Foundations and Trends® in
Electronic Design Automation
Vol. 2, No. 3 (2007) 255–370

DOI: 10.1561/1500000007

now
the essence of knowledge

# Thermally Aware Design

**Yong Zhan[1], Sanjay V. Kumar[2] and Sachin S. Sapatnekar[3]**

[1] *Cadence Design Systems, 555 River Oaks Parkway, San Jose, CA 95134, USA, yongzhan@cadence.com*
[2] *Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, sanjay@umn.edu*
[3] *Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, sachin@umn.edu*

## Abstract

With greater integration, the power dissipation in integrated circuits has begun to outpace the ability of today's heat sinks to limit the on-chip temperature. As a result, thermal issues have come to the forefront, and thermally aware design techniques are likely to play a major role in the future. While improved heat sink technologies are available, economic considerations restrict them from being widely deployed until and unless they become more cost-effective. Low power design is helpful in controlling on-chip temperatures, but is already widely utilized, and new thermal-specific approaches are necessary. In short, the onus on thermal management is beginning to move from the package designer toward the chip designer. This survey provides an overview of analysis and optimization techniques for thermally aware design. After beginning with a motivation for the problem and trends seen in the semiconductor industry, the survey presents a description

of techniques for on-chip thermal analysis. Next, the effects of elevated temperatures on on-chip performance metrics are analyzed. Finally, a set of thermal optimization techniques, for controlling on-chip temperatures and limiting the level to which they degrade circuit performance, are described.

# 1

# Introduction

## 1.1 Overview

Thermal analysis is important in ensuring the accuracy of timing, noise, and reliability analyses during chip design. The thermal properties of integrated systems can be studied at a number of levels and length scales, as partly illustrated in Figure 1.1. For the problem of cooling racks of computing servers in a data center, the cooling structure must cover an area of the order of meter to tens of meters. At the next level, board-level cooling operates at length scales of the order of a tenth of a meter, while package level cooling corresponds to lengths of the order of centimeters. Within-chip solutions include microrefrigeration solutions, whose sizes range from the order of a millimeter to a centimeter [130] and can operate at the architectural level, to solutions that can scale down to several tens of microns [129] and operate at about the logic level.

In other words, the thermal problem is important at a wide range of length scales, and known cooling solutions exist at all of these levels. These solutions range in complexity and cost from the use of passive heat sinks, to active convective cooling using fans, to more exotic


Fig. 1.1 Manifestations of the thermal problem at a variety of length scales [72].

technologies based on microchannels and microrefrigeration. Some of these solutions are more traditional and have been available, in some form, for quite a few years, while others are relatively newer, and are actively being researched.

The genesis of thermal problems is in the fact that electronic circuitry dissipates power. This power dissipated on-chip is manifested in the form of heat, which, in a reasonably designed system, flows toward a heat sink. The power generated per unit area is often referred to as the heat flux. Temperature and power (or heat flux) are intimately related, but it is important to note that they are distinct from each other. For instance:

- For the same total power, it is possible to build systems with different peak temperature and heat flux distributions, simply by changing the spatial arrangement of the power sources. If all the high power sources are concentrated together in a region, that area will probably see a high peak temperature. Such a thermal bottleneck can often be relieved by moving the power sources apart.

- The relative location of the power sources to the heat sink also plays a part in determining the on-chip temperature, and by providing high-power elements with a conductive path to the heat sink, many thermal problems can be alleviated.

Most such optimizations can result in tradeoffs: for instance, thermal considerations imply that highly active units should be moved apart, but if these units communicate with each other, performance requirements may dictate that they be kept close to each other.

The focus of this survey is on presenting solutions for the within-chip thermal problem. However, an essential prerequisite to addressing thermal issues is the ability to model heat transfer paths of a chip with its surrounding environment, and to analyze the entire thermal system, including effects that are not entirely within the chip. Figure 1.1 shows a representative chip in a ceramic ball grid array (CBGA) packaging and its surrounding environment. This is modeled as a network of thermal resistors, using the thermal–electrical analogy to be described in Section 2.3.1, where power values map on to electrical currents, temperatures to voltages, and the ground node to the ambient.

In Figure 1.2, the chip is placed over a ceramic substrate, connected through flip-chip, or C4, connections all over its area. The substrate is connected to the printed circuit board through CBGA connections, and a small portion of the heat generated on-chip flows through this region to the ambient, which is denoted by the ground connection in the thermal circuit. At the other end, a heat sink with a large surface area is connected to the chip, with a thermal interface material lying between the chip and the heat sink. The role of the thermal interface material (TIM) is to act as a heat spreader. In the absence of this material, the surface roughness of the chip and the heat sink imply that the actual contact between the two surfaces could be as low as 1%–2% of the apparent surface area [110], and this is accentuated by warpage of the die under thermal stress; adding that the TIM improves the contact area, and consequently, the thermal resistance at this interface. The upper half of the thermal circuit shows how this region can be modeled.

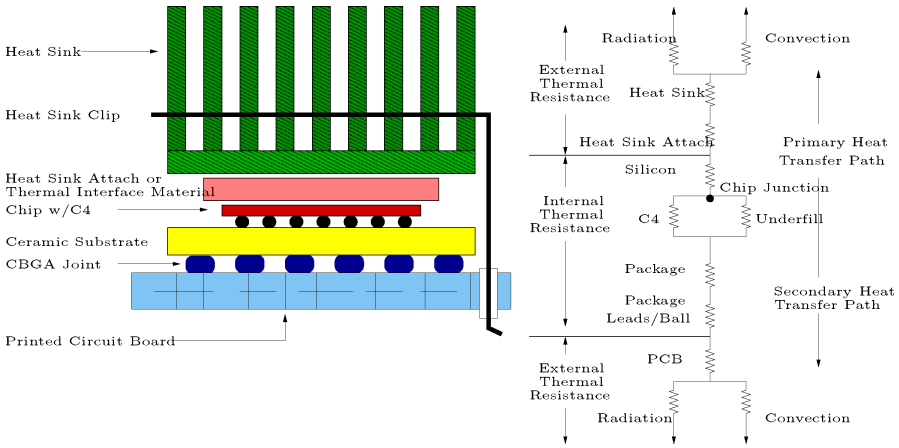



Fig. 1.2 Heat dissipation paths of a chip in a system.

Additional air-cooling schemes, such as fans that are connected to the heat sink, can be incorporated into this model.

A crucial step in design involves the choice of a heat sink: here, approximate techniques may be used to obtain a reasonable sinking solution [88], based on a gross characterization of the sink by its thermal resistance. At the package design level, a key item of interest is the thermal design power (TDP), which is the maximum sustained power dissipated by an integrated circuit. This is not necessarily the peak power: if the peak power is dissipated for a small period of time that is below the thermal time constant, it does not appreciably affect the choice of the heat sink. If the peak temperature is to be maintained at a temperature $T_{\text{peak}}$ above the ambient temperature, then the maximum thermal resistance of the heat sink is given by a simple formula based on a lumped DC analysis:

$$R_{\text{sink}} = T_{\text{peak}}/\text{TDP}. \tag{1.1}$$

The choice of the heat sink can be made on the basis of this requirement. Note that this is a very coarse analysis that does not consider transient effects.

