Second, since power blurring is a linear method based on superposition principle, it was not able to incorporate the change in thermal conductivity of silicon due to large temperature variations. In this work, the model is developed and extended so that it can surmount these limitations.

The rest of this thesis is organized as follows. In chapter two, the power blurring method for both steady-state and transient thermal analysis is introduced and the new algorithm for transient thermal analysis is presented. In chapter three, a comparison between power blurring and two of the commonly used architecture level thermal simulators, namely HotSpot and SESCTherm simulators, in both steady state and transient regimes has been performed. The significance of the new algorithm for transient thermal analysis is highlighted in this chapter. In chapter four, the power blurring method has been extended so that it considers, in a self-consistent manner, the dependence of silicon's thermal conductivity on temperature. Two new algorithms are presented and their corresponding methods are named "Adaptive Power Blurring methods" one and two (APB_I & APB_II), respectively. A detailed comparison between the results obtained by the three methods, APB, the PB and ANSYS Finite Element Modeling software [6], is carried out in this chapter. Finally, chapter 5 concludes this thesis by presenting a summary and proposing some future possibilities for continuation of this work.

Chapter 2

Power Blurring Method

In this chapter, a brief overview of thermal simulation techniques for VLSI ICs is given. The necessity of a fast and accurate thermal analysis method is highlighted, and power blurring as an alternative method is introduced. Power blurring methodologies for acquiring steady-state and transient temperature information in ICs are described. The new and old transient methodologies are compared for a simple case study.

2.1 Related Works

To obtain thermal profiles for a region of interest, the heat diffusion equation shown in Equation (2.1) has to be solved with conditions imposed on boundaries [7].

$$k\frac{\partial^2 T}{\partial x^2}+k\frac{\partial^2 T}{\partial y^2}+k\frac{\partial^2 T}{\partial z^2}+q^*=\rho c_p\frac{\partial T}{\partial t} \qquad (2.1)$$

Here, k is the thermal conductivity (W/m·K), ρ is the density (kg/m^3), c_p is the specific heat (J/kg·K), q^* is the heat generation per unit volume (W/m^3), and T is the temperature of the location (x, y, z) at time t. In IC thermal analysis, the heat diffusion equation has been conventionally handled by grid-based methods, such as Finite Difference Method (FDM) or Finite Element Method (FEM), which generate 3-D volumetric meshes of solid structures [8]. The accuracy of these FDM and FEM comes at the price of long execution times, and exhaustive CPU and memory usage. Since the computation time rises significantly with the number of meshed elements, these approaches are unpractical if we are to integrate them in an interactive place-and-route IC design program. Besides, finite element analysis programs require a thorough design of the meshing to attain convergence. This procedure cannot be easily automated for a complex geometry or power dissipation loads. A Green's function-based method was introduced for thermal analysis as an alternative [2], [8]. This method has an advantage over FEM or FDM owing to its lower dependence on the volumetric mesh, but nevertheless, the analytical expression of the Green's function could be found only for simple 1D planar geometries and requires an infinite series. The Green's function for infinite multilayered structures has been calculated analytically by several groups and is readily available online. For example, in [9], a fast algorithm for calculating temperature profiles based on analytical Green function of the bounded box is presented. However, most of these techniques fail to handle the real geometry of the heat spreader underneath the chip. All the previous literatures assume the heat spreading layers have the same dimensions as the silicon die.

However, in fact, the heat spreading layer closer to the heat source has smaller surface area than the one farther away, similar to a pyramid. This configuration is important for heat spreading to the heat sink and neglecting it can result in an overestimation of the temperature of the chip. Besides, the shape of the temperature profile is also affected significantly by the size of the heat sinks. Whereas the average chip temperature can be scaled by an appropriate scaling of the convection coefficient, the change in temperature profile is not that straightforward [8]. Figure 2.1 shows how the real geometry of the package can affect the temperature profile. In this figure, the temperature fields, shown in part (a) and (b), result from applying the power map, shown in part (c) on two geometries drawn in part (d). The geometries contain a Si die on the top, a heat spreader in the middle (Cu), a heat sink in the bottom (Cu), and two layers of Thermal Interface Materials (TIM) in between these materials to bind them together. This structure is discussed in more detail in the next section 2.2.1. As it can be seen in Figure 2.1d, the two geometries are different such that for the Dice geometry, all the layers have the same size as the silicon die, while in Pyramid geometry, heat spreader and heat sink are about 10 and 50 times larger in surface than the chip. Even though perfect convection is assumed on the top surface of heat sink in both cases so that the total convection remains the same for both chips, the average temperature rise will be significantly different, since the mounts are different in shape. Bagnoli has managed to find an analytical system of equation that relates the temperature and heat flux at the material interfaces of the pyramid structure [10]. These equations can be discretized and solved by the usual matrix inversion. While this technique can be applied to the pyramid multilayer structure, it still requires significant computational resources. The main drawback of tackling the more realistic pyramid geometry is that a simple analytic expression no longer exists and we must resort using semi-analytical methods where a part of algorithm must rely either on empirical parameters or previous simulations if we want to be reasonably fast. Acceleration techniques for grid-based methods are introduced in [11] and [12], in which the acceleration was achieved by either decomposing a multiple dimensional problem into 1-D problems or reducing the thermal network.

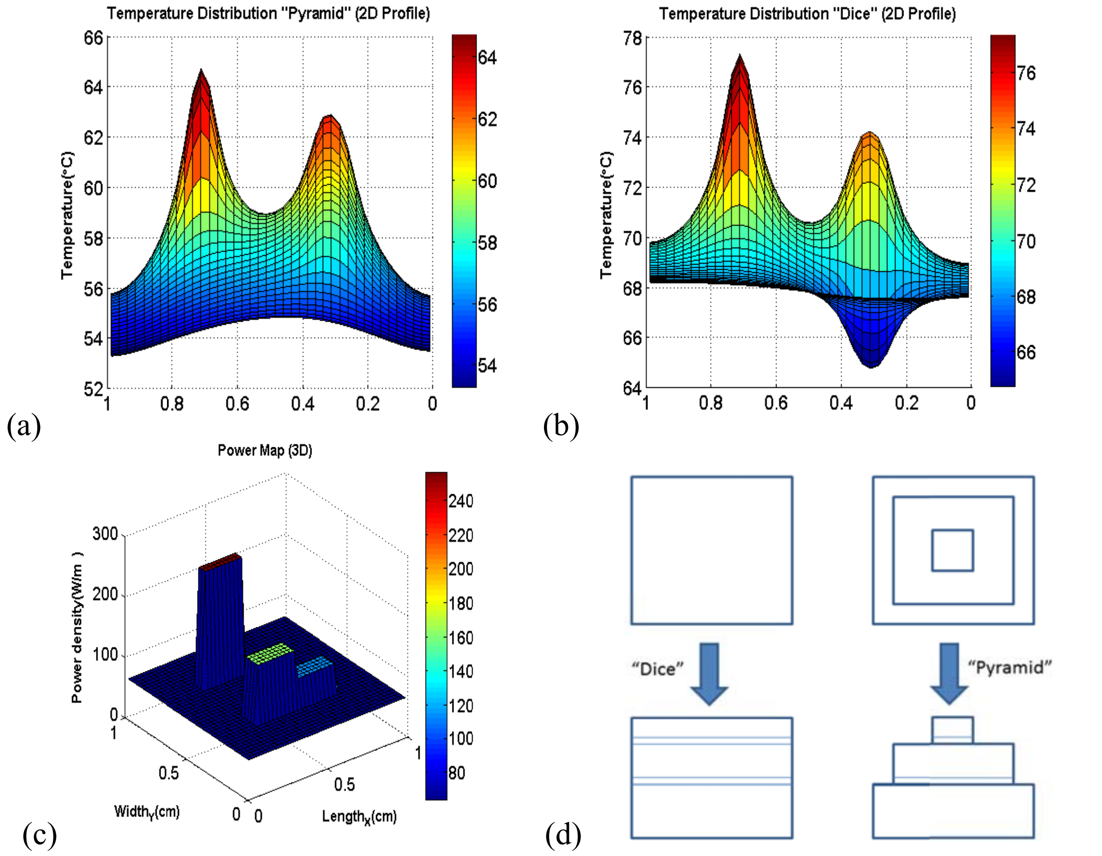


Figure 2.1: The effect of the thermal mount geometry on the shape and the average value of the temperature profile (a) Temperature Profile obtained by Dice geometry (b) Temperature Profile obtained by Pyramid geometry (c) Applied power map (d) "Dice" and "Pyramid" geometries (Dimensions are presented in Table 2.2).

However, both failed to include a realistic package model. Ignoring the lateral heat spreading in realistic packages can result in overestimated chip temperatures [13]. In [14], the Diffusive Representation (DR) is used to construct a compact thermal model as a state-space representation. The method is dedicated to the representation of nonrational systems based on infinite contributions. A discrete formulation of the DR yields a practical engineering modeling method. DR enables modeling of a given system, provided that the temperature to be monitored is observable for the purpose of prior identification. The main drawback of the method is that it cannot construct geometry dependent models. The formal construction of thermal compact models from the geometry description of the system is limited to regular and simple shapes. In the following section, the Power Blurring method is described, which resolves many of these limitations and it could predict the static and dynamic temperature profiles of IC chips fast and accurately.

2.2 Steady-State Power Blurring

2.2.1 Methodology

The "Power Blurring (PB)," method has its theoretical basis on the Green's function method. Implementation is similar to image blurring used for image processing [15]. The Green's function method finds a solution to the partial differential equation with a point source as the driving function in the first step ($G(\vec{r},\vec{r'})$). This solution is called the "Green's function", which is equivalent to the impulse response of a system. Subsequently, a solution to an actual source is represented as a superposition of the impulse responses to the point sources at different locations [16]. This is expressed in the Equation (2.2), in which V′ is the volume over which the heat $(q(x',y',z'))$ is generated.

$$\iiint_{V'} G(\vec{r},\vec{r'})q(x',y',z')dv' \qquad (2.2)$$

Thus, the Green's function is a building block to construct an actual solution.

In image processing, an image is blurred by a convolution with a filter mask. The filter mask is a matrix whose elements define a process in which the modification (i.e. blurring) of an image occurs. For instance, an image, *f*, is convoluted with a filter mask, *w*, to produce a blurred image, *g,* by Equation (2.3).

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t) \qquad (2.3)$$

where *a=(m-1)/2* and *b=(n-1)/2* for a $m \times n$ mask [15].

In thermal analysis, the impulse response (i.e. Green's function) of a system corresponds to a heat-spread function, namely, its thermal mask. The thermal mask represents how much temperature rise takes place in a solid due to a unit point heat source. The power map for a given IC chip can be estimated using voltages and currents in each device or in circuit blocks [17]. If we think of the power map as an image, *f*, the thermal profile of the IC chip resulting from its power map can be regarded as a blurred image, *g*, when the filter mask, *w*, conforms to the thermal mask.

In principle, once the thermal mask is obtained for a given package assembly, a temperature profile can be easily obtained by a simple matrix convolution for an arbitrary power distribution map. Table 2.1 summarizes the analogy between the image blurring and PB methods.

TABLE 2.1
ANALOGY BETWEEN IMAGE PROCESSING AND POWER BLURRING.

Image Processing	Power Blurring
Image (f)	Power Map
Filter Mask (w)	Impulse Response
Blurred Image (g)	Temperature Profile

2.2.2 Full-Chip Package Model

Figure 2.2 illustrates a cross-sectional view of a typical flip chip package structure along with our simplified thermal model, which is composed of three main components: the Si die, heat spreader (Cu), and heat sink (Cu). The Heat spreader improves heat transfer between the Si die and heat sink. A thermal interface material (TIM) serves as a bonding material and also enhances thermal conductivity at the interface [18]. The bottom layer of the die, where devices are fabricated, is very thin compared to the substrate of the die. Hence, the die is considered bulk silicon. ICs have passivation layers between interconnects [19], and thus, the thermal resistance of the minor heat transfer path is much higher than that of the major heat transfer path (through the substrate). Most of the heat is assumed to be transferred into the ambient

(35°C) by conduction and convection along the major heat transfer path, and other minor heat transfer paths are neglected. To this end, adiabatic boundary conditions are imposed on the four sides and the bottom surface of the Si die. The material properties and dimensions are listed in Table 2.2. The die thickness is assumed to be 0.775mm as most chips in 90nm copper CMOS technology.

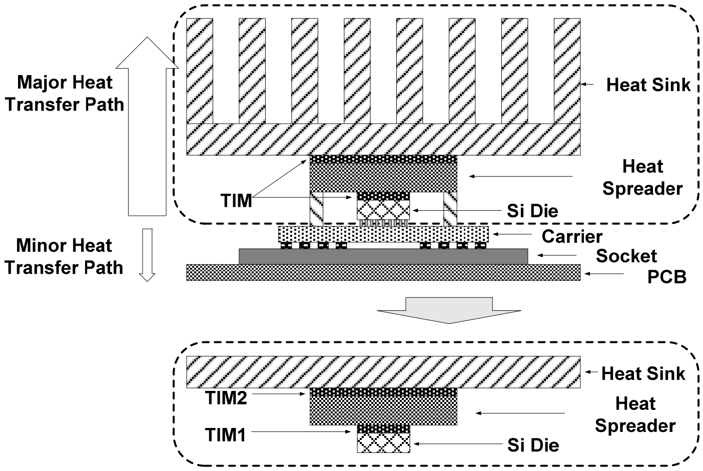


Figure 2.2: Schematic of a typical flip chip package assembly (adapted from [18]) and our simplified thermal model.

2.2.3 The Thermal Mask

As mentioned earlier, the thermal mask is an impulse response of a system in the space domain. According to the Green's function method, we can build a solution to a partial differential equation with an arbitrary driving function once we have a solution corresponding to an impulse (a point source). In thermal analysis of IC chips, temperature distribution is the physical quantity of interest, and heat (i.e. power consumption in ICs) is the driving function. Thus, the thermal mask conforms to a steady-state temperature distribution induced by a point heat source, which is applied to the center of the Si die. In practice, the die area is discretized into grids and an approximate delta function simulating a point heat source is applied to the center element of the grid. Subsequently, the difference between the resulting temperature and ambient temperature is normalized with the amount of the input power. Although the thermal mask can be obtained in analytical form for a simple 1D geometry, measurement or 3D finite element analysis (FEA) simulator such as ANSYS [6] is needed for a realistic structure shown in Figure 2.3, where heat spreading is important.

TABLE 2.2
MATERIAL PROPERTIES AND DIMENSIONS OF THE PACKAGE MODEL [23].

	Area (mm^2)	Thickness (mm)	Thermal Conductivity (W/m-K)	Density (kg/m^3)	Specific Heat (J/kg-K)
Si Die	10×10	0.775	117.5	2330	700
TIM1	10×10	0.025	5.91	1930	15
Heat Spreader	28×28	1.8	395	8933	397
TIM2	28×28	0.025	3.5	1100	1050
Heat Sink	60×80	6	395	8933	397

Figure 2.3 shows the thermal mask for the given package model. Its units are in

thermal resistance (°C/W), and hence, the thermal mask generates a temperature profile when it is convoluted with a power distribution map (W).

The shape of the thermal mask depends on the location of the point heat source. In other words, point heat sources at a corner and at an edge of the Si die surface produce different temperature profiles, and hence different thermal masks. For those regions which have proximity to the boundaries, the thermal mask shown in Figure 2.3 is inappropriate for the convolution. This source-location-dependence of the thermal mask prevents using a single thermal mask for the convolution with a power map.