## 1.2 Thermal Trends

### 1.2.1 The Importance of Temperature as a Design Consideration

The problem of getting the heat out of a chip is not new: indeed, power issues have been at least partially if not wholly responsible for the demise of a variety of technologies before CMOS. For instance, as demonstrated in Figure 1.3, the rapidly increasing power dissipation trends in bipolar circuits played a large role in their displacement as the dominant technology of the day, being taken over first by NMOS and then by CMOS. Today, no clear successor to CMOS has emerged, but on-chip power dissipation has emerged as a major design bottleneck, and it is ever more important to build cooling solutions from the system level down to the subchip level. Historical trends, illustrated in Figure 1.3, show an ever-increasing profile for the volume of the external heat sink as on-chip power goes up, and this is unsustainable.

As illustrated in Figure 1.4, trends show that the cost of the cooling solution is a nonlinear function of the chip power dissipation: the initial rise is gentle, but beyond the point of convective cooling, the costs rise steeply. This knee point is a function of cooling system complexity and

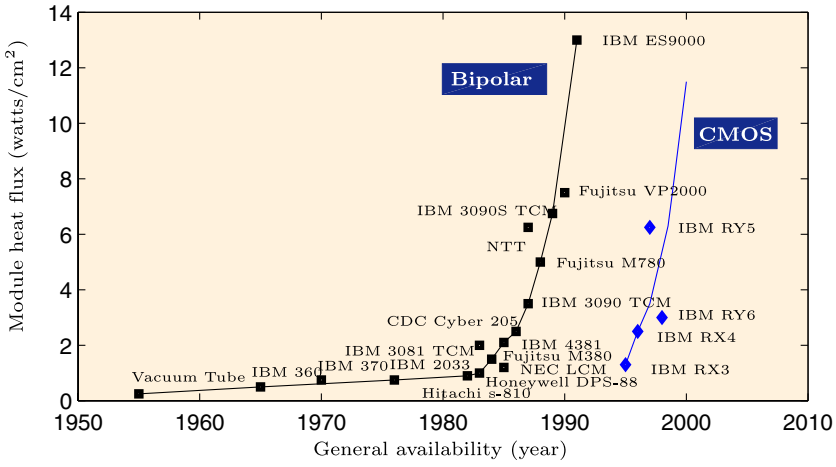



Fig. 1.3 Trends for the heat flux for state-of-the-art systems over the years [27].

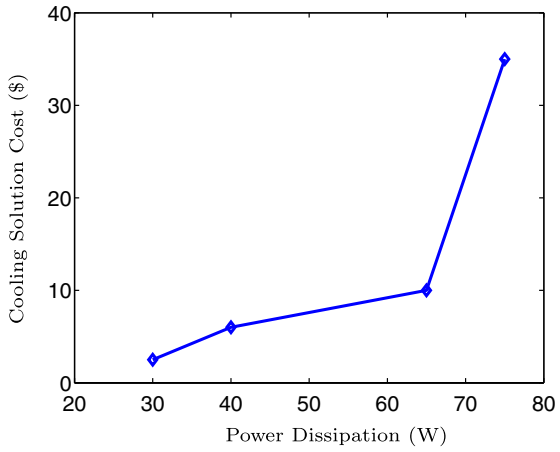



Fig. 1.4 Cooling costs as a function of the power dissipation [57].

the volume of actively cooled packages/technologies, and may arguably permit slightly higher on-chip power dissipation in the future as newer technologies gain economies of scale, but the fundamental nature of the curve — of having a gentle ascent followed by a steep rise beyond a knee point — is unlikely to change. This has consequences on the size of the heat sink, and Figure 1.5 shows how the volume of the heat sink has increased with increased on-chip power.

To achieve the required heat sinking solution, it may be necessary to increase the heat sink size to unreasonable levels, or to move to new cooling technologies. For contemporary high-end, large-volume parts, anything that is more complex than air-cooling is probably too expensive. Although several of the bipolar chips in Figure 1.3, after 1980, used some form of water cooling [27], liquid cooling is not seen as a very viable solution today. There have been numerous improvements even in air-cooled technologies and improved thermal interface materials in the recent past, which have progressively shifted the knee of the cooling cost curve of Figure 1.4 progressively to the right, so that the heat fluxes that are currently obtained by air cooling could only be achieved by liquid cooling in the 1980s [120]. However, even these improvements cannot keep up with the capability of Moore's law to integrate more functionality on a chip. Indeed, while it is possible to

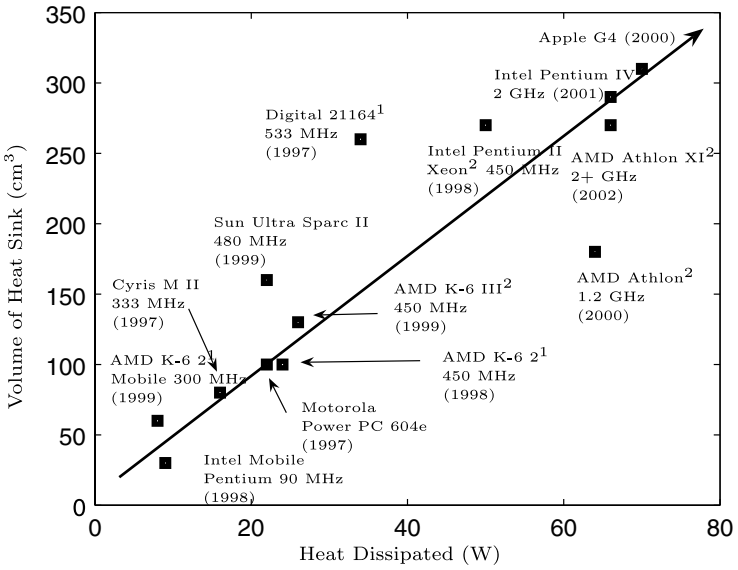



Fig. 1.5 Historical data showing the volume of the heat sink as a function of the total on-chip power [72].

pack more transistors on a chip today, only a fraction of them can actually be used to full potential, because of power and thermal limitations. More advanced solutions using, for example microfluidic channels and microrefrigeration, have been proposed, these are not cost-effective enough for widespread use today.

### 1.2.2 Thermal Issues in 3D Integrated Circuits

The previous subsection explains why temperature must be an important consideration in the design of nanoscale integrated circuits. A further motivator for thermally conscious design has come about with the advent of three dimensional (3D) integration, which makes the on-chip problem particularly acute.

Unlike conventional 2D circuits, where all transistors are placed in a single plane, with several layers of interconnect above, 3D circuits stack tiers of such 2D structures, one above the other. 3D structures may be built by stacking tiers of dies above each other, where the separation between tiers equals the thickness of the bulk substrate, which is of

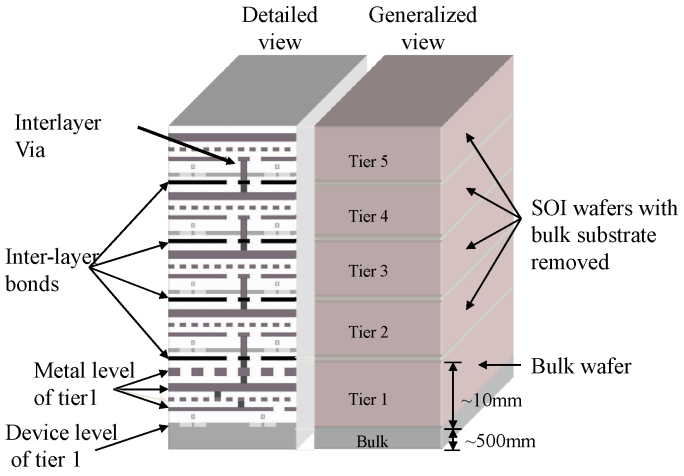



Fig. 1.6 A schematic of a 3D integrated circuit.

the order of several hundreds of microns. Advances in industrial [54], government [18], and academic [117] research laboratories have demonstrated 3D designs with inter-tier separations of the order of a few microns, enabling short connections between tiers, accentuating the advantages of short vertical interconnections in these 3D structures. A schematic of a 3D chip is illustrated in Figure 1.6, showing five tiers stacked over each other. The lowest tier sits over a bulk substrate, while the other tiers are thinned to remove the substrate, and provide inter-tier distances of the order of ten microns.

With these technological advances, 3D technologies provide a roadmap for allowing increased levels of integration within the same footprint, in a direction that is orthogonal to Moore's law. Moreover, 3D technologies provide the ability to locate critical blocks close to each other, e.g., by placing memory units in close proximity to processors by placing them one above the other. These, and other, advantages make 3D a promising technology for the near future.

However, the increased packing density afforded by 3D integration has the drawback of exacerbating thermal issues. Based on a simple back-of-the-envelope calculation, a $k$-tier 3D chip could use $k$ times as much current as a single 2D chip of the same footprint; however, the

packaging technology is not appreciably different. This implies that the corresponding heat generated must be sent out to the environment using a package with essentially similar heat sinking capabilities. As a result, the on-chip temperature on a 3D chip could be $k$ times higher than the 2D chip. While this is a very coarse analysis with very coarse assumptions, the eventual conclusion — that thermal effects are a major concern in 3D circuits — is certainly a strong motivator for paying increased attention to thermal issues today.

## 1.3 Organization of the Survey

This survey begins its discussion of on-chip thermal effects by surveying techniques for evaluating the distribution of temperature on a chip. These analysis techniques essentially solve a partial differential equation (PDE) that relates the power dissipated on a chip to its temperature profile. While the solution of PDEs is a well-studied problem, it is possible to take advantage of some specific properties of the on-chip problem to obtain an efficient solution. Moreover, thermal analysis shows similarities to other well-studied problems in integrated circuit design, most notably, those of analyzing on-chip power grids [126], and of substrate analysis [46, 36], and techniques from these domains can be borrowed to enhance the quality of algorithms for thermal analysis.

Next, we study the manner in which on-chip temperatures affect the properties and performance of a circuit. In terms of delay, the performance of transistors and the resistance of interconnect wires can be affected; in terms of power dissipation, there is a strong relationship, with potential feedback, between temperature and leakage power; in terms of reliability, the lifetime of both devices and interconnects all depend critically on the operating temperature of the circuit. These are all critical factors in ensuring circuit performance, and the complexity of these problems makes it essential to build efficient and scalable CAD solutions for on-chip thermal analysis. Finally, we overview some representative techniques for thermally driven circuit optimization.

# 2

# Thermal Analysis Algorithms

## 2.1 Overview

A basic requirement for addressing on-chip thermal issues is to develop the ability to analyze and predict the thermal behavior of an integrated circuit. Several algorithms for on-chip thermal analysis have been presented in the literature, for application areas ranging from architecture level analysis to cell level analysis to transistor-level analysis. This section overviews these methods and provides the background required to understand techniques for formulating and solving the thermal PDE.

The foundations of thermal analysis lie in classical heat transfer theory, which has been well-studied over the years. However, there are several features that are specific to the on-chip context that a thermal analyzer may take advantage of. For example:

- On-chip geometries are strongly rectilinear in nature: for example, all wires are routed in the coordinate directions in mainstream technologies; standard cells are generally rectilinear in outline, as are many functional blocks on a chip. Therefore, it is reasonable to assume, at above the device

level, that the topologies that we will deal with are rectilinear. This implies the existence of a certain set of rectangular geometric symmetries that may be leveraged.

- The devices, which are the major heat-producing elements, lie in either a single layer (in classical 2D technologies), or in multiple layers (in 3D technologies) atop a substrate, and the points at which a user is typically interested in analyzing temperature are within the device layer(s). This implies that the substrate, which is much more voluminous than the device layer, can be abstracted away into a macromodel. This is particularly useful when repeated evaluations are necessary, corresponding to, for example, different placements within the same die area.
- Due to the top-down nature of the IC design process, the level of accuracy of the power estimates vary at different steps in the design cycle. Correspondingly, the accuracy requirements are also variable. Generally speaking, early steps of the design cycle can be very influential in impacting the overall performance of the design: in fact, the most effective optimizations can be made at these stages, but they must necessarily operate under limited information. A coarser analysis is appropriate at this stage. At later steps in the design cycle, as more of the design becomes concrete, the information that is available is more detailed, but the flexibility to make changes is limited. The role of detailed thermal analysis at this stage is to ensure that the assumptions made in early stages of design are maintained (or improved upon), so that high-level optimizations made under these assumptions can be effective.

This survey presents an overview of techniques for full-chip thermal analysis. Another important problem, that of Joule heating in wires, is not covered here, but the reader is referred to, for example, [3, 4, 26] for further guidance on this topic. We begin with an explanation of the underlying PDEs that describe the thermal system. Next, we describe techniques for solving these PDEs in the steady-state, followed by a description of techniques for solving the linear equations that arise

from discretizing the PDEs. Finally, we describe techniques for solving the transient analysis problem.

## 2.2 The Thermal PDE

### 2.2.1 Overview

The problem of on-chip thermal analysis requires the solution of the heat equation, which is described in terms of a PDE. Most commonly, PDEs are solved using a discretization or meshing scheme, which converts the problem to one of solving a set of algebraic equations that are typically linear. Several approaches and algorithms for thermal analysis have been published in the literature. Depending on the context, the stage of design, and the degree of accuracy that is necessary, different classes of algorithms may be appropriate. Various choices are possible along the way, in the PDE that is used to describe the thermal analysis problem, in the solution framework and discretization scheme for the numerical solution of the PDE, and in the solution technique for the resulting equations after discretization. In many cases, our methods will focus on uniform discretization for simplicity, but it is understood that there is a need for nonuniform discretization, to obtain more accurate solutions to temperature-sensitive parts of a chip.

The precise PDE that is used to model the thermal problem depends on the length scale being considered. For full-chip thermal analysis, the analysis is necessarily performed at a coarse-grained level, and macroscale analysis based on the Fourier equation is adequate. However, if the analysis is to be performed at small length scales, then a more detailed solution that takes quantum-mechanical effects into account is required. Another choice is related to determining whether the thermal problem is to be solved for the steady-state or the transient case. The former assumes that the temperature waveforms are steady over time, so that expressions related to time-derivatives can be ignored, while the latter presents the temporal response to (possibly) time-varying input stimuli. In other words, transient analysis solves the full PDE, considering both the space variable $\mathbf{r}$ and the time variable $t$ as independent variables, unlike steady-state analysis, which sets all partial derivatives with respect to $t$ in the full PDE to zero and thus

entirely removes the time variable $t$ from the equation. To the reader schooled in circuit analysis, these ideas will seem familiar, since they parallel the notion of steady-state and transient simulation in circuit analysis.