For simplicity, it is desirable to avoid dividing the chip into “arbitrary” regions at the edges and corners. In the following section, a method to reduce errors from a single thermal mask will be discussed in detail.

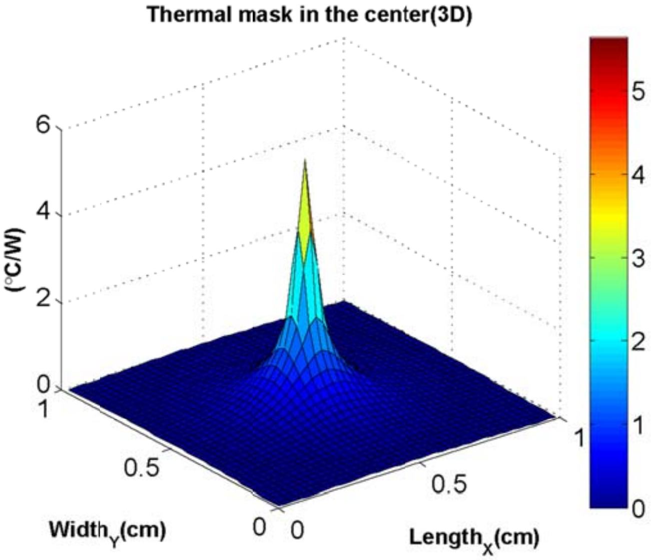
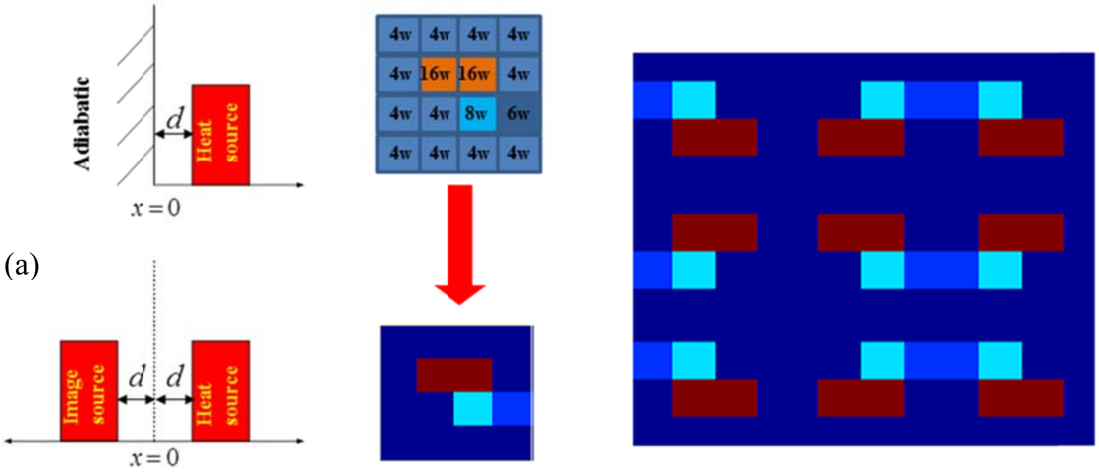


Figure 2.3: The thermal mask; the surface of the Si die is discretized into 40×40 grids.

A. Method of Image

Thermal problems are often converted into electrical problems based on duality to take advantage of well-developed analysis methods in electrical systems. One of these is the Method of Image from electromagnetism [21]. Electromagnetic problems associated with a planar perfect electric conductor can be solved through the Method of Image, in which the boundary is replaced by mirror image (equivalent) sources with appropriate signs [16, 21].



4w	4w	4w	4w
4w	16w	16w	4w
4w	4w	8w	6w
4w	4w	4w	4w

Figure 2.4: Method of Image. (a) Method of Image (b) An arbitrary power map on Si die and its eight nearest neighbor mirror images.

A similar principle can be applied to the thermal problems involving adiabatic boundary conditions. Consider the case of a heat source located at a distance *d* from an adiabatic surface as shown in Figure 2.4a. No heat transfer can occur at the adiabatic boundary. If we replace the adiabatic boundary with an image source, net heat flux at *x=0* is zero, and the boundary condition is satisfied. However, the problem becomes more manageable in this approach.

In our thermal package model, adiabatic boundary conditions are imposed on four sides of the Si die. Thus, the power distribution on the Si die can be extended with mirror images, as shown in Figure 2.4b. In this way, a single thermal mask shown in Figure 2.3 can be employed for the direct convolution, and the edge effect is taken care of simultaneously. The disadvantage is that now the size of the power dissipation matrix is nine times bigger. Since the spatial convolution could be done very fast in the Fourier domain, this is not a factor which will limit the PB computation time significantly. Additionally, even though the new matrix is 9 times bigger, mathematically, only two thirds of this matrix is needed for the final temperature calculation [15].

$$P_{image} = [P(x,y) + P(x,-y) + P(-x,y) + P(x,2W-y) + P(-x,-y) + P(-x,2W-y) + P\ (2L-x,y) + P(2L-x,-y) + P(2L-x,2W-y)] \quad (2.4)$$

Here, P_{image} is the extended power map, P is the power map, and W and L are the width and the length of the power map, respectively. P_{image} extends from (-W, -L) to (2W, 2L) which is 9 times larger than P. For the final calculation of the temperature profile two thirds of this matrix is needed.

The direct convolution result is given by

$$T(x,y) = h(x,y) * [P_{image}(-\tfrac{L}{2}:\tfrac{3L}{2}, -\tfrac{W}{2}:\tfrac{3W}{2})] \quad (2.5)$$

where, T is the thermal profile and h is the thermal mask. The term in brackets is the portion of the extended power map needed for the final temperature calculation. In the post-process, the center region of the thermal profile corresponding to the original power map needs to be retrieved.

B. Intrinsic Error Compensation

For an accurate IC temperature calculation, the thermal profile obtained by the direct convolution with Method of Image (DCMI) requires an additional step, namely, the intrinsic error compensation. Due to the pyramid structure of our thermal package model, 3D heat spreading plays an important role in heat transfer. In spite of adiabatic boundary conditions imposed on the four sides and the bottom surface of the die, those regions along the die perimeter have better chances of heat removal than the center region. This situation can be more clearly represented with an example of a uniform power distribution.

Consider a uniform power distribution shown in Figure 2.5a. The thermal profile corresponding to the power distribution should be bell-shaped. A FEA simulation result predicts such a profile shown in Figure 2.5b. However, DCMI generates a uniform temperature profile shown in Figure 2.5c. The temperature difference between Figure 2.5b and Figure 2.5c is shown in Figure 2.5d. This deviation is an artifact of the Method of Image, since this method does not take into account the larger size of the heat spreader and the heat sink with respect to the silicon die. When the Method of Image is applied to the uniform power map, the resulting power map is another uniform power map with enlarged dimensions.

Although the thermal mask (the impulse response) is obtained for a three dimensional geometry, the convolution is processed in two dimensions. The effect of 3D heat spreading is not appropriately handled with DCMI. Thus, the temperature deviation along the perimeter of the die is intrinsic to DCMI, and these intrinsic errors need to be compensated for to obtain the final result.

The temperature rise due to uniform power input is linearly proportional to the input power level. Therefore, the intrinsic error in Figure 2.5d is a linear function of the input power level. On the other hand, the relative deviation given in Equation (2.6) below is a constant function regardless of the input power level.

$$E_r = (T_{DCMI} - T_{ANSYS})/T_{ANSYS} \qquad (2.6)$$

T_{DCMI} and T_{ANSYS} are thermal profiles for a uniform power map obtained by DCMI and ANSYS, respectively. Both $\mathrm{T_{DCMI}}$ and $\mathrm{T_{ANSYS}}$ represent the difference of the temperature profile of the chip and ambient temperature. E_r can serve as a position-dependent scaling factor, and it can be employed for any kind of power dissipation profile. After calculation of E_r, the final temperature can be obtained using the following formula.

$$T_{finall} = \frac{T_{DCMI}}{1+E_r} + T_{ambient} \qquad (2.7)$$

Equation (2.7) obviously compensates for the error from the Method of Image.

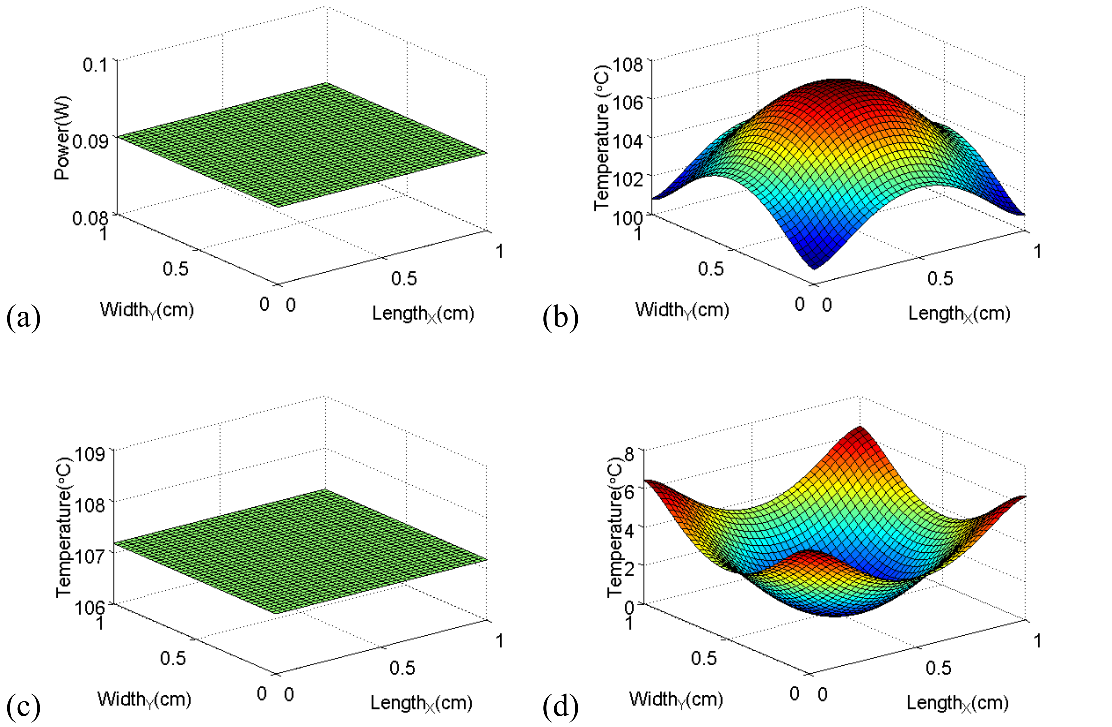


Figure 2.5: Intrinsic error: (a) uniform power map; (b) thermal profile by ANSYS; (c) thermal profile by the PB method; (d) temperature deviation.

2.3 Steady-State Analysis Case Study

2.3.1 Error Metrics

In order to study the accuracy of each method, we calculated the relative error compared to that of ANSYS, which is a standard FEM tool for thermal analysis, using Equation (2.8). In all the simulations, the ambient temperature is set to 35ºC.

$$Err = (T_{\text{Method}} - T_{\text{ANSYS}}) / (T_{\text{ANSYS}} - T_{\text{Ambient}}) \quad (2.8)$$

Max. Error: For each grid across the entire chip, Equation (2.8) is used to calculate the error and then the maximum error is reported.

Hot-spot Error: Equation (2.8) is used to calculate the error in the hottest spot in the temperature profile.

Average Error: Equation (2.8) is applied to the average temperature across the chip.

Absolute Temperature Error range: This error is obtained using Equation (2.9) and then the range is reported.

$$\mathrm{Err} = (T_{\mathrm{Method}} - T_{\mathrm{ANSYS}}) \quad (2.9)$$

2.3.3 Case Study

The PB method is implemented on a PC (3.2 GHz Core 2 CPU, 2GB memory) using MATLAB [22]. The input power map for an IC (1×1 cm^2) is given in Figure 2.6a. The results of the proposed methodology have been validated against FEA software, ANSYS, which has been widely used in the industry. Simulations for two different grid sizes, 40×40 and 80×80, are performed.

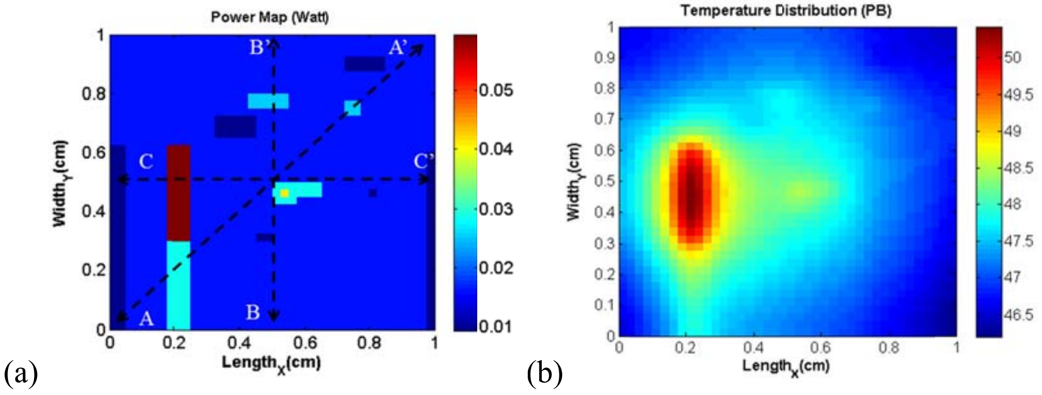


Figure 2.6: (a) A typical power map of an IC (b) the corresponding thermal profile

Figure 2.6b shows the thermal profile obtained by the power blurring method using the power map shown in Figure 2.6a. The PB method results and the ANSYS simulation results are compared in Figures 2.7 and 2.8. The results shown in Figures 2.7 and 2.8 are for 40×40 and 80×80 grid sizes, respectively. Figures 2.7a and 2.8a are thermal profiles along A-A' path specified in Figure 2.6a. Figures 2.7d and 2.8d provide the overall relative temperature error. As it can be seen, the PB method renders accurate thermal profiles with maximum temperature error of less than 0.85% with respect to ANSYS simulation results. ANSYS parameters for these simulations are set according to Table 2.2. In all simulations, convection between the top surface of heat sink and air is considered using a convection coefficient of 0.15(W/m^{2}K).

In Table 2.3, a comparison between ANSYS and PB is presented. In this table, the execution time of each of these methods for different meshing size, as well as maximum error, error in the hottest spot, and average error of results obtained by PB relative to ANSYS are shown.

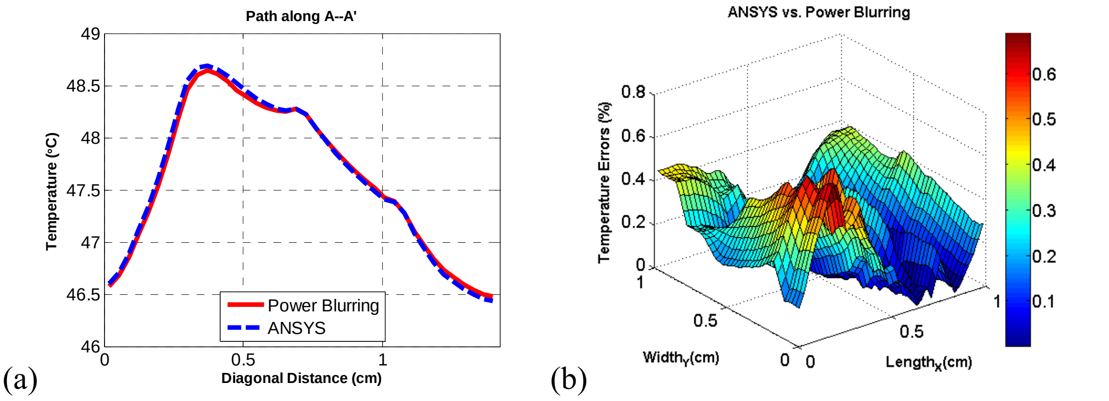


Figure 2.7: Comparisons between the Power Blurring method and ANSYS results (40×40 grid size): (a) Thermal profile along A-A'; (b) the overall relative error.