This survey is primarily directed toward solving full-chip thermal analysis problems, which implies that the underlying PDE is the heat equation, as derived from Fourier's law (although we provide a brief review of the transistor-level analysis problem). To solve this problem numerically, one of several frameworks for PDE solution may be employed, such as the finite difference method (FDM), the finite element method (FEM), and the Green function method. In case of the FDM and FEM, a set of linear equations is generated, which must be solved to determine the temperature distribution within the system. Depending on the properties of the linear equations, they can be solved by, for example, a direct method based on LU/Cholesky factorization, or an iterative method such as ICCG and GMRES, or a random walk based method, as described in Section 2.4.

### 2.2.2 Macroscale Fourier-based Analysis

Conventional heat transfer in a chip is described by Fourier's law of conduction [103], which states that the heat flux, $q$ (in W/m$^2$), is proportional to the negative gradient of the temperature, $T$ (in K), with the constant of proportionality corresponding to the thermal conductivity of the material, $k_t$ (in W/(m K)), i.e.,

$$q = -k_t \nabla T. \tag{2.1}$$

The divergence of $q$ in a region is the difference between the power generated and the time rate of change of heat energy in the region. In other words,

$$\nabla \cdot q = -k_t \nabla \cdot \nabla T = -k_t \nabla^2 T = g(\mathbf{r}, t) - \rho c_p \frac{\partial T(\mathbf{r}, t)}{\partial t}. \tag{2.2}$$

Here, $\mathbf{r}$ is the spatial coordinate of the point at which the temperature is being determined, $t$ represents time (in sec), $g$ is the power density per unit volume (in W/m$^3$), $c_p$ is the heat capacity of the chip material

(in J/(kg K)), and $\rho$ is the density of the material (in kg/m$^3$). This may be rewritten as the following heat equation, which is a parabolic PDE:

$$\rho c_p \frac{\partial T(\mathbf{r},t)}{\partial t} = k_t \nabla^2 T(\mathbf{r},t) + g(\mathbf{r},t). \tag{2.3}$$

The thermal conductivity, $k_t$, in a uniform medium is isotropic, and thermal conductivity values for silicon, silicon dioxide, and metals such as aluminum and copper are fundamental material properties whose values can be determined from standard tables. In practice, in early stages of analysis and for optimization purposes, integrated circuits may be assumed to be layer-wise uniform in terms of thermal conductivity. The bulk layer has the conductivity of bulk silicon, and the conductivity of the metal layers is often computed using an averaging approach: this region consists of a mix of silicon dioxide and metal, and depending on the metal density within the region, an effective thermal conductivity may be used for macroscale analysis. The value of $k_t$ is weakly dependent on temperature, which makes the heat equation nonlinear. However, for on-chip analysis, this nonlinearity is often ignored, although approaches to solving thermal problems under nonlinear thermal conductivities [11] have been proposed. In many applications, the precise worst-case power patterns have some uncertainty associated with them, so that such approximations are acceptable.

The solution to Equation (2.3) corresponds to the transient thermal response. In the steady state, all derivatives with respect to time go to zero, and therefore, steady-state analysis corresponds to solving the PDE:

$$\nabla^2 T(\mathbf{r}) = -\frac{g(\mathbf{r})}{k_t}. \tag{2.4}$$

This is the well-known Poisson's equation.

The time constants of heat transfer are of the order of milliseconds, and are much longer than the subnanosecond clock periods in today's VLSI circuits. Therefore, if a circuit remains within the same power mode for an extended period of time, and its power density distribution remains relatively constant, steady-state analysis can capture the thermal behavior of the circuit accurately. Even if this is not the case, steady-state analysis can be particularly useful for early and more

approximate analysis, in the same spirit that steady-state analysis is used to analyze power grid networks early in the design cycle. On the other hand, when greater levels of detail about the inputs are available, or when a circuit makes a number of changes between power modes at time intervals above the thermal time constant, transient analysis is possible and potentially useful.

To obtain a well-defined solution to Equation (2.3), a set of boundary conditions must be imposed. The most generally used boundary conditions in thermal analysis for chip design are described in Dirichlet form, specifying information on the boundary, $\Gamma$, of the region. In the succeeding discussion, we will use $T_c$ to denote a constant temperature, $n$ for the outward normal direction of the boundary surface, $h$ is the effective heat transfer coefficient of the ambient, and $T_a$ is the temperature of the ambient. Examples of typical boundary conditions for the on-chip thermal analysis problem are listed below (for details, the reader is referred to a standard text on heat transfer, such as [103]):

- The *isothermal* boundary condition can be applied when a surface of the chip is attached to a constant temperature heat reservoir with a significantly larger heat capacity and higher thermal conductivity than those of the chip itself, and is specified as:

$$T(\mathbf{r}, t) = T_c \quad \text{where } \mathbf{r} \in \Gamma. \tag{2.5}$$

  This boundary condition is sometimes used to model the effect of the heat spreader and heat sink in the chip-package stack shown in Figure 2.1. However, the assumption of constant temperature at the boundary that is made here is an approximation in real designs that is not appropriate when higher accuracies are necessary.
- The *heat flux* boundary condition can be applied when power sources are placed on a surface of the chip. For example, if the heat transfer within interconnect layers is ignored, the power generating devices can be modeled using the following heat flux boundary conditions as far as the silicon substrate

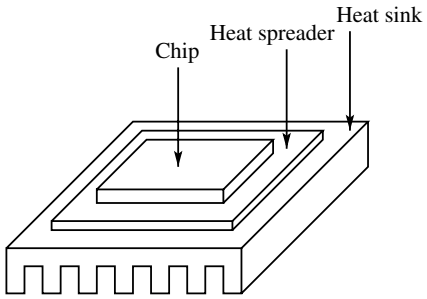



Fig. 2.1 Schematic of an IC chip with the associated packaging.

is concerned:

$$k_t \frac{\partial T(\mathbf{r},t)}{\partial n} = p(\mathbf{r},t) \quad \text{where } \mathbf{r} \in \Gamma, \tag{2.6}$$

where $p(\mathbf{r},t)$ is the heat flux at a boundary point $\mathbf{r}$, as a function of time. In this case, the heat source is idealized to be part of the boundary conditions; more realistically, the heat source is part of the material structure being studied.

- A special case of the heat flux boundary condition has proven to be rather useful, i.e., the *adiabatic* boundary condition, which can be obtained by setting the power term to zero in the heat flux condition. The adiabatic boundary condition can be applied when there is no heat exchange across the surface of the chip, and it is often a good approximation when the corresponding surface of the chip is covered by thick oxide, which is a thermal insulator, or for heat exchange on the sides of a chip.
- The *convective* boundary condition is often used to model the effect of the heat spreader and heat sink in thermal analysis, and can be written as:

$$-k_t \frac{\partial T(\mathbf{r},t)}{\partial n} = h_c(T(\mathbf{r},t) - T_a) \quad \text{where } \mathbf{r} \in \Gamma. \tag{2.7}$$

This condition states that the heat flow to the surface from the chip side equals the heat dissipated to the ambient by the heat sink and the surrounding air. It is worth noticing

that the effective heat transfer coefficient $h_c$ used in thermal analysis for chip design is a property of the heat sink, and is usually much larger than that of the air, due to the fact that the heat sink has significantly increased the effective heat dissipation area of the chip.

When a more detailed package model is available, it may be appended into the overall thermal analysis approach. For example, Hotspot [60, 61, 62, 63, 64, 135, 134] demonstrates how a package model can be inserted into a coarse FDM solver; the boundary conditions for the package are taken to be isothermal.

### 2.2.3 Nanoscale Thermal Analysis

The condition under which Equation (2.3) can be applied is in the Fourier regime, where it is reasonable to assume that local thermodynamic equilibrium can be maintained. Here, "local thermodynamic equilibrium" refers to the case where the properties of the system such as the distribution of energy-carrying particles as a function of location $\mathbf{r}$ and velocity $\mathbf{v}$ may change with space and time, but they change so slowly that any macroscopically small but microscopically large region within the system can be well approximated by an equilibrium state at any time. For systems violating such conditions, quantum-mechanical effects must be taken into account. The material in this section draws strongly from [109], which is commended to the reader for further details. Since the focus of this survey is on macroscale thermal effects, only a brief overview is provided in this section, for completeness.

The local thermodynamic equilibrium assumption is valid for coarse-grained, system-level or full-chip thermal analysis, but for fine-grained analysis of systems with aggressively scaled devices under very strong electric fields, the systems may show nonequilibrium properties, so that the application of Equation (2.3) can underestimate the effective temperatures of the hot spots [121]. At these microscales, it is necessary to consider the effects of electron–phonon interactions. Phonons essentially correspond to energy due to lattice vibrations in a crystal, and their effects are quantum-mechanical. As device dimensions become comparable to the mean free path of electrons and phonons

(whose values are, respectively, about 5–10 nm and 200–300 nm for bulk silicon at 300 K [109]), it is necessary to consider their effects on fine-grained thermal analysis.

This mechanism can be summarized as follows: the applied electric field causes electrons to accelerate as they reach the drain of a transistor, causing them to gain energy and heat up. This can lead to electron population to lose energy due to scattering with lattice vibrations (i.e., phonons), which results in Joule heating as the lattice is heated up and gains temperature. The phonons further affect the electron mobility: in essence, the electrons and phonons affect behavior of each other. Low energy electrons scatter with lower-frequency acoustic phonons, while higher energy electrons scatter with the higher-frequency optical phonons. Acoustic phonons move faster through silicon, they dominate heat transfer in silicon in tenths of picoseconds, while optical phonons degrade over longer periods of time (of the order of picoseconds) to acoustic modes, which can cause temperature rises.

At the fine-grained level within a transistor, power dissipation in CMOS transistors occurs in tens of picoseconds, during a switching event, but Fourier conduction modes have time constants of the order of tens of nanoseconds across the length of a gate. This implies that due to the electron–phonon interactions, the local thermodynamic equilibrium assumption does not hold at these length scales. Temperature is a statistical quantity that depends on an equilibrium state, and these length and time scales disallow a continuum assumption for heat transfer. Instead, the distribution of phonons can be used to determine an "effective temperature" by equating the nonequilibrium energy density to that from Bose–Einstein statistics.

To determine phonon distributions, one can solve the Boltzmann transport equation

$$\begin{aligned}\frac{\partial N_{\mathbf{q}}(\mathbf{r},t)}{\partial t} + \mathbf{v}_g{\cdot}\nabla_{\mathbf{r}} N_{\mathbf{q}}(\mathbf{r},t) + \mathbf{F}{\cdot}\nabla_{\hbar\mathbf{q}} N_{\mathbf{q}}(\mathbf{r},t) \\ = \frac{\partial N_{\mathbf{q}}(\mathbf{r},t)}{\partial t}\Big|_{\text{collision}} + \frac{\partial N_{\mathbf{q}}(\mathbf{r},t)}{\partial t}\Big|_{\text{generation}}. \qquad (2.8)\end{aligned}$$

This has to be used to study the movement of energy-carrying particles and the phenomenon of heat transfer. Here, $N_{\mathbf{q}}(\mathbf{r},t)$ is the number

of particles in a particular mode with wave vector $\mathbf{q}$ at position $\mathbf{r}$ and time $t$, $\mathbf{v}_g$ is the group velocity of that mode, and $\mathbf{F}$ is an external force acting on the particle. The two terms on the right-hand side of the equation correspond to changes in the number of particles in that mode due to collisions and the generation of particles. The Boltzmann transport Equation (2.8) is valid for the semi-classical transport regime where charge and energy carriers can be treated as particles between scattering events but the frequency and nature of the scattering is described using quantum mechanics. Because the collision and generation terms are not known *a priori*, the solution of the Boltzmann transport equation often involves some Monte Carlo type of simulations of the actual interactions between individual particles.

The work in [121], for example, analyzes the thermal effects in nanoscale transistors by splitting the analysis into two sub-systems, i.e., the electron sub-system and the phonon sub-system. The electrons are handled using an electron Monte Carlo (EMC) technique based on [108], while the phonons are described using a split-flux Boltzmann transport equation (SF-BTE) [132]. In each iteration of thermal analysis, two independent simulations are performed, one for electrons and the other for phonons. The output of each simulation is fed back to the opposite sub-system and the thermal analysis proceeds until the iterations converge.

## 2.3 Steady-state Thermal Analysis Algorithms

The goal of steady-state thermal analysis is to determine the temperature distribution within a chip given a power density distribution that does not change with time. Mathematically, the steady-state temperature distribution is governed by Poisson's equation (2.4) under a set of boundary conditions. In this section, we will describe steady-state analysis techniques based on the application of finite difference method (FDM), the finite element method (FEM), and Green functions.

The FDM and FEM methods both discretize the entire chip and form a system of linear equations relating the temperature distribution within the chip to the power density distribution. The major difference between the FDM and FEM is that while the FDM discretizes the

differential operator, the FEM discretizes the temperature field. The primary advantage of the FDM and FEM is their capability of handling complicated material structures, particularly nonuniform interconnect distributions in a VLSI chip. However, a direct application of the FDM or the FEM may be computationally wasteful since it may discretize the entire volume of the chip. Such an approach fails to recognize that the regions on a chip where power is generated, or whose temperature is of interest, are usually located only on a few discrete planes, e.g., the device layer and interconnect layers. In other words, a direct application of these methods will result in intimidatingly large systems of algebraic equations, which can take a large number of CPU cycles to solve, leading to relatively low efficiencies in thermal analysis. Using macromodeling techniques, it is possible to abstract away [139] the nodes in the FDM and FEM meshes that the users of the thermal simulator are not interested in, although it should be noted that this macromodeling procedure can be computationally intensive.

In contrast with these methods, the Green function method provides a semi-analytical solution in which only the layers where the power is generated or whose temperature is of interest are analyzed. Therefore, the resulting problem size is usually small compared with that of the FDM and FEM, which reduces the time that is needed to reach a solution to the problem. However, this improvement comes at a cost: the application of the Green function method is usually based on the assumption that the chip materials are layer-wise uniform, which is often not satisfied especially for interconnect layers. As a result, the Green function method is more suitable for early stages of physical design, where the accuracy requirement on thermal analysis is moderate but the efficiency requirement is high. The computational efficiency is driven by the fact that no algebraic equations have to be solved in the Green function method after the Green function has been determined. Therefore, a significant portion of research works related to the application of the Green function method in thermal analysis for chip design have been directed toward the fast evaluations of the Green function.