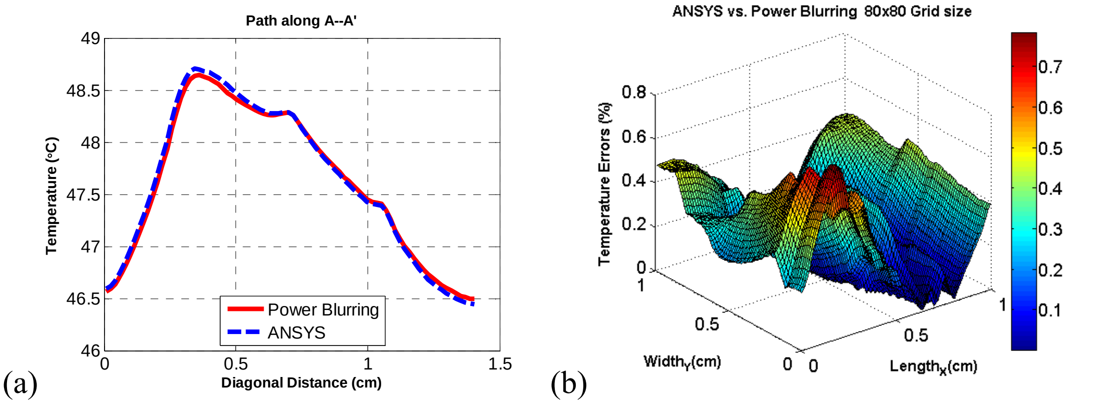


Figure 2.8: Comparisons between Power Blurring method and ANSYS results (80×80 grid size): (a) Thermal profile along A-A'; (b) the overall relative error.

The advantage of the computation time reduction becomes more prominent when we conduct simulations with finer grid size power maps. For example, as it can be seen in Table 2.3, although PB in a finer grid size (80×80 meshed elements) runs slower than smaller grid size (40×40 meshed elements) with a very slight change (.016 s), the ratio of computation time reduction increases from 345 times faster (for 40×40) to 650 times faster (for 80×80), while the maximum relative error is under 0.8%. This is because the number of elements in a volume grid increases significantly, while a Fast Fourier Transform (FFT) for spatial convolution is very fast [4].

TABLE 2.3
A COMPARISON BETWEEN THE POWER BLURRING AND ANSYS RESULTS.

No. of meshed elements	ANSYS execution Time	PB execution Time	Max. error	Hot-spot error	Average error
(40×40)	14 s	0.035 s	0.78%	0.47%	0.27%
(80×80)	33 s	0.051 s	0.85%	0.47%	0.29%

2.4. Transient Power Blurring

2.4.1 Old Algorithm for Transient Methodology

For steady-state thermal simulation, the thermal mask is obtained with a spatial impulse (i.e. point heat source). The PB method can be applied to the transient thermal simulation with a minor adjustment. The difference is that the time evolution of the thermal mask resulting from spatiotemporal impulse is employed for a transient simulation. To obtain an impulse response in the time domain, a delta function needs to be applied to the center of the die. In practice, a point heat source is applied for a very short time period (approximate delta function), and the corresponding thermal response is recorded at each time step, which is shorter than the width of the approximate delta function. The width of the delta function is determined by the desired level of temporal resolution. The resulting thermal responses are normalized with respect to the amount of applied power. The series of the thermal masks acquired at the end of this procedure constitutes a transient thermal mask (i.e. time evolution of the thermal mask) [5].

Once the transient thermal mask is prepared, the transient temperature profile is obtained through superposition principle. The convolution of the transient thermal mask with a given power map at each time step creates a series of thermal profiles, which makes up a transient thermal basis, it can be understood as the time evolution of a thermal profile resulting from the input of the specific power map for the corresponding time period (i.e. the width (T) of the delta function). It serves as a building block for a pulse input function over a longer duration. The overview of the process is depicted in Figure 2.9. The Method of Image and intrinsic error compensation mentioned previously are applied at each and every time step when the convolution is performed.

When a VLSI chip performs tasks, the on-chip power map varies in time. In time the domain, the activity pattern of a VLSI chip power can be represented by a series of rectangular pulses with various duration, during which the power map is fixed. Consider a case where a chip is active for a certain time period (D) with fixed power dissipation as shown in Figure 2.10. This pulse can be regarded as a group of consecutive approximate delta functions. To obtain a transient thermal profile corresponding to the rectangular pulse with duration D, the transient temperature basis is shifted in time domain accordingly to fit the pulse duration. The overall transient temperature is then the aggregate of those transient temperature bases. This approach can be applied for any kind of chip activity pattern and corresponding power map.

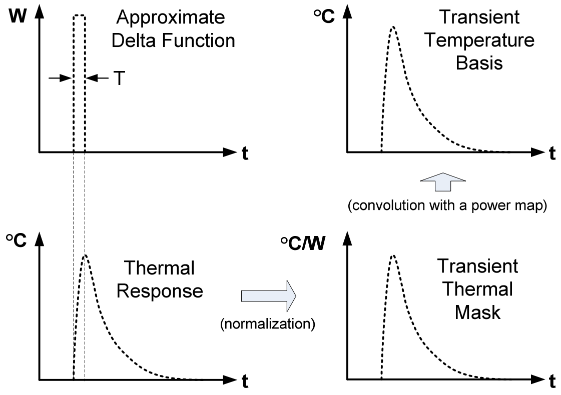


Figure 2.9: Obtaining the transient thermal mask and transient temperature basis (the visualization is given for a single spot on the die) [5].

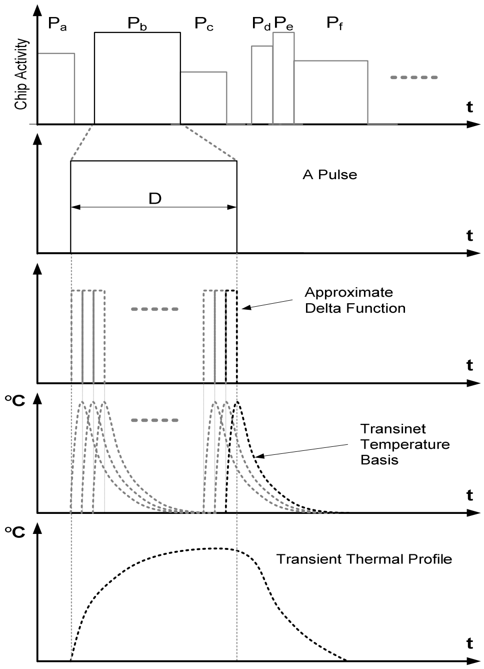


Figure 2.10: Schematic overview of the transient power blurring method (old algorithm); a rectangular pulse can be regarded as an aggregate of approximate delta functions. The overall transient thermal profile can be obtained by a superposition of shifted transient temperature bases [5].

2.4.2 New algorithm for Transient Methodology

The main limitation of the old algorithm, for transient power blurring, manifests itself when we are handling real workloads. In order to create only one thermal basis, the power map should be discretized with the same width as impulse heat and should then be convolved with the entire thermal mask length. When there are a large number of power cycles, as the case for a real workload, the number of thermal bases which have to be recorded are very large and occupy a huge portion of memory. Additionally, after these bases are created, they have to be read from the memory, shifted, and aggregated accordingly, so that the final temperature profile can be obtained. This will require a huge RAM to process.

In the new algorithm presented in this section, we offer three improvements:

1. There is no need to create thermal bases.
2. The discretized power pulses do not need to be convolved with the entire length of the transient thermal mask.
3. If the temperature response at a certain time step is needed, the operations need to be performed only at that time step.

In the new algorithm, first we obtain the transient thermal mask in the same manner as before. Then, we discretize the power pulses to be the same width as the width of the time steps of the transient thermal mask. If we assume the point heat source is divided to r time steps, the power map is discretized to n time steps, and thermal mask has m time steps, and then the flowchart shown in Figure 2.11 is employed to calculate the temperature profile at each time step. This algorithm efficiently minimizes the number of convolutions as well as aggregation operations needed to calculate the temperature profile at each time step. The flowchart is divided into three parts. In the first part, the temperature responses at the times smaller than the width of the point heat source impulse are calculated. As it can be seen, at those times we need only one convolution to calculate the temperature profile. At the times longer than the width of the point heat source impulse, but shorter than the full length of the transient thermal mask, the number of convolutions increase one term at each r time steps. At the times longer than the length of the transient thermal mask, the number of convolution is fixed.

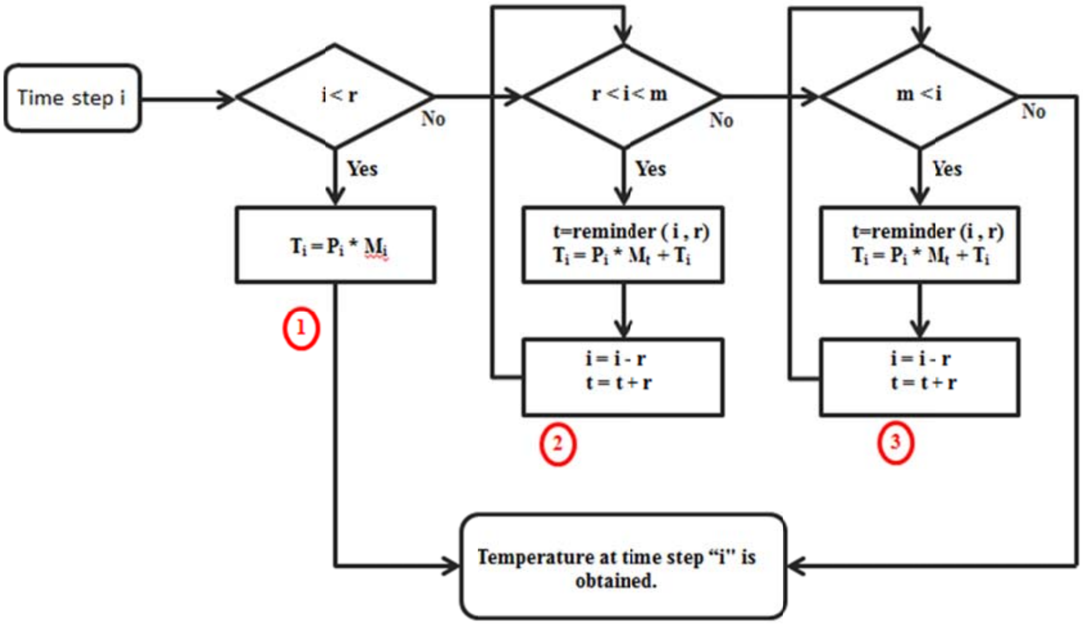


Figure 2.11: The new algorithm for transient power blurring; upon user request, the temperature profile at any time step, I, can be computed. r: No. of time steps in impulse heat source, m: No. of time steps in total length of thermal mask, T_i: Temperature at time step i, M: thermal mask, P_i: Power at time step i.

Let’s assume r=3, n=15, and m=10 time steps. This means we have 15 power cycles and need to calculate temperature at those cycles and afterwards. The calculation procedure shown in Figure 2.11 results in:

Part 1:

$$T_1 = P_1 * M_1; T_2 = P_2 * M_2; T_3 = P_3 * M_3; \tag{2.11}$$

Part 2:

$$T_4 = P_4 * M_1 + P_1 * M_4; T_5 = P_5 * M_2 + P_2 * M_5; T_6 = P_6 * M_3 + P_3 * M_6;$$

$$T_7 = P_7 * M_1 + P_4 * M_4 + P_1 * M_7; T_8 = P_8 * M_2 + P_5 * M_5 + P_2 * M_8;$$

$$T_9 = P_9 * M_3 + P_6 * M_6 + P_3 * M_9; \tag{2.12}$$

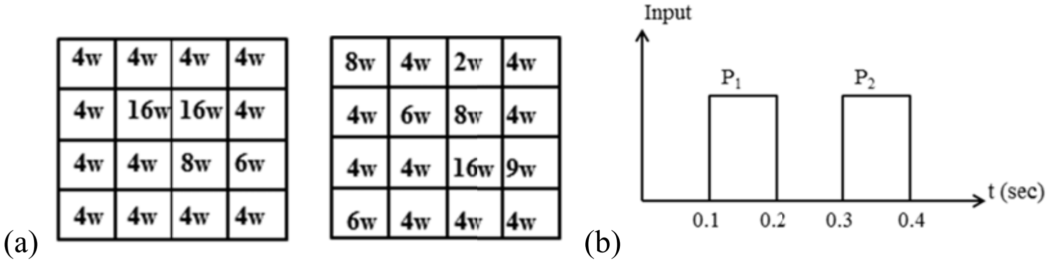
Part 3:

$$T_{10} = P_{10} * M_1 + P_7 * M_4 + P_4 * M_7 + \underline{\mathbf{(P_1 * (M_{10}=0))}} = P_{10} * M_1 + P_7 * M_4 + P_4 * M_7$$

$$T_{11} = P_{11} * M_2 + P_8 * M_5 + P_5 * M_8 \tag{2.13}$$

…

As it is seen, all the unnecessary convolutions and aggregations are eliminated. For example, in calculating T_is, terms such as P_1*M_2, P_1*M_5, etc. do not need to be computed, while the old algorithm computes and records all these unnecessary terms. Additionally, in the new algorithm, no thermal basis is recorded and memory is saved. If the number of power cycles is huge, the previous algorithm is not efficient and consumes a huge amount of time and memory. In the next section, the new and old methodologies are compared for two simple case studies.



4w	4w	4w	4w
4w	16w	16w	4w
4w	4w	8w	6w
4w	4w	4w	4w

8w	4w	2w	4w
4w	6w	8w	4w
4w	4w	16w	9w
6w	4w	4w	4w

Figure 2.12: Pulse input of the coarse power maps: (a) coarse power maps; (b) power dissipation pattern.

2.5 Transient Analysis Case Studies

In order to compare the old and new methodology for transient power blurring, the contrived power maps (40×40 grids) of an IC (1×1 cm^2) are used to demonstrate the capability of the PB method (see Figure 2.12a). Two different power maps (P_1 and P_2) in Figure 2.12a are applied at t=0.1 sec and t=0.3 sec for 0.1 sec duration, respectively. Resulting temperature profiles at t=0.15s and 0.35s are presented in Figures 2.13a & 2.13b, respectively. The relative errors for the PB method compare to ANSYS at these times are also shown in Figures 2.13c & 2.13d.

The maximum error (about 3%) occurs at the edges, in which the temperature is very close to ambient and a small change in temperature can cause a large relative error. Evolution of the maximum temperature (hot spot temperature) over time on top of the silicon is presented in Figure 2.14.

As it can be seen in Figure 2.15, the PB method renders transient thermal profiles with very high accuracy. The maximum error in the temperature profile over the entire chip in comparison with ANSYS, as well as in the hottest spot of the chip at each time step, are drawn in Figure 2.15. The relative error in the hottest spot on the chip is less than 0.7% throughout the case study. Additionally, the maximum error is about 30%, which occurs at the time when there is no power being dissipated in the chip and the obtained temperature of both methods is very close to ambient temperature. Therefore, a very small change leads to a large relative error value. For example, in this figure, the maximum error is at t=0.46s, in which there is no power being dissipated in the chip and the temperature difference between the results of the two method is 1.48 (Temperature obtained by ANSYS is 39.84˚C and for the PB method is 41.32˚C) while their values are very close to ambient temperature.

In terms of computational efficiency, ANSYS simulation took 7488s. The old and new algorithms for PB took 29.98s (reduction factor of 249) and 15.51s (reduction factor of 499), respectively, to perform the same simulation. As it can be seen for this simple case study, the new methodology for transient power blurring is twice as fast as the old methodology. The new methodology utilized one thirds of the memory needed for the old methodology, while it prevented the need for recording two 24MB of thermal bases. For a more complicated power map, these numbers increase. For example, the power maps shown in Figure 2.16 are applied to the same chip as in the previous case study. The evolution of the hot spot temperature for these power maps is shown in Figure 2.17. The old methodology took 61s to obtain the temperature profile, while for the new methodology the execution time remained the same (15.53s). Making the power map a bit more complicated compared to the first case study caused the old methodology to run almost twice as slow. The new algorithm utilized the same amount of memory to perform this case study while the old algorithm needed 1.7 times more memory. Also this time, the old methodology needed to record four 24MB thermal bases. For both case studies, the new methodology needed less than $\frac{4}{r} \times m$ convolutions (m is length of thermal mask and r is number of time steps in impulse input), while the old methodology needed 2×m and 4×m convolutions, respectively. The advantage of the computation time and memory usage reduction becomes more prominent when we conduct simulations with real workloads and finer grid sizes (Next chapter).

2.6 Summary

In this chapter, an overview of the power blurring method in both steady-state and transient regimes is given. A new algorithm to increase the speed of the power blurring in the transient regime, while keeping the memory usage low, is presented. For simple case studies presented in this chapter, the new algorithm indicated 4 times improvement in speed and about 6 times improvement in memory usage. In the next chapter, when we are handling real power traces, these ameliorations become more significant.

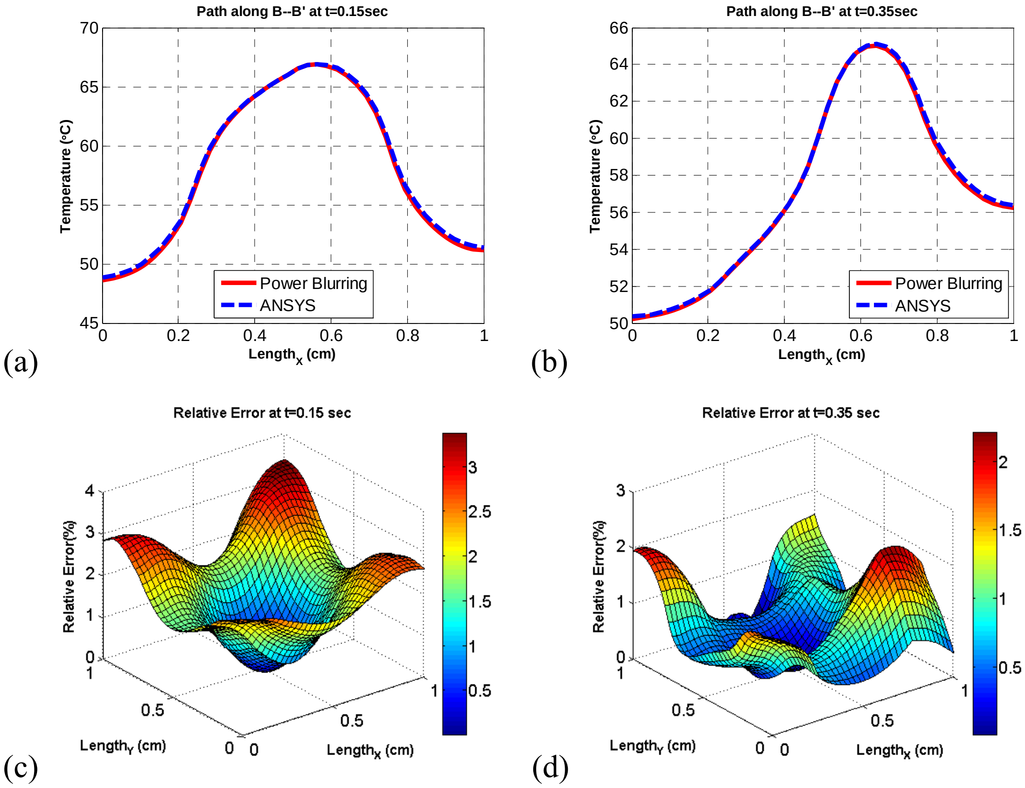


Figure 2.13. Thermal profile B-B’ at (a) t=0.15s and (b) t=0.35s (c) Relative error profile at t=0.15s (d) Relative error profile at t=0.35s.

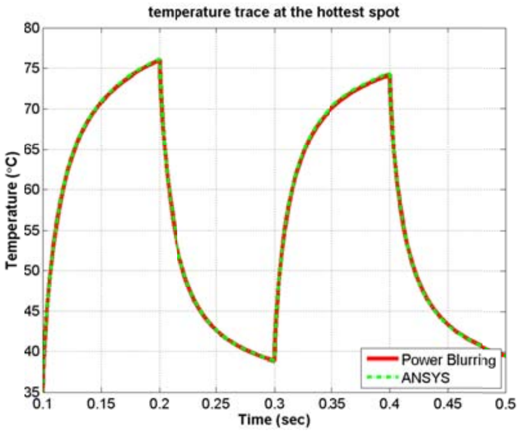


Figure 2.14: Evolution of transient hot spot over the chip.

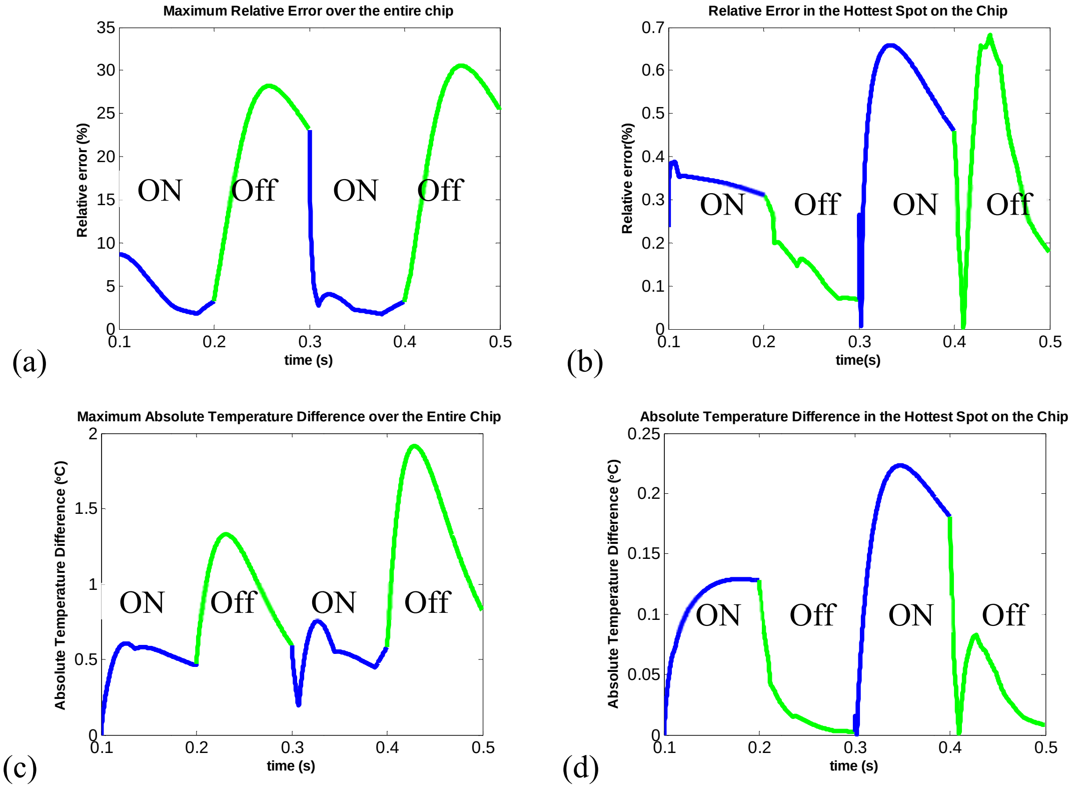
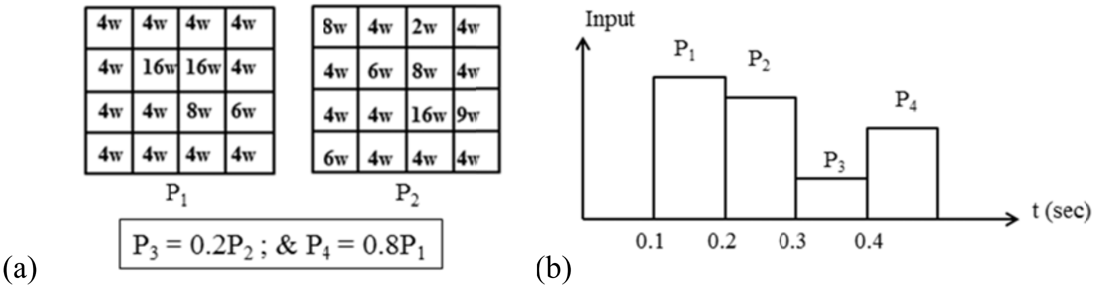


Figure 2.15: Evolution of relative error over time. The blue lines belong to the time when the device is ON and power is being dissipated (a) Maximum relative error (b) Relative error in the hottest spot on the chip. (c) Maximum absolute temperature error (d) Absolute temperature error in the hottest spot on the chip.



4w	4w	4w	4w
4w	16w	16w	4w
4w	4w	8w	6w
4w	4w	4w	4w

P_1

8w	4w	2w	4w
4w	6w	8w	4w
4w	4w	16w	9w
6w	4w	4w	4w

P_2

$P_3 = 0.2P_2$; & $P_4 = 0.8P_1$

Figure 2.16: Power maps for transient case study 2: (a) coarse power maps; (b) power dissipation pattern.

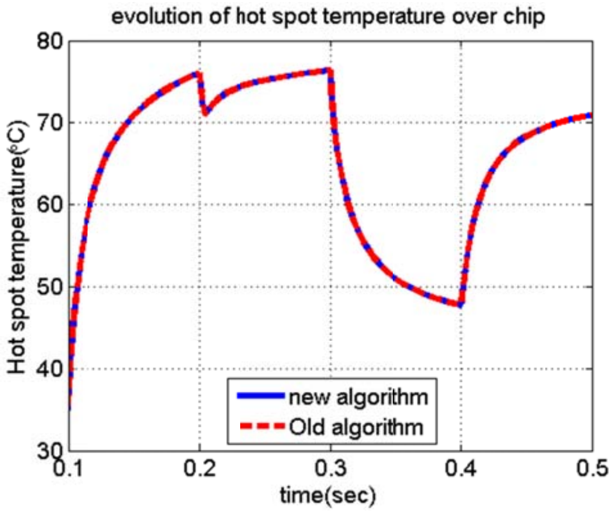


Figure 2.17: Evolution of transient temperature hot spot over the chip for power maps shown in Figure 2.16.

Chapter 3

Architecture Level Thermal Simulators

Architecture level simulators are designed to calculate temperature profiles which are accurate for the experiments at architecture level (block sizes in millimeter range), and still fast enough to allow for simulation of long dynamic temperature traces on the order of seconds. Their main feature, small computation time compared to detailed finite element models, comes at the cost of accuracy. However, this allows architects to study thermal and performance trade-offs in their system design. For that, a lot of details typically considered in a full thermal design, is deliberately neglected. In this chapter we introduce two commonly used fast architecture level thermal simulators, namely HotSpot [24] and SECSTherm[25]. A detailed comparison taking into account the accuracy and the computation speed, in both steady state and transient regimes, is performed between these methods and power blurring.

3.1 HotSpot

HotSpot is one of the thermal simulators widely used in computer architecture community. It is based on an equivalent circuit of thermal resistances and capacitances that correspond to the micro-architecture blocks. The essential aspects of the thermal package are also taken into account [24]. HotSpot solves the heat differential equations describing the RC circuit at each time step using a fourth-order Runge-Kutta method. The number of iterations for the Runge-Kutta solver is adaptive, to account for the different number of iterations required for convergence at different sampling intervals. HotSpot is already configurable for purposes of modeling new floorplans. HotSpot can model steady-state as well as transient cases. It can be run in two levels of accuracy: block model and grid model. While block level simulation has higher speed, the grid model is more accurate due to smaller spatial granularity of elements. We compare our method with the HotSpot in grid mode.

3.2 SESCTherm

SESCTherm is a thermal modeling infrastructure based on finite-difference analysis techniques. At the core of SESCTherm is a finite-difference model (FEM). Finite-difference analysis involves taking a problem and segmenting the problem into smaller pieces. SESCTherm divides the chip, package and associated components into a series of regular cross sectional regions. Each region is a quadrilateral, and no two cross sectional regions have abutting sides that are of different height or length. Each cross-sectional region is called a temperature node, and each region has a series of properties. To accurately characterize complex materials and dimensions, SESCTherm subdivides regions by material and geometry. This means that each temperature node is considered to be one material or an approximation of a combination of materials. Further, this means that any irregular shape is approximated by a combination of quadrilateral regions. Each node of a thermal system can be considered a heat source, heat sink, thermal capacitance, or thermal resistance. SESCTherm also updates the material properties as the temperature changes. It could support stacked layers of die and has models for interconnect layer, package and mainboard, as well as bulk silicon and silicon-on-insulator. To model temperature dependent material properties, SESCTherm updates material properties periodically based upon temperature variations. It supports configurable spatial granularity for different levels of accuracy. Unlike HotSpot, The meshing system is a consolidation of grid and block modes in which floorplan edges are extended and then each resulted region is meshed based on a given granularity [25]. However, the cores of the differential equation solver for both HotSpot and SESCTherm are based on the same algorithm.

3.3 PB vs. HotSpot vs. SESCTherm

To compare the three methods, we used two floorplans. One models the power dissipation profile of a typical out-of-order mobile processor, and the other one is a simple 4×4 grid. Additionally, the first floorplan is utilized to model the gcc workload from SPEC CPU 2000 benchmark suite. We configured each tool with the same parameters in terms of heat-sink and package characteristics.

3.3.1 Error Metrics

As it is described in section 2.2 error values are obtained using equations (2.8) and (2.9). In all the simulations the ambient temperature is set to 35ºC.

3.3.2 Methodology Validation

The convection coefficient between the heat sink and air is obtained using equation (3.1). In this equation R is the convection resistance between the heat sink and air, A is the surface area of the heat sink, and h is the equivalent convection coefficient.

$$h = 1/(R \times A) \qquad (3.1)$$

The convection resistance between heat sink and air is 0.1 K/W, and the surface area of the heat sink is 36 cm^2. This results in a convection coefficient of 0.2778 W/m^2-K. For ANSYS and Power Blurring we used this value of convection coefficient for our simulations. To make sure the overall chip and package models for all four methods match, and make the comparison fair, we perform one step of calibration before doing the evaluation. For that, instead of setting the parameters in HotSpot and SESCTherm to match the 0.2778 W/m^2-K, we try to adjust the convection coefficient to such a value that the overall average error is minimized. Then we evaluated the relative error values. For example in the steady-state case study, the initial average error obtained for HotSpot and SESCTherm were 19% and 26%, respectively. After applying this adjustment procedure, the average errors are minimized to 6% and 15%, which is shown in next section (Table 3.2). Then for the same optimized parameters we obtained the transient case study results.