### 2.3.1 The Finite Difference Method

#### 2.3.1.1 Formulation of the FDM Equations

The steady-state Poisson's Equation (2.4) can be discretized by writing the second spatial derivative of the temperature, $T$, as a finite difference in rectangular coordinates. The spatial region may be discretized into rectangles, each represented by a node, with sides along the $x$, $y$, and $z$ directions, with lengths $\Delta x$, $\Delta y$, and $\Delta z$, respectively. Let us assume that the region of interest is placed in the first octant, with a vertex at the origin. We will use $T_{i,j,k}$ to represent the steady-state temperature at node $(i\,\Delta x, j\,\Delta y, k\,\Delta z)$, and there is one equation associated with each node inside the chip.

This discretization can be used to write an approximation for the spatial partial derivatives. For example, in the $x$ direction, we can write

$$\frac{\partial^2 T(\mathbf{r})}{\partial^2 x} \approx \frac{\frac{T_{i-1,j,k}-T_{i,j,k}}{\Delta x} - \frac{T_{i,j,k}-T_{i+1,j,k}}{\Delta x}}{\Delta x} \tag{2.9}$$

$$= \frac{T_{i-1,j,k} - 2T_{i,j,k} + T_{i+1,j,k}}{(\Delta x)^2}. \tag{2.10}$$

Similar equations may be written in the $y$ and $z$ spatial directions.

Let us define the operators $\delta_x^2$, $\delta_y^2$, and $\delta_z^2$ as

$$\begin{aligned} \delta_x^2 T_{i,j,k} &= T_{i-1,j,k} - 2T_{i,j,k} + T_{i+1,j,k}, \\ \delta_y^2 T_{i,j,k} &= T_{i,j-1,k} - 2T_{i,j,k} + T_{i,j+1,k}, \\ \delta_z^2 T_{i,j,k} &= T_{i,j,k-1} - 2T_{i,j,k} + T_{i,j,k+1}. \end{aligned} \tag{2.11}$$

The FDM discretization of Poisson's equation using finite differences thus results in the following of linear equations:

$$\frac{\delta_x^2 T_{i,j,k}}{(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}}{(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}}{(\Delta z)^2} = -\frac{g_{i,j,k}}{k_t}. \tag{2.12}$$

A better visualization of the discretization process, particularly for an electrical engineering audience, employs another standard device in heat transfer theory that builds an equivalent *thermal circuit* through

the so-called *thermal–electrical analogy*. Each node in the discretization corresponds to a node in the circuit. The steady-state equation corresponds to a network where "thermal resistors" are connected between nodes that correspond to spatially adjacent regions, and "thermal current sources" that map on to power sources. The voltages at the nodes in this thermal circuit can then be computed by solving the circuit, and these yield the temperature at that node. Mathematically, this can be derived from Equation (2.4) by writing the discretized equation in a slightly different form from Equation (2.12). For example, in the $x$ direction, the finite difference in Equation (2.9) can be rewritten as

$$\frac{\partial^2 T(\mathbf{r})}{\partial^2 x} \approx \left[ \frac{T_{i-1,j,k} - T_{i,j,k}}{R_{i-1,j,k}} - \frac{T_{i,j,k} - T_{i+1,j,k}}{R_{i,j,k}} \right] \cdot \frac{1}{k_t A_x \Delta x}, \tag{2.13}$$

where $R_{i-1,j,k} = \frac{\Delta x}{k_t A_x}$ and $A_x = \Delta y \Delta z$ is the cross-sectional area of the element when sliced along the $x$-axis, to obtain the following discretization:

$$\begin{aligned} &\left[ \frac{T_{i-1,j,k} - T_{i,j,k}}{R_{i-1,j,k}} + \frac{T_{i+1,j,k} - T_{i,j,k}}{R_{i,j,k}} \right] \\ &\quad + \left[ \frac{T_{i,j-1,k} - T_{i,j,k}}{R_{i,j-1,k}} + \frac{T_{i,j+1,k} - T_{i,j,k}}{R_{i,j,k}} \right] \\ &\quad + \left[ \frac{T_{i,j,k-1} - T_{i,j,k}}{R_{i,j,k-1}} + \frac{T_{i,j,k+1} - T_{i,j,k}}{R_{i,j,k}} \right] = -G_{i,j,k}, \end{aligned} \tag{2.14}$$

where $G_{i,j,k} = g_{i,j,k} \Delta V$ is the total power generated within the element, and $\Delta V = A_x \Delta x = A_y \Delta y = A_z \Delta z$.

The representation (2.14) can readily be recognized as being equivalent to the nodal equations at each node of an electrical circuit, where the node is connected to the nodes corresponding to its six adjacent elements through thermal resistors, as shown in Figure 2.2. In other words, the solution to the thermal analysis problem using FDM amounts to the solution of a circuit of linear resistors and current sources.

The ground node, or reference, for the circuit corresponds to a constant temperature node, which is typically the ambient temperature.

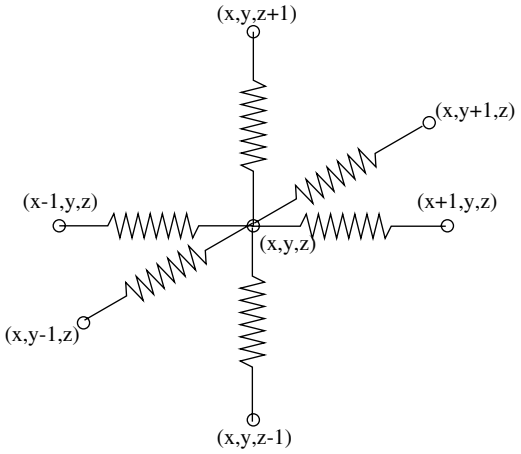



Fig. 2.2 Thermal resistances connected to a node $(x, y, z)$ after FDM discretization.

If isothermal boundary conditions are to be used, this simply implies that the node(s) connected to the ambient correspond to the ground node. On the other hand, it is possible to use a more detailed thermal model for the package and heat spreader, consisting of an interconnection of thermal resistors and thermal capacitors, as used in HotSpot [60, 61, 62, 63, 64, 134, 135], or another type of compact model such as a reduced-order model. In either case, one or more nodes of the package model will be connected to the ambient, which is taken to be the ground node. Such a model can be obtained by applying, for example, the FDM or FEM on the package, and extracting a (possibly sparsified) macromodel that preserves the ports that connect the package to the chip and to the ambient.

The overall equations for the circuit may be formulated using modified nodal analysis [28], and we may obtain a set of equations

$$G\mathbf{T} = \mathbf{P}. \tag{2.15}$$

Here $G$ is an $n \times n$ matrix and $\mathbf{T}, \mathbf{P}$ are $n$-vectors, where $n$ corresponds to the number of nodes in the circuit. It is easy to verify that the $G$ matrix is a sparse conductance matrix that has a banded structure, is symmetric and is diagonally dominant.

#### 2.3.1.2 Building a Compact Model for the Substrate

The finite difference approach was utilized for on-chip thermal analysis in [139], which also realized that a large number of nodes in the temperature vector can be eliminated. Specifically, these nodes in the finite difference discretization can be classified into two categories: those on the surface of the region, in the region where the active devices lie, and those within the substrate, which we will call the internal nodes. In most cases, the temperatures at internal nodes are not of interest, and the power density at these nodes is zero. Therefore, the corresponding variables can be eliminated to build a compact model for the thermal system.