In order to achieve the minimum average temperature error in HotSpot, we explicitly changed the value of convection resistance to 0.13 K/W. This is equivalent to 0.2137 W/m^2-K for convection coefficient which is a factor of 1.3 smaller than its real value (0.2778 W/m^2-K). In the case of SESCTherm, the error reduction procedure was different. SESCTherm does not allow the user to explicitly specify a value for convection resistance. Instead, it computes this value based on the given geometric and material properties, as well as the parameters of the cooling solution.

TABLE 3.1
MATERIAL PROPERTIES AND DIMENSIONS OF THE PACKAGE MODEL

	Area (mm^2)	Thickness (mm)	Thermal Conductivity (W/m-K)	Density (kg/m^3)	Specific Heat (J/kg-K)
Si Die	17×11.35	0.15	100	2330	751
TIM1	17×11.35	0.020	4	1930	2072.5
Heat Spreader	30×30	1	400	8933	397.4
TIM2	30×30	0.020	4	1930	2072.5
Heat Sink	60×60	6.9	400	8933	397.4

3.3.3 A Static Case Study

For steady-state thermal comparison, we used the power map from a typical mobile processor model. The dimensions of the chip and its cooling solution are set according to Figure 2.2 and Table 3.1. The power map, the floorplan of the device as well as the results obtained by three methods are shown in Figure 3.1 and 3.2, respectively. It should be mentioned that in order to obtain these results, a 64×64 meshing size was used, which is a default grid size in HotSpot [25]. This meshing also results in 266×177 μm^2 granularity which is fairly fine-grained for thermal evaluation of a processor at architecture level.

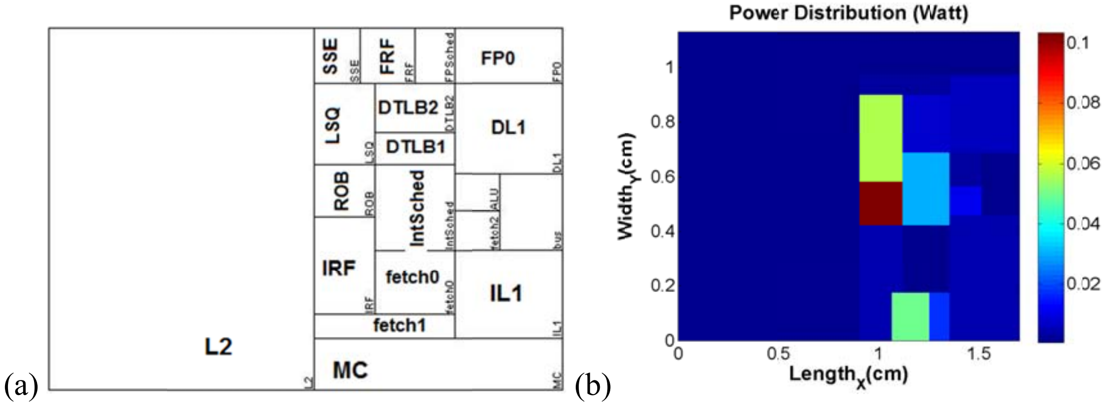


Figure 3.1: The floorplan and power map for steady-state case study. (a) The floorplan of the device. (b) A typical mobile processor model power map.

Table 3.2 summarizes the computation time, the absolute temperature error range, maximum error, the average error, and error in the hottest spot of the chip for each of the Power Blurring, HotSpot, and SESCTherm methods.

From Figure 3.2 and Table 3.2, it can be concluded that the PB method offers a more accurate result while its execution time is shorter than the SESCTherm and HotSpot. The latter delivers a more accurate result compare to SESCTherm, although it cannot provide the temperature profile with accuracy and resolution of the PB method. Since SESCTherm does not support a separate method for steady state analysis, we do not report the execution time for this case. In order to obtain the static result using SESCTherm, we have applied the power input and obtained the temperature profile after a long period of time to be able to consider it as a steady-state response. The computation time for ANSYS is 56s. In ANSYS we have used the sparse direct equation solver algorithm in which the time complexity is of the order of $O(n^2)$, for a thermal circuit with n nodes, while the time complexity of the FFT algorithm used in PB method is of the order of O(nlog(n)). SESCTherm and HotSpot employ traditional integration based solvers for which the lower bound of the computation time complexity is of the order of $O(n^c)$ where c is a number between 1.5 to 2 [26]. Considering these orders, one can see by increasing the number of nodes in thermal grid model, the computation time of the PB method increases with asymptotically slower rate than the other methods.

The error of PB compare to ANSYS in the hottest spot of the chip is only 0.14%, while it is 12.9% for HotSpot, and 13.1% for the SESCTherm simulator. The maximum error of 13% for the PB method relative to ANSYS in the entire profile is due to a temperature difference of 0.6ºC (39.1-38.5). Since this small change of temperature occurs at the very edge of the chip, in which the temperature is much lower than the center and very close to ambient temperature (35ºC), it will result in a large relative error value (see equation (2.8)) even though it is in fact a negligible change.

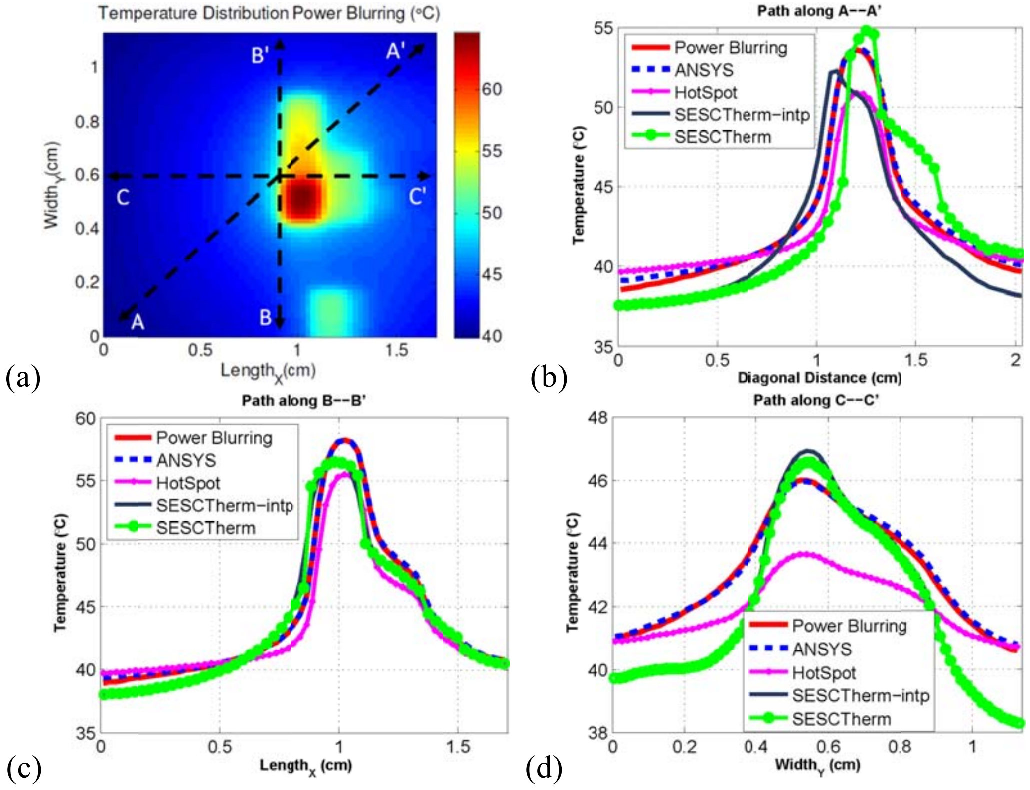


Figure 3.2: A static comparison between ANSYS, PB, HotSpot, and SESCTherm (both raw and interpolated temperature profiles). (a) The temperature profile obtained by the PB method corresponding to the power map shown in Figure 3.1b. (b) Temperature profiles over the diagonal of the chip. (c) Temperature profiles along the device length. (d) Temperature profiles along the device width.

TABLE 3.2
COMPARISON BETWEEN HOTSPOT, SESCTHERM AND POWER BLURRING.

	SESCTherm	HotSpot	PB
Computation Time	-	0.11s	0.041s
Err. in hot-spot	13.1%	12.9%	0.14%
Max. Err	43.7%	25.7%	13.7%
Avg. Err	15%	6.5%	2.5%
Abs. Err. range	0-5.3˚C	0-4.2 ˚C	0- 0.56˚C

The maximum errors in the HotSpot and SESCTherm simulators are 4 and 5 degrees temperature differences, respectively. The average error and absolute temperature error range in Table 3.2 indicate that the PB method accurately estimates the temperature distribution throughout the chip. All the aforementioned error values are obtained after adjustment of convection coefficient in HotSpot and SESCTherm. In Figures 3.3a and 3.3b a comparison between the cross sections of steady state results of these methods with the nominal value of convection coefficient, i.e. 0.2778 W/m^2-K, is presented. As mentioned, the procedure of meshing in SESCTherm is different from other methods. Based on a known granularity, structure of the blocks, and aspect ratio of the chip, SESCTherm automatically determines the meshing. In this case study the meshing size for the specified 230μm spatial granularity is 79×53. This results in asymmetric plots for the diagonal view of the temperature profile as it can be seen in Figure 3.2b.

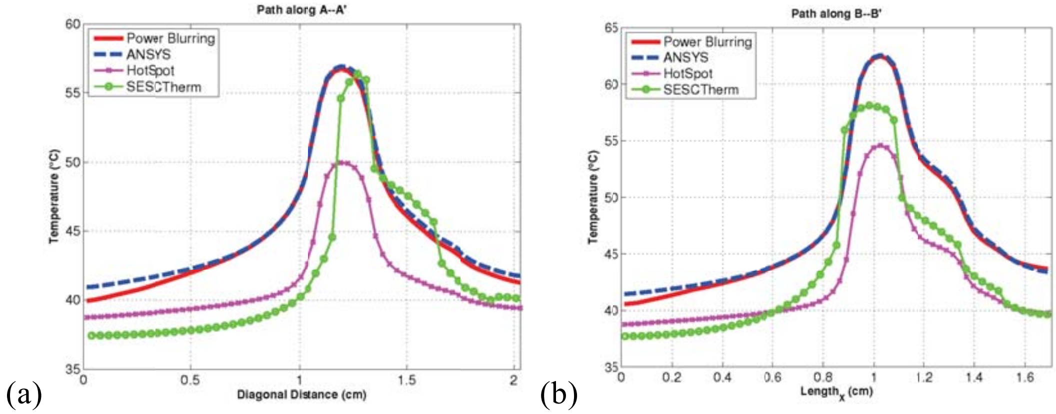


Figure 3.3: Steady state comparison between four methods using nominal value of convection coefficient (0.2778 W/m^2-K). (a) Temperature profiles over the diagonal of the chip. (b) Temperature profiles along the device length

To be able to calculate the relative error, the matrix dimensions of temperature profiles must be the same. Therefore, we interpolated the temperature profiles obtained by SESCTherm to 64×64 matrices. This, in turn, might add some inaccuracy to the relative error results obtained by SESCTherm, even though it does not change the average error. The cross sections of the interpolated result are also indicated in Figure 3.2.

3.3.4 Transient case study I

Transient simulations are performed for two case studies, a simple contrived power map as well as a real processor workload. The first case study is for the power train input shown in Figure 2.12. A 1×1 cm^2 chip with a cooling solution similar to the one shown in Figure 2.2, and the material properties shown in Table 2.2, is used. The chip has been meshed with 64×64 grid size. The resulting temperature profiles at t=0.15s and 0.35s are presented in Figure 3.4a, and 3.4b, respectively.

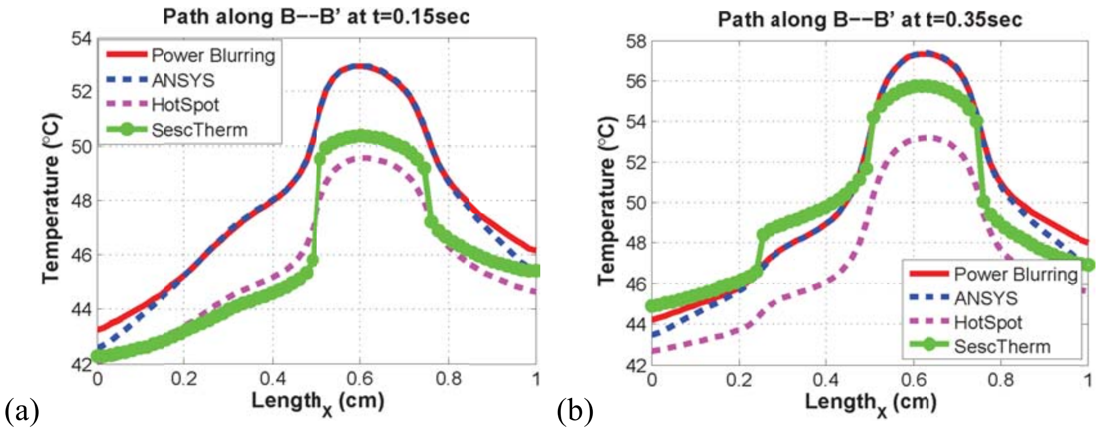


Figure 3.4: Transient case study comparison between the four methods. Thermal profile B-B at: (a) t=0.15s; and (b) t=0.35s.

In Table 3.3 a detailed comparison between the three methods is presented. The errors calculated in the table are for the time t=0.15s and 0.35s. The maximum error of 18% in PB corresponds to 1ºC absolute error. This occurs at the edge of the device where the temperature value is close to the ambient temperature and leads to a very small value in the denominator of the error function. A same argument is valid for large errors in HotSpot and SESCTherm and their absolute error values have to be considered which are provided in Table 3.3.

The execution time for the new transient PB algorithm was 73.1 seconds while this value was 194 for the old one. Again for a simple case study the new algorithm runs about 2.7 times faster than old algorithm. The simulation time for SESCTherm, HotSpot, and ANSYS is 290, 697, and 27858 seconds, respectively. Both new and old algorithms perform faster than other methods. For this case study the new PB algorithm is about 4, and 9.5 times faster than SESCTherm and HotSpot, respectively. PB also provides more accurate results. It should be mentioned that the PB method relies on two FEA simulations (or measurements) giving the unit impulse response at the center of the chip and the additional correction factor from uniform power dissipation profile. These calculations could be done offline, so the main advantage of PB is in multiple thermal simulations when different placements of the IC blocks are studied. All the matrix arithmetic calculations in that PB method have been done in MATLAB. While this is flexible and it allows the use of image processing tools, it is anticipated that direct implementation of the matrix convolution in a higher level program (e.g. C) can increase significantly the speed of the PB method.

TABLE 3.3
Comparison between HotSpot, SescTherm and Power Blurring (Transient)

	SescTherm	HotSpot	PB
Computation Time	290s	697s	73.1s
Err. in hot-spot @ 0.15s	3.9%	16.7%	0.24%
Max. Err @ 0.15s	33.2%	22.2%	16%
Avg. Err @ 0.15s	9.8%	11.7%	2.1%
Abs. Err. Range @ 0.15s	0-6.7˚C	0-4.7 ˚C	0- 1˚C
Err. in hot-spot @ 0.35s	7.5%	16%	0.13%
Max. Err @ 0.35s	39%	22.3%	18.6%
Avg. Err @ 0.35s	7.8%	15.3%	3%
Abs. Err. Range @ 0.35s	0-6.9˚C	0-4.5 ˚C	0- 1.34˚C

The evolution of error over time in the PB method is demonstrated in Figure 3.5. When the device is on and power being dissipated, the maximum absolute temperature error over the entire chip is less than 2ºC. The absolute error in the hottest spot over the entire chip with power blurring method is less than 0.2ºC.

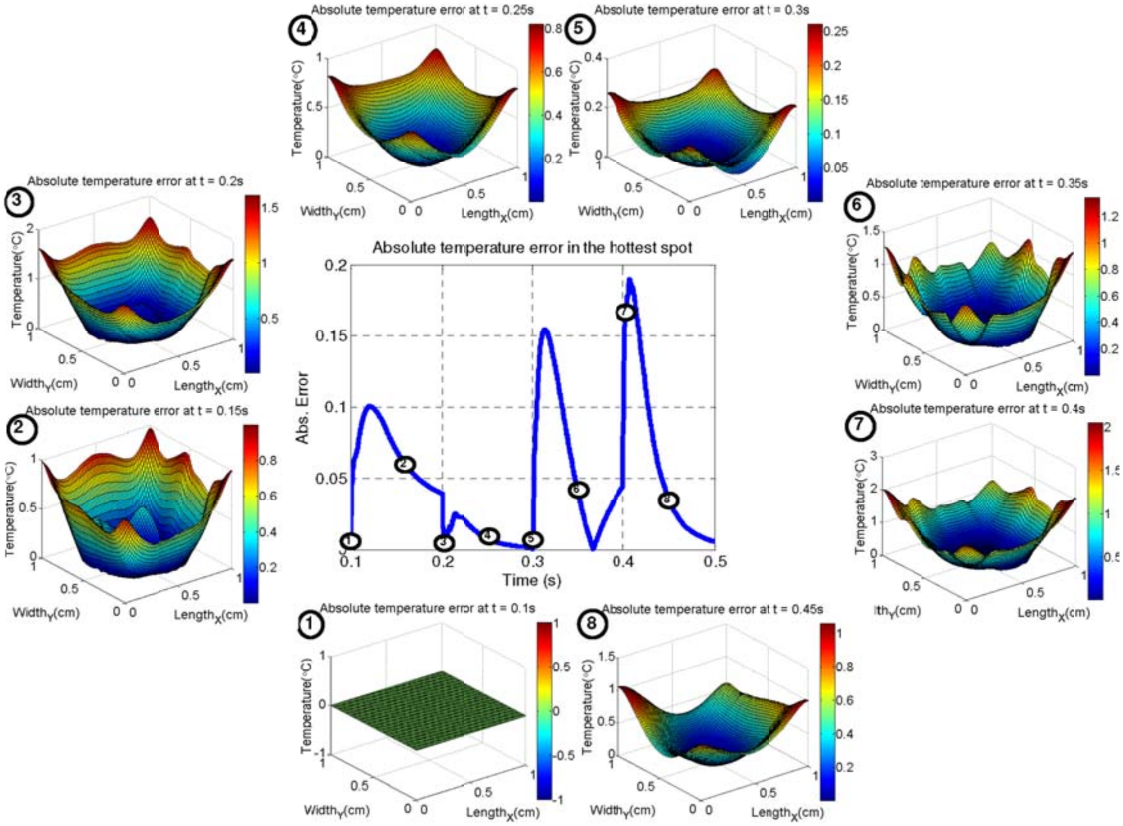


Figure 3.5: Center image shows the evolution of absolute temperature error in the hotspot over the entire chip. (1-8) Absolute temperature error profiles at different times over the chip.

3.3.5 Transient Case Study II (A real workload)

In the second case study, we evaluated transient response of the mobile processor executing gcc workload from SPEC CPU 2000 benchmark suite. We evaluate only one workload because the goal is to show the capability of running transient simulation with PB method. As shown in [27], not all the workloads show varied enough thermal transients, so we made sure to pick the selected workload, i.e. gcc, from the thermally interesting category introduced in that work.

We used SESC architectural performance simulator [28] to run the workload. To obtain the power trace, we modified SESC to send activity counters of each micro-architectural block to McPAT microarchitectural power model [29] every 3µs (around every 10K cycles at 3GHz as recommended in [23]). The total duration of power trace is 3.3s which is equivalent of 100K samples. The floorplan and dimensions of the chip are shown in Figure 3.1a and Table 3.1, respectively. The chip is meshed with grid size of 64×64.

To obtain transient thermal mask an impulse heat with the width of 100µs and the time step of 33µs is applied on the center element of the chip. The result is recorded for 60ms duration. To perform PB faster the power trace can be averaged over a number of cycles. In our code user can define this. A comparison between the results obtained without averaging power trace and with averaging every 400 cycles (every 13.33ms) is shown in Figure 3.6. As it can be seen in the figure, averaging (red curve) of power cycles is a good approximation and does not alter the final temperature results significantly.

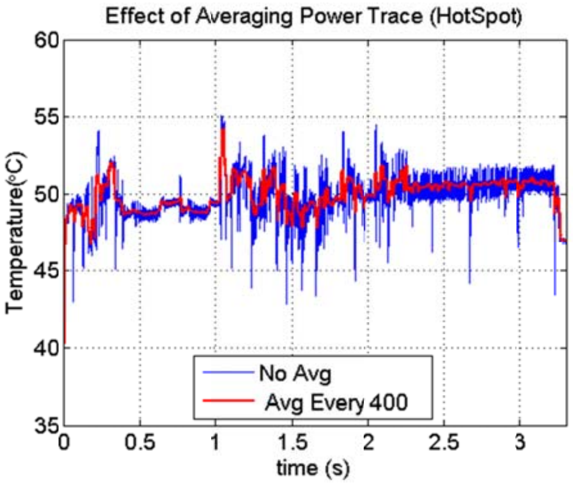


Figure 3.6: Effect of averaging power trace on evolution of hot spot temperature.

Evolution of hot spot as well as average temperature in results acquired by Power Blurring and HotSpot are compared in Figures 3.7a, and 3.7b, respectively. For power blurring results in these images, a 60ms length thermal mask is employed. In both methods we averaged the power trace every 400 Cycles. Evolution of maximum dissipated power as well as average dissipated power in the blocks over time is shown in Figures 3.8a, and 3.8b, respectively. We cannot report the error for each method because we are unable to run that long of transient simulation with ANSYS in order to obtain the reference temperature profile. However, this experiment emphasizes on the capability of PB method to integrate with an architectural performance simulator and to perform transient simulation at grid level for real time workloads. For this simulation, HotSpot took 193 minutes to obtain the results while this number was 80 minutes for PB.

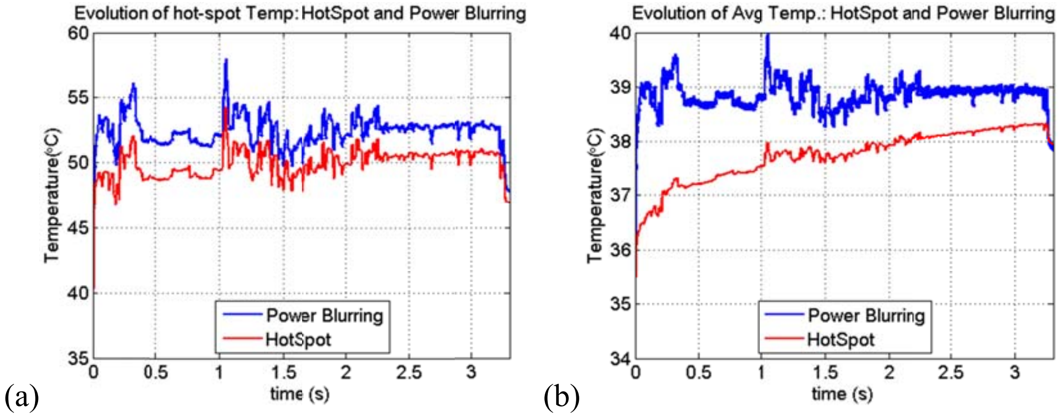


Figure 3.7: A comparison between HotSpot and Power Blurring. Evolution of (a) hot spot temperature, and (b) average temperature, in floorplan blocks.

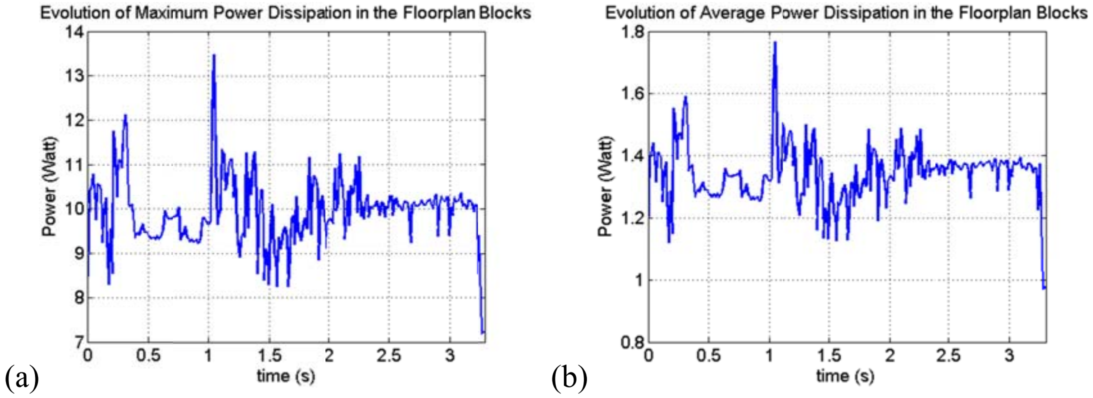


Figure 3.8: Evolution of (a) maximum dissipated power, and (b) average dissipated power, in floorplan blocks.

In order to increase PB speed the thermal mask can be truncated. We truncated the thermal mask at 30ms and 15ms. The execution time for the two cases of 30ms and 15ms long thermal masks was about 64 and 27 minutes, respectively. In Figure 3.9 the effect of truncation of thermal mask at 15ms, on the evolution of hot spot and average temperature, is illustrated. By sacrificing about 10% accuracy, compare to 60ms thermal mask results, PB obtained the transient temperature profile about 7 times faster than HotSpot. In Table 3.4 the execution time of the PB and HotSpot methods for this case study are presented. Finally, it should be mentioned that the old algorithm for power blurring needed about 5GB memory space (250×20MB) to record the thermal bases. Also, it needed a huge memory to read these bases and perform operation on them. Therefore, we did not perform this simulation with the old transient power blurring methodology.

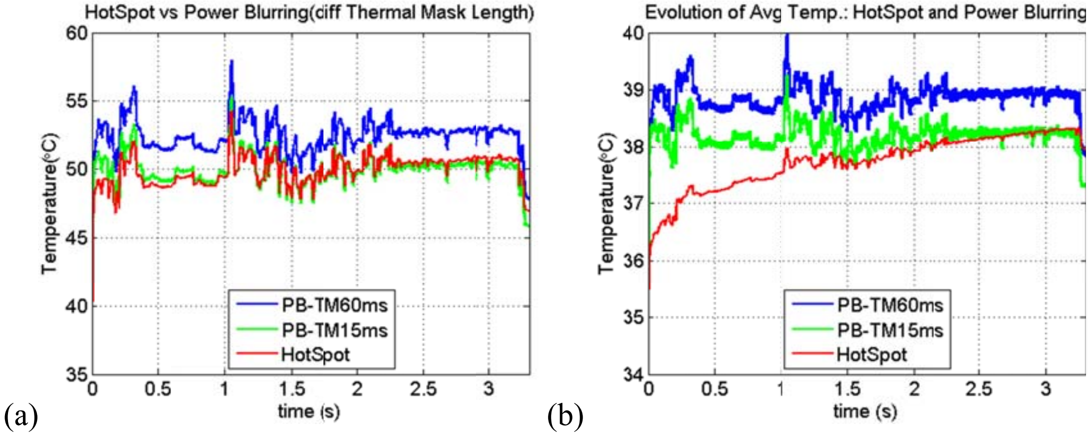


Figure 3.9: A comparison between HotSpot and Power Blurring with different thermal mask lengths. Evolution of (a) hot spot temperature, and (b) average temperature, in the blocks.

TABLE 3.4
EXECUTION TIME OF HOTSPOT AND PB FOR A REAL WORKLOAD

	PB with 60ms Thermal Mask	PB with 30ms Thermal Mask	PB with 15ms Thermal Mask	HotSpot
Execution Time	80min	64min	27min	193min

3.4 Summary

Fast and accurate thermal model enables in depth thermal evaluation in processor design. In this chapter, three different thermal models, namely Power Blurring (PB), HotSpot and SESCTherm are compared. HotSpot and SESCTherm are two standard architecture level simulators, and PB method has been recently proposed for thermal simulations. We adjusted the convection coefficient to such a value that the overall average errors in HotSpot and SESCTherm are minimized. This adjustment is a fitting parameter which does not have a scientific justification and could be different for different packages or even for different power profiles. PB does not need fitting the convection coefficient as it starts with the impulse response of the package with appropriate boundary conditions. Having validated the comparison methodology, both steady-state and transient case studies have indicated that the PB method can provide more accurate temperature profiles with shorter execution times. This is advantageous for in depth exploration of trade-offs in early stages of processor design process. Another application is in very high precision thermal simulation with micron scale power dissipation profile. The results shown in this chapter demonstrate that power blurring has the potential of becoming a promising thermal simulation tool at architecture and circuit levels.

Chapter 4

Adaptive Power Blurring

The thermal conductivity of silicon depends on temperature. Temperature rises of 40-50°C on the chip will reduce the thermal conductivity of the silicon by 10-20%. This could affect the hot spot temperature by 5-7°C. Power blurring is a linear method based on the superposition principle. In this chapter, we explain how we incorporated temperature dependent thermal conductivity of silicon in power blurring in a self-consistent manner.

4.1 PB and APB Methods

The principle of the Power Blurring (PB) method is described in chapter 2. Even though the Power Blurring method was successfully used to compute temperature profiles of IC's with good accuracy and max 1% error, it is limited to the case where temperature variation in devices is low. For high temperature changes in ICs, the method will have 7-12% errors. This is due to the fact that for large temperature variations, we can no longer neglect the temperature dependency of silicon thermal conductivity. Thermal conductivity of silicon versus temperature is shown in Figure 4.1 [30]. In order to overcome this limitation of PB, we propose two algorithms so that PB can adaptively (self-consistently) include the temperature dependency of silicon thermal conductivity in the calculation of the temperature profiles.