Under this classification, Equation (2.15) can be rewritten, by partitioning $G$ into submatrices, as

$$\begin{bmatrix} G_P & G_C^T \\ G_C & G_I \end{bmatrix} \begin{bmatrix} T \\ T_I \end{bmatrix} = \begin{bmatrix} P \\ 0 \end{bmatrix}, \tag{2.16}$$

where $T_I$ is the vector of temperatures at the internal nodes, and $T$ corresponds to the temperatures at the remaining nodes. Note that the right-hand side vector shows zero power at the internal nodes. The matrix $G$ is appropriately partitioned according to the classification.

Eliminating the temperatures at the internal nodes from Equation (2.16), we have

$$\begin{aligned} G'\mathbf{T} &= P, \\ \text{where } G' &= G_P - G_C^T G_I^{-1} G_C \end{aligned}. \tag{2.17}$$

If the total number of nodes in $n$, and $n - m$ of these are internal nodes, then the computational cost for directly calculating $G'$ by inverting $G_I$ is $O(n^3)$ for a general matrix, if $n \gg m$, which is not trivial. As an example, for a mesh with a $40 \times 40$ grid (i.e., $m = 1600$) in the $x$–$y$ direction, and 6 grids in the $z$ direction, we have $n = 8000$.

The work in [23] generates the matrix $G$ by the column, observing that the $i$th column of $G$ is given by $G\mathbf{e_i}$, where $\mathbf{e_i}$ is a $m \times 1$ vector that is zero in all positions except the $i$th position, which is 1. From Equation (2.17), we have

$$G\mathbf{e_i} = G_P\mathbf{e_i} - G_C^T G_I^{-1} G_C \mathbf{e_i}. \tag{2.18}$$

The right-hand side is calculated in a two-step manner:

(1) Let us consider the second term of Equation (2.18). To find $q = G_I^{-1} G_C \mathbf{e_i}$, we must solve $G_I q = G_C \mathbf{e_i}$; note that $G_C \mathbf{e_i}$ is simply the $i$th column of $G_C$. Since $G_I$ is a sparse positive definite matrix, this set of equations can be solved using the Conjugate Gradient Method with incomplete Cholesky preconditioning [47]. Practically this type of iterative method converges within a constant number of iterations, and therefore, the cost for this step is $O(n)$.
(2) Next, the entire right-hand side of Equation (2.18) is computed by calculating $G_P \mathbf{e_i} + G_C^T q$. The cost for this step is $O(m)$, since $G_C^T$ is an $m \times n$ sparse matrix and $G_P \mathbf{e_i}$ is the $i$th column of $G_P$.

The process is repeated for all $\mathbf{e_i}$, $1 \leq i \leq m$ to find all of the columns of $G$, so that the overall complexity is $O(mn + m^2)$. If $n \gg m$, as is typical, the complexity is $O(mn)$.

### 2.3.2 The Finite Element Method

The FEM provides another avenue to solve Poisson's equation described by Equation (2.4). While it is a generic, classical, and widely used technique for solving such PDEs, it is possible to use the properties of the on-chip problem, outlined in the introduction to this survey, to compute the solution efficiently [49]. Before describing the FEM in detail, let us first take another look at the boundary conditions in Equations (2.5)–(2.7). It can be seen that all three boundary conditions can be cast into the form:

$$\text{Inward heat flow} = k_t \frac{\partial T(\mathbf{r})}{\partial n} = \alpha T(\mathbf{r}) + \beta(\mathbf{r}) \tag{2.19}$$

under the steady-state condition. For the isothermal condition, $\alpha = 0$, and $\beta(\mathbf{r})$ is unknown. For the heat flux condition, $\alpha = 0$, and $\beta(\mathbf{r}) = p(\mathbf{r})$. For the convective condition, $\alpha = -h$, and $\beta(\mathbf{r}) = hT_a$. Therefore, we will only focus on the boundary condition (2.19) in the FEM.

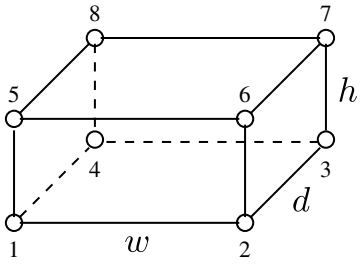



Fig. 2.3 An 8-node hexahedral element used in the FEM.

A succinct explanation of FEM, as applied to the on-chip case, is provided in [48]. In finite element analysis, the design space is first discretized or meshed into elements. Different element shapes can be used such as tetrahedra and hexahedra. For the on-chip problem, where all heat sources are modeled as being rectangular, a reasonable discretization for the FEM [49] divides the chip into 8-node rectangular hexahedral elements as shown in Figure 2.3. In the on-chip context, hexahedral elements also simplify the book-keeping and data management during FEM. The temperatures at the nodes of the elements constitute the unknowns that are computed during finite element analysis, and the temperature within an element is calculated using an interpolation function that approximates the solution to the heat equation within the elements, as shown below:

$$T(x,y,z) = \sum_{i=1}^{8} N_i(x,y,z)T_i, \tag{2.20}$$

where $N_i(x,y,z)$ is the shape function associated with node $i$ and $T_i$ is the temperature at node $i$. Let $(x_c, y_c, z_c)$ be the center of the element, and denote the width, depth, and height of the element by $w$, $d$, and $h$, respectively. The temperature at any point within the element is interpolated in FEM using a shape function, $N_i(x,y,z)$, which in this case is written as the trilinear function:

$$N_i(x,y,z) = \left(\frac{1}{2} + \frac{2(x_i - x_c)}{w^2}(x - x_c)\right) \times \left(\frac{1}{2} + \frac{2(y_i - y_c)}{d^2}(y - y_c)\right) \times \left(\frac{1}{2} + \frac{2(z_i - z_c)}{h^2}(z - z_c)\right). \tag{2.21}$$

The property of this function is that its value is 1 at vertex $i$, and zero at all other vertices, which satisfies the elementary requirement corresponding to a vertex temperature, as calculated in Equation (2.20).

From the shape functions, the thermal gradient, $\mathbf{g}$, can be found, using Equation (2.20), as follows:

$$\mathbf{g} = \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} = B\mathbf{T}, \tag{2.22}$$

$$\text{where } B = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots & \frac{\partial N_8}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \cdots & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \cdots & \frac{\partial N_8}{\partial z} \end{bmatrix} \tag{2.23}$$

As in the case of circuit simulation using the modified nodal formulation [28], stamps are created for each element and added to the global system of equations, given by:

$$K_g\mathbf{T} = \mathbf{P}, \tag{2.24}$$

where $T$ is the vector of all the nodal temperatures. This system of equations is typically sparse and can be solved efficiently.

In FEA, these stamps are called element stiffness matrices, $K$, and their values can be determined using techniques based on the calculus of variations. While a complete derivation of this theory is beyond the scope of this survey, and can be found in a standard text on FEM (such as [86]), it suffices to note that the end result yields the following stamps. For the case where only heat conduction comes into play, we have

$$K = \int_V B^T DB dV, \tag{2.25}$$

where $V$ is the volume of the element, and $D = \text{diag}(k_{t,x}, k_{t,y}, k_{t,z})$ is a $3 \times 3$ diagonal matrix in which $k_{t,i}, i \in \{x, y, z\}$ represents the thermal conductivity in each of the three coordinate directions, for the case where the region is anisotropic along the three coordinate directions; in many cases, $k_{t,x} = k_{t,y} = k_{t,z} = k_t$.