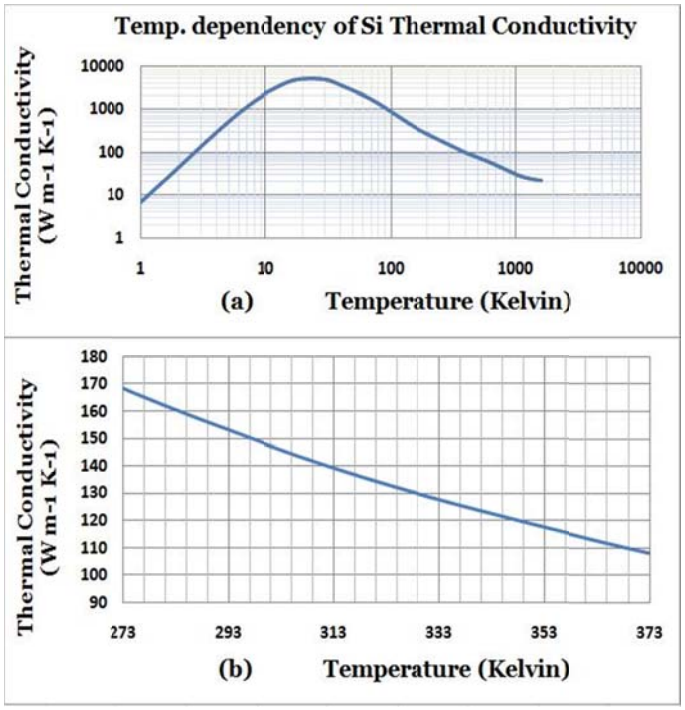


Figure 4.1: Silicon thermal conductivity vs. temperature for (a) 0-1500 K, and (b). 273-373 K (0-100˚C)

The algorithms for the two Adaptive Power Blurring methods, APB_I and APB_II, are shown in Figure 4.2 and 4.3, respectively. As it can be seen, in both cases we are using an iterative approach. In order to acquire an initial estimate for the temperature profile of the IC, both methods employ PB using the thermal mask obtained with room temperature thermal conductivity of silicon. The key difference between the two methods manifests itself in the way they analyze the initial temperature profile, and in turn, choose the corresponding thermal mask from a look-up table for the next iterations. In the first APB method (APB_I), the average increase in temperature field in the entire chip is obtained using a weighted averaging of power and preliminary temperature profiles as shown in equation (4.1).

$$T = \frac{\int\int P(x,y)T(x,y)\,dx\,dy}{\int\int P(x,y)\,dx\,dy} \quad (4.1)$$

Then, a new thermal mask, based on the average increase in temperature and the according change in Si thermal conductivity, is selected from the thermal mask's look-up table. The new thermal mask is convolved with the power map to provide a new estimate of the temperature profile. This scheme is then applied iteratively until the temperature profile converges to the final result.

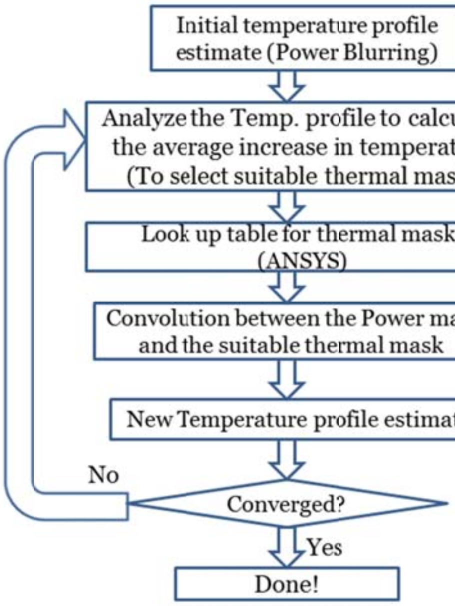

Initial temperature profile estimate (Power Blurring)
Analyze the Temp. profile to calculate the average increase in temperature (To select suitable thermal mask)
Look up table for thermal masks (ANSYS)
Convolution between the Power map and the suitable thermal mask
New Temperature profile estimate
No
Converged?
Yes
Done!

Figure 4.2: Algorithm for the first Adaptive Power Blurring method (APB_I).

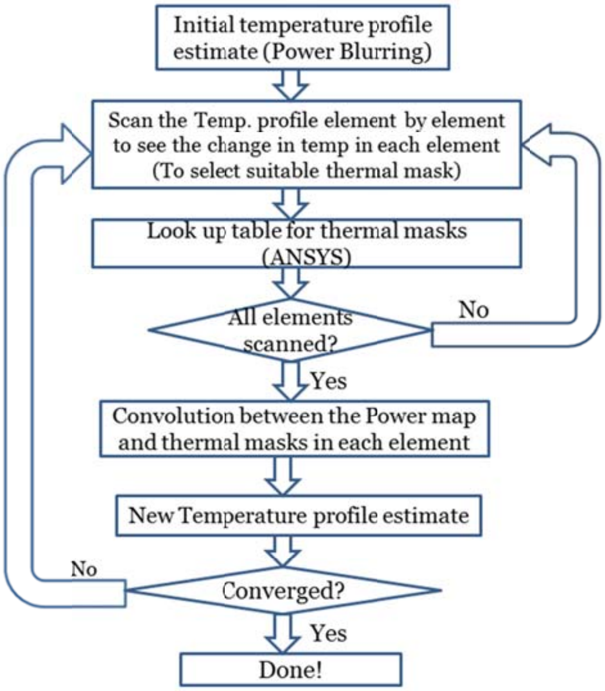

Initial temperature profile estimate (Power Blurring)
Scan the Temp. profile element by element to see the change in temp in each element (To select suitable thermal mask)
Look up table for thermal masks (ANSYS)
All elements scanned?
No
Yes
Convolution between the Power map and thermal masks in each element
New Temperature profile estimate
No
Converged?
Yes
Done!

Figure 4.3: Algorithm for the second Adaptive Power Blurring method (APB_II).

On the other hand, in the second APB method (APB_II), the initial temperature profile is scanned element by element and the local change in the temperature of each element is calculated. Then, based on the increase in temperature in each element and corresponding silicon thermal conductivity, an appropriate thermal mask is chosen. Therefore, for each element on the temperature profile, a new thermal mask is acquired. These thermal masks (impulse responses) are convolved element by element with the power map and yield a new temperature profile estimate. This scheme is again exploited iteratively until the temperature profile converges to the final result. For every iterative method, an appropriate initial guess is needed so that the final result could converge correctly. In our iterative methods, we are using the Power Blurring method to give us the initial guess. This by itself already provides a good estimate of the temperature profile, ensuring fast and accurate convergence of the two Adaptive Power Blurring techniques.

4.2 Comparison Results and Discussions

4.2.1 The Thermal Masks Look-up Table

For these simulations, the package shown in Figure 2.1 is utilized. The material properties are set according to Table II. The Si IC was orthogonally meshed with an element size of 0.025 × 0.025 cm^2 (40×40 meshed elements). In order to obtain thermal masks, a point heat source with a heat flux of 6250 W/cm^2 is applied to the center element on the Si chip. The ambient temperature is set to 27˚C. By changing the thermal conductivity value based on the data shown in Figure 4.1, we obtained different thermal masks for the look-up table. In [31], the thermal mask is parameterized for thermal conductivity of silicon. This can be used to produce the thermal masks look-up table.

4.2.2 Case studies

In order to assess the improvement offered by the iterative schemes, we performed the PB and the two APB methods on the power maps shown in Figure 4.4. The power distribution shown in Figure 4.4a ("Edge" power map) was aimed at providing a worst case scenario by concentrating all the power on the edges. Figures 4.4b, 4.4c, and 4.4d are realistic representations of what a power distribution on a modern-day ASIC might look like. They aimed at revealing the accuracy of our methods in estimating hot spots.

The results obtained with each of these methods for either of the power maps are compared with the results obtained by ANSYS for the same power map. It should be mentioned that ANSYS also considers the changes in thermal conductivity based on the data shown in Figure 4.1.

The temperature profile acquired by ANSYS for the power map shown in Figure 4.4d is depicted in Figure 4.5. As it can be seen in this figure, the temperature rise relative to ambient temperature (27˚C) ranges between 20 to 75 degrees. The temperature fields obtained for other power maps have also large variations relative to ambient temperature.

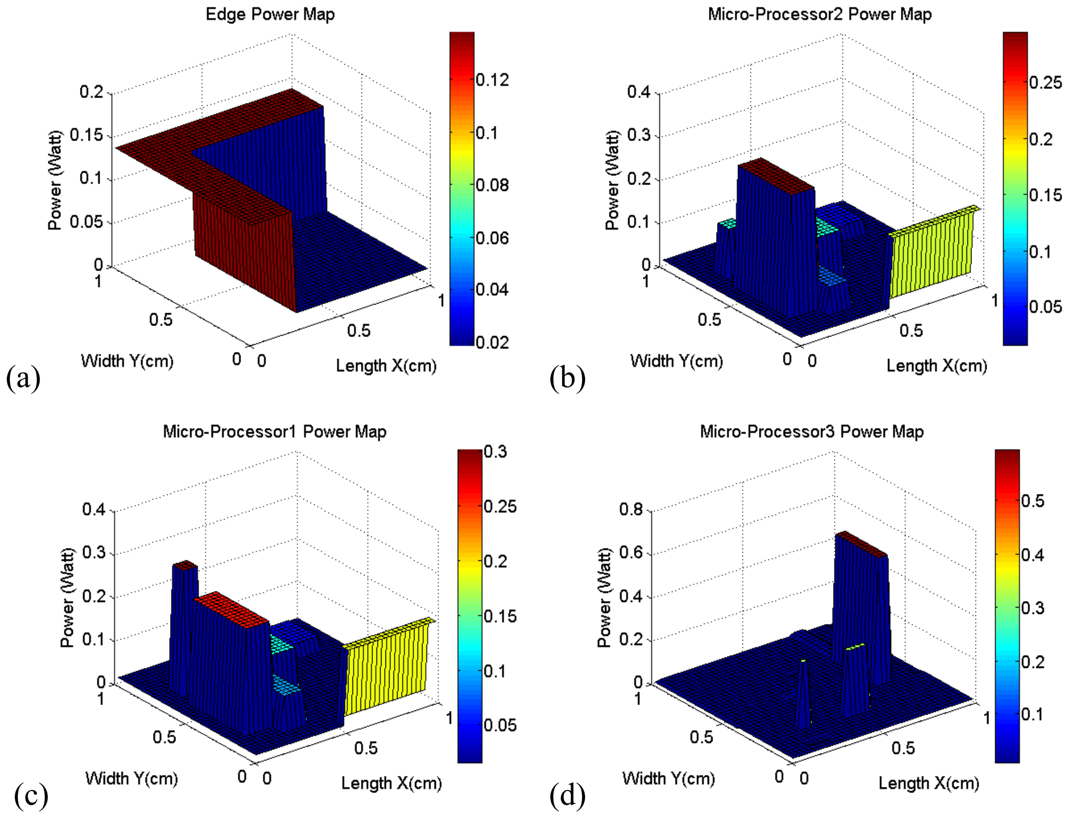


Figure 4.4: 3D view of the four sample power dissipation maps selected for our study. (a) “Edge” Power map (b) “µprocessor 1” power map (c)“µprocessor 2” power map, and (d) “µprocessor 3” power map.

The average, maximum and hottest spot error are calculated by comparison between the PB, APB_I and APB_II methods, relative to simulations done by ANSYS. These relative errors are obtained for all four power maps and are tabulated in Table (4.1). As it can be seen in the table, both the APB_I and APB_II are decreasing the maximum, average error and that in hot spots. The APB_II in general provides more accurate results in comparison with the APB_I, because it scans the entire temperature profile and selects the appropriate thermal mask at each location. Therefore, temperatures at hot spots are calculated using the APB_II with an order of 2-20 times better (less error) than the APB_I method and 8-100 times better than the PB method.

As an example for the power map shown in Figure 4.4c, the error in computing hot spot temperature relative to ANSYS using APB_II is 0.25%, while this error is 9 times (2.21%) and 25 times (6.4%) larger for the APB_I and PB methods, respectively. On the other hand, the speed of calculation and convergence is the major advantage of the APB_I over the APB_II method. This is due to the fact that the second iterative method (APB_II) needs to scan the entire temperature profile and perform the convolution for each element. Considering the results obtained by the APB_I method, one observes that while the method has an error rate close to what is obtained from the APB_II method, it provides calculation speed 5-8 times better than the APB_II. For instance, the execution time for the PB, APB_I, APB_II methods and the ANSYS simulation performed on the power map shown in Figure 4.4c were 0.04s, 0.15s, 0.67s, and 19.8s, respectively. In each case for both iterative methods, it took just 3-4 iterations to converge. This fast convergence is due to the fact that we used the PB method to obtain the initial guess. In order to stop iterations we selected the point in which the mean square difference between the last two results produced in each of the APB method is less than 10^{-9}.

TABLE (4.1)
COMPARISON BETWEEN PB, APB_I AND APB_II

Power Map		PB	APB_I	APB_2
-	Computation Time	0.04s	0.15s	0.67s
Edge	Max. Err	4.54%	3.47%	3.6%
	Avg. Err	2.36%	0.97%	0.92%
	Err. in hot-spot	4.53%	0.77%	0.04%
µprocessor 1	Max. Err	7.32%	3.05%	2.16%
	Avg. Err	1.54%	0.78%	0.62%
	Err. in hot-spot	7.04%	2.46%	0.96%
µprocessor 2	Max. Err	6.69%	2.69%	2.09%
	Avg. Err	1.55%	0.74%	0.59%
	Err. in hot-spot	6.4%	2.21%	0.25%
µprocessor 3	Max. Err	7.9%	1.84%	1.78%
	Avg. Err	1.14%	0.56%	0.61%
	Err. in hot-spot	7.9%	1.83%	1.02%

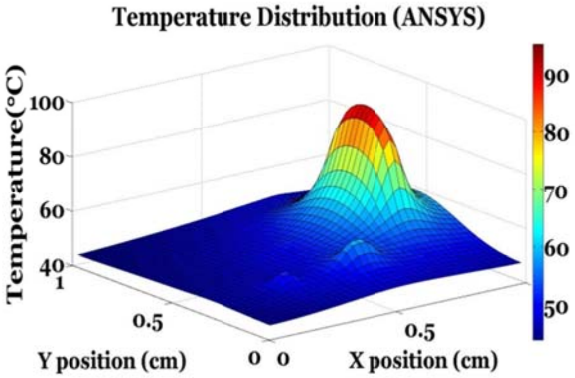


Figure 4.5: Temperature profile obtained by ANSYS for the power map shown in Figure 4.4d.

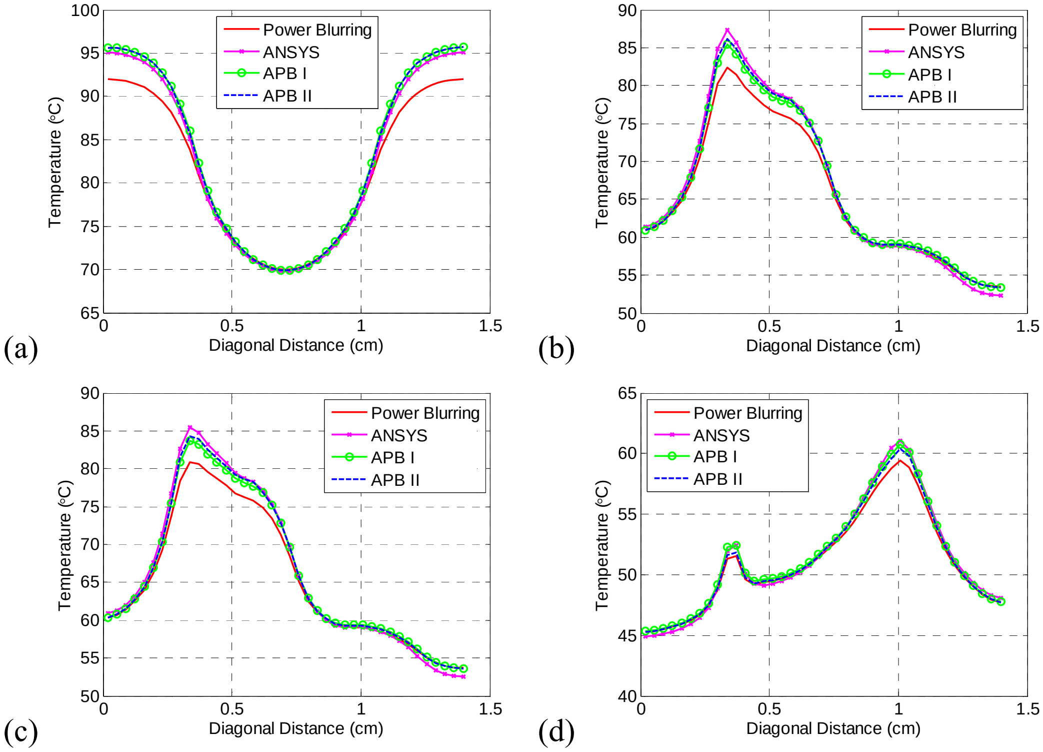


Figure 4.6: Diagonal cross section of the temperature profiles for different power distributions and algorithms. (a)"edge" power map, (b)"µprocessor 1" power map, (c)"µprocessor 2" power map, (d)"µprocessor 3" power map.

Figures 4.6a, 4.6b, 4.6c, and 4.6d feature the cross section of the temperature profiles calculated with the PB method, the two APB methods and simulated by ANSYS. As it can be seen, both iterative procedures provide results in good agreement with what has been calculated by ANSYS.

4.3 Validation of ANSYS Nonlinear Simulations

In order to confirm the validity of nonlinear simulations carried out in ANSYS, considering the temperature dependency of thermal conductivity, two criteria are required to be met. First, the sum of the total heat fluxes at all the surfaces of the model need to be zero for a converged solution. Second, a mesh independence test has to be conducted on the model by running the same simulation with a finer mesh.

The ANSYS results, obtained for these case studies, are studied. The total heat flux of surfaces equals zero. Also, the temperature profile with a finer mesh (80×80 meshed elements) is calculated. The maximum difference between the finer meshed model results and current model (40×40 meshed elements) was 0.385ºC. This means that refining the meshing did not have a significant effect on the results. Executing these two procedures confirm the validity of nonlinear ANSYS simulation performed in our case studies.

4.4 Summary

Adaptive Power Blurring methods are presented as an extension to the Power Blurring method to account for the temperature dependence of silicon's thermal conductivity. It has been shown that for large temperature variations in ICs, APB methods can estimate the temperature with much less error than the PB method. In addition, even though these are iterative methods, they converge quickly and accurately (in just 3 or 4 iterations). This can be attributed to the fact that the PB method, giving a good estimate of temperature distribution, is used to get the initial guess for our iterative procedure. A comparison between two APB methods is conducted. The APB_II method, which selects an appropriate thermal mask point by point based on the local temperature, provides a better assessment of the temperature, particularly at hot spots. Though it runs slower due to the multiple numbers of convolutions, it is still considerably faster than conventional FEM calculations. ANSYS nonlinear simulations are also validated, which in turn confirms the accuracy of the results obtained by the adaptive power blurring methods.

Chapter 5

Conclusion

5.1 Contributions

In this work, power blurring's steady-state and transient methodologies are explained. A new algorithm for power blurring transient thermal analysis is proposed. This new algorithm increases the efficiency of the method in terms of speed and memory usage.

One of the main applications of fast transient simulations is in architectural simulations. This is because long temperature traces need to be computed for the execution of applications on the processor under study. Power blurring is compared with two standard fast architecture level thermal simulators, namely HotSpot and SESCTherm. The comparison is carried out in both static and transient regimes. Multiple case studies are investigated. Power blurring performed much better in term of accuracy and speed compare to these two methods with grid level resolution. It is concluded that power blurring has the potential of becoming a promising thermal simulation tool for architecture and circuit simulations.

Finally, adaptive power blurring methods are developed to incorporate the temperature dependent thermal conductivity of silicon in a self-consistent manner. It is indicated that these methods accurately estimate the temperature profile of VLSI ICs, while they maintain their high speed compared to finite element methods.

5.2 Future work

The first future plan for this work is to fully implement the code in a lower level programming language (e.g. C language) and integrate the simulator into power/ performance simulator, such as SESC. Fast transient thermal simulations enable the architecture designers to develop a reliable and efficient thermal-aware floorplaning method [32].

In [33], power blurring is employed in an inverse procedure to extract the power dissipation map from temperature profile. Thermoreflectance thermal imaging has been widely used to experimentally capture the temperature profile of ICs [34, 35, 36]. Another future plan for this work is to integrate the inverse method described in

[33] with experimental thermal imaging cameras in order to extract a power dissipation profile in ICs.

Power blurring has been employed to calculate the temperature profile of 3D ICs considering thermal vias [37]. It is shown that when the heat coming from the bottom of the thermal via (bottom die) reaches the top of the thermal via in the top die (closer to the heat sink), it causes elevation in the temperature in those region. Integrating solid-state coolers into the chip on top of the thermal via (between the die and heat spreader) is suggested. This way, heat can easily flow to the heat sink. Thermal simulation of these models is very time- consuming and inefficient with FEM. Employing a fast and accurate tool is very advantageous at the early stages of design for the routing and placing of via regions and determining the density of via material (copper) in those regions. Extending power blurring so that it can be used for thermal simulation of these models is another future direction of this work.

Appendix A

List of Publications

[1] A. Ziabari, Z. Bian, and A. Shakouri. Adaptive Power Blurring Techniques to Calculate Temperature Profiles Under Large Temperature Variations. International Microelectronic and Packaging Society (IMAPS) ATW on Thermal Management, Best Paper Award, Sep. 28-3-, 2010.

[2] A. Ziabari, J.H. Park, E. K. Ardestani, J. Renau, S.M. Kang, and A. Shakouri. Power Blurring: Fast Chip-Level Static and Transient Thermal Analysis Method for Packaged Integrated Circuits. Submitted, IEEE Transaction on CAD of Integrated Circuits and Systems.

[3] A. Ziabari, E. K. Ardestani, J. Renau, and A. Shakouri. Fast Thermal Simulators for Architectural Level Circuit Design. Proc. of 27th Annual Thermal Measurement, Modeling, and Management Symposium (SemiTherm), San Jose, CA, March 2011.

[4] A. Ziabari, and A. Shakouri. Fast Thermal Simulation of Vertically Integrated Circuits (3D ICs) Including Thermal Vias. 13th IEEE Intersociety Conference on Theraml and Thermomechanical phenomenon in Electronic Systems (ITherm), May 30, 2012

[5] K. Maize, A. Ziabari, X. Wang, W. French, Ph. Lindorfer, B. OCannel, and A. Shakouri. Thermoreflectance Imaging of Self-Heating in Power Transistor Arrays. Submitted, IEEE Transaction on Power Transistor Arrays, Feb 27, 2011.

[6] E. K. Ardestani, A. Ziabari, A. Shakouri, and J. Renau. Enabling Power Density and Thermal-Aware Floorplanning. Proc. of 28th Annual Thermal Measurement, Modeling, and Management Symposium (SemiTherm), San Jose, CA, March 2012.

Bibliography

[1] A. H. Ajami, K. Banerjee, and M. Pedram. Modeling and Analysis of Non-uniform Substrate Temperature Effect on Global ULSI Interconnects. IEEE Trans. Computer Aided Design Integrated Circuits Syst., vol. 24, no. 6, pp. 849-861, 2005.

[2] Y. K. Cheng, and S. M. Kang. A Temperature-Aware Simulation Environment for Reliable ULSI Chip Design. IEEE Trans. Computer Aided Design Integrated Circuits Syst., vol. 19, no. 10, pp. 1211-1220, 2000.

[3] V.M. Heriz, J.H. Park, T. Kemper, S.M. Kang, A. Shakouri. Method of Images for the Fast Calculation of Temperature Distributions in Packaged VLSI Chips. International Workshop on Thermal Investigation of ICs and Systems (Therminic), pp.18-25, 2007.

[4] J.H. Park, X. Wang, A. Shakouri, S.M. Kang. Fast Computation of Temperature Profile of VLSI ICs with High Spatial Resolution. Semiconductor Thermal Measurement, Modeling, and Management Symposium (SemiTherm 24), pp. 50-55, 2008.

[5] J.H. Park, A. Shakouri, and S.M. Kang. Fast Evaluation Method for Transient Hot Spots in VLSI ICs in Packages. 9th International Symposium on Quality Electronic Design (ISQED 08), San Jose, pp. 600-603, 2008.

[6] ANSYS R 11.0, Swanson ANSYS Inc. 2007.

[7] A. J. Chapman. Heat Transfer. MacMillan Publishing Company, 4th Edition, 1984.

[8] B. Wang, and P. Mazumder. “Accelerated Chip-Level Thermal Analysis Using Multilayer Green’s Function. IEEE Trans. Computer Aided Design Integrated Circuits Syst., vol. 26, no. 2, 2007, pp. 325-344.

[9] Y. Zhan, and S. Sapatnekar. A High Efficiency Full-Chip Thermal Simulation Algorithm. Proceedings of the 2005 IEEE/ACM International Conference on Computer-aided Design, San Jose, CA, pp. 635-638, 2005.

[10] P. E. Bagnoli, C. Bartoli, and F. Stefani. "Validation of DJOSER Analytical Thermal Simulator for Electronic Power Devices and Assembling Structures. Workshop on Thermla Investigation of ICs and Systems (Therminic), vol. 38, no. 2, pp. 185-196, 2007.

[11] T. Y. Wang, and C. C. P. Chen. 3D Thermal ADI: A Linear-Time Chip Level Transient Thermal Simulator. IEEE Trans. Computer-aided Design Integrated Circuits Syst., vol. 21, no. 12, 2002, pp. 1434-1445.

[12] L. Codecasa, D. D'Amore, and P. Maffezzoni. An Arnoldi Based Thermal Network Reduction Method for Electro-Thermal Analysis. IEEE Trans. Components and Packaging Technologies, vol. 26, no. 1, 2003, pp. 186-192.

[13] S. C. Lin and K. Banerjee. An Electrothermally-Aware Full-Chip Substrate Temperature Gradient Evaluation Methodology for Leakage Dominant Technologies with Implication for Power Estimation and Hot-Spot Management. Int. Conf. Comput.-Aided Design, San Jose, CA, 2006.

[14] B. Allard, X. Jorda, P. Bidan, A. Rumeau, H. Morel, X. Perpina,M. Vellvehi, and S. M'rad. Reduced-Order Thermal Behavioral Model based on Diffusive Representation. IEEE TRANSACTIONS ON POWER ELECTRONICS, vol. 24, no. 12, pp. 2833-2846, 2009.

[15] R. C. Gonzalez, and R. E. Woods. Digital Image Processing. 2nd Edition, Prentice Hall, New Jersey, 2001.

[16] C. A. Balanis. Advanced Engineering Electromagnetics. John Wiley & Sons, New York, 1989, chap. 14.

[17] F. N. Najm. A Survey of Power Estimation Techniques in VLSI Circuits. IEEE Trans. Very Large Scale Integ. (VLSI) Systems, vol. 2, no. 4, 1994, pp. 446-455.

[18] S. C. Lin, G. Chrysler, R. Mahajan, V. K. De, K. Banerjee, "A Self-Consistent Substrate Thermal Profile Estimation Technique for Nanoscale Ics-Part I:

Electrothermal Couplings and Full-Chip Package Thermal Model," *IEEE Trans. Electron Devices*, vol. 54, no 12, 2007, pp. 3342-3350.

[19] C. H. Díaz, S. M. Kang, and C. Duvvury. Circuit-Level Electrothermal Simulation of Electrical Overstress Failures in Advanced MOS I/O Protection Devices. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 13, no. 4, 1994, pp. 482-493.

[20] T. Kemper, Y. Zhang, Z. Bian, and A. Shakouri, "Ultrafast Temperature Profile Calculation in IC chips," Proc. of 12th International Workshop on Thermal investigations of ICs (THERMINIC), Nice, France, 2006, pp. 133-137.

[21] I. V. Lindell. Image Theory for Electromagnetic Sources in Chiral Medium Above the Soft and Hard Boundary. IEEE Transactions on Antennas and Propagation, vol. 49, no. 7, July 2001, pp. 1065-1068.

[22] MATLAB R2006b, The MathWorks Inc., 2006.

[23] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Computer Systems: Opportunities and Challenges, IEEE Micro, vol. 23, no. 6, pp. 52-61, 2003.

[24] J. Nayfach, and J. Renau. SOI, Interconnect, Package, and Mainboard Thermal Characterization, International Symposium on Low Electronics and Design (ISLPED), pp. 327-330, Aug. 2009.

[25] http://lava.cs.virginia.edu/HotSpot/index.htm

[26] P. Liu, Z. Qi, H. Li, L. Jin, W. Wu, S. X. D. Tan, and J. Yang. Fast Thermal Simulation for Architecture Level Dynamic Thermal Management. IEEE/ACM International Conference on Computer-Aided Design, ppp. 639-644, 2005.

[27] F. J. Mesa-Martinez, E. K. Ardestani, and J. Renau. Characterizing processor thermal behavior. Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems 2010, pp 193–204.

[28] J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, P. Montesinos. SESC simulator, http://sesc.sourceforge.net,2005

[29] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 42, 2009, pp 469–480.

[30] http://www.efunda.com/materials/elements/TC_Table.cfm?Element_ID=Si

[31] J.H. Park, V.M. Heriz, A. Shakouri, and , S.M. Kang. Ultrafast Calculation of Temperature Profiles of VLSI ICs in Thermal Packages Considering Parameter Variations. IMAPS 40th Int. Symp. Microelectronic, San Jose, Ca, Nov. 2007.

[32] E. K. Ardestani, A. Ziabari, A. Shakouri, and J. Renau. Enabling Power Density and Thermal-Aware Floorplaning. 28th Annual Thermal Measurement, Modeling and Management Symposium (SEMITHERM), San Jose, March 2012.

[33] X. Wang, S. Farsiu, P. Milanfar, and A. Shakouri. Power Trace: An efficient Method for Extracting the Power Dissipation Profile in IC Chip From Its Temperature Map. IEEE Transaction on Components and Packaging Technologies, vol. 32, No. 2, pp. 209-316, June 2009.

[34] J. Christofferson, and A. Shakouri. Thermoreflectance Base Thermal Microscope. Review of Scientific Instruments, vol. 76, 024903-1-6, Jan. 2005.

[35] K. Maize, X. Wang, J.H. Park, J. Christofferson, S. Kang, and A. Shakouri. High Speed Transient Thermal Characterization and Simulation of Integrated Circuits. Japanese Journal of Applied Pgysics, June 2008.

[36] B. Vermeersch, J. Christofferson, K. Maize, A. Shakouri, and G. De Mey. Time and Frequency Domain CCD-Based Thermoreflectance Techniques for High-Resolution Transient Thermal Imaging. Proceedings of IEEE 26th SEMITHERM, Santa Clara CA, ppp. 228-234, Feb 2010.

[37] A. Ziabari, and A. Shakouri. Fast Thermal Simulation of Vertically Integrated Circuits (3D ICs) Including Thermal Vias. To be presented at ITherm , San-Diego, CA, May 30, 2012.