For the convective case, if a surface $S$ of the element participates in convective heat transfer, then the corresponding element stamp is given by

$$K = \int_S h_c N^T N dS, \tag{2.26}$$

where $h_c$ is the effective heat transfer coefficient for convection, as described in Equation (2.7). Note that the dimension of $K$ in this case corresponds to the number of nodes on one surface of the element.

For our hexahedral element, the stamp for the conductive case is given by the $8 \times 8$ symmetric matrix whose entries depend only on $w$, $h$, and $d$, and is given by

$$K = \begin{bmatrix} A & B & C & D & E & F & G & H \\ B & A & D & C & F & E & H & G \\ C & D & A & B & G & H & E & F \\ D & C & B & A & H & G & F & E \\ E & F & G & H & A & B & C & D \\ F & E & H & G & B & A & D & C \\ G & H & E & F & C & D & A & B \\ H & G & F & E & D & C & B & A \end{bmatrix}, \tag{2.27}$$

where

$$\begin{aligned} A &= \frac{k_{t,x}hd}{9w} + \frac{k_{t,y}wd}{9h} + \frac{k_{t,z}wh}{9d} \\ B &= -\frac{k_{t,x}hd}{9w} + \frac{k_{t,y}wd}{18h} + \frac{k_{t,z}wh}{18d} \\ C &= -\frac{k_{t,x}hd}{18w} - \frac{k_{t,y}wd}{18h} + \frac{k_{t,z}wh}{36d} \\ D &= \frac{k_{t,x}hd}{18w} - \frac{k_{t,y}wd}{9h} + \frac{k_{t,z}wh}{18d} \\ E &= \frac{k_{t,x}hd}{18w} + \frac{k_{t,y}wd}{18h} - \frac{k_{t,z}wh}{9d} \\ F &= -\frac{k_{t,x}hd}{18w} + \frac{k_{t,y}wd}{36h} - \frac{k_{t,z}wh}{18d} \end{aligned}$$

$$G = -\frac{k_{t,x}hd}{36w} - \frac{k_{t,y}wd}{36h} - \frac{k_{t,z}wh}{36d}$$
$$H = \frac{k_{t,x}hd}{36w} - \frac{k_{t,y}wd}{18h} - \frac{k_{t,z}wh}{18d}.$$

For the convective case, if the surface containing nodes 1, 2, 3, and 4 of Figure 2.3 is exposed to convective boundary conditions, then the stamp is given by:

$$K = \frac{h_c wh}{36}\begin{bmatrix} 4 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 \\ 2 & 1 & 2 & 4 \end{bmatrix}, \tag{2.28}$$

and $h_c whT_a/4$ would be added to the stamp of the right-hand side power vector at each location corresponding to these four nodes.

The stamps from various elements, including separate conductive and convective stamps, if applicable, and the power dissipation vector may now be superposed to obtain the global stiffness matrix. The entire mesh consists of these hexahedral elements aligned in a grid, with each node being shared by at most eight different elements. The element stiffness matrices are stamped into a global stiffness matrix, $K_g$, by adding together the components of the element matrices corresponding to the same node. Each entry of the global power vector, $P$, contains power dissipated or heat generation as represented at the corresponding node, as well as possible additions from the convective element.

All of these stamps are incorporated into the global set of Equations, (2.24). In case of isothermal boundary conditions, or if a node is connected to the ambient, the corresponding temperature is set to the ambient. The number of equations and variables can be correspondingly reduced. For example, if $\mathbf{T}_1$ is the vector of unknown temperatures, and all nodes in the subvector $\mathbf{T}_2$ are connected to fixed temperatures, then the global stiffness matrix can be written in the form:

$$\begin{bmatrix} K_{g,11} & K_{g,12} \\ K_{g,21} & K_{g,22} \end{bmatrix}\begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix}. \tag{2.29}$$

The fixed values in $\mathbf{T}_2$ can be moved to the right-hand side to obtain the reduced set of equations

$$K_{g,11}\mathbf{T}_1 = \mathbf{P}_1 - K_{g,12}\mathbf{T}_2. \tag{2.30}$$

### 2.3.3 Green Function-based Analysis

An alternative to discretization-based methods uses the notion of Green functions to perform steady-state thermal analysis. The Green function $G(\mathbf{r}, \mathbf{r}')$ is defined as the temperature distribution at the point with coordinate $\mathbf{r}$ within the chip, when a unit point power source is placed at location $\mathbf{r}'$. The calculation of the Green function proceeds by solving Poisson's equation under a unit impulse excitation at $\mathbf{r}'$, under a similar set of boundary conditions as the original problem, except that the ambient temperature, $T_a$, is taken as the reference. If this function can be determined, the temperature distribution within the chip under an arbitrary power density distribution can be calculated, using the principle of superposition, as

$$T(\mathbf{r}) = T_a + \int_V G(\mathbf{r}, \mathbf{r}')g(\mathbf{r}')dV, \tag{2.31}$$

where $g(\mathbf{r}')$ is the volume power density distribution, and $V$ is the volume of the chip.

Green function-based methods have been used for thermal analysis in several works in the literature [70, 148, 149, 150, 166].

We focus here on the work presented in [166]. This begins with the notion of Green functions, and introduces a number of efficiency enhancements that exploit the properties of the on-chip thermal analysis problem. One such property is that the primary on-chip heat sources are the devices, and the points at which the temperature is to be computed correspond to locations in the device layer or one of the interconnect layers. In other words, the points where temperatures must be calculated are typically on a set of discrete planes, which will be referred to as the field planes, and the power sources also lie on a set of discrete planes, which we call the source planes. Therefore, in the development of Green function-based thermal analysis algorithms, it is often sufficient to focus on a pair of source and field planes (which

are permitted to be identical to each other). When multiple source and field planes are present, the temperature distribution can be obtained easily through superposition, since the underlying PDE is linear.

The ensuing discussion assumes that the dimension of the chip in the $x$ and $y$ directions is $a$ and $b$, respectively. For on-chip applications, it is reasonable to assume that the source and field planes are parallel to each other, and that they correspond to different values of the $z$ coordinate, say, $z$ and $z'$, respectively. Given a point $(x, y)$ on the field plane, and a point $(x', y')$ on the source plane, the Green function can be written in the following form:

$$G(x, y, x', y') = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} C_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \times \cos\left(\frac{m\pi x'}{a}\right) \cos\left(\frac{n\pi y'}{b}\right). \quad (2.32)$$

This formulation assumes that adiabatic boundary conditions are applied on the sidewalls of the chip. The temperature distribution at a point on the field plane can then be calculated using the following expression, in accordance with Equation (2.31):

$$T(x, y) = T_a + \int_{x\text{-range}} dx' \int_{y\text{-range}} dy' G(x, y, x', y') p(x', y'), \quad (2.33)$$

where $p(x', y')$ is the area power density on the source plane.

The problem with calculating the temperature distribution on the field plane directly using Equation (2.33) is that the Green function (2.32) contains infinite summations, and in actual implementations of thermal analysis algorithms, the summations have to be truncated at high indices in order to ensure a reasonable accuracy, and this often leads to excessively long runtimes.

In [166], the authors developed several high efficiency Green function-based thermal analysis algorithms using ideas similar to those proposed by Ghurpurey and Meyer [46] and Costa et al. [36] in the context of substrate parasitic extraction, and these algorithms can be used to perform both localized analysis and full-chip temperature analysis. Three algorithms are presented in the work, and are outlined below: the first algorithm performs localized analysis, the second performs

full-chip analysis with equal levels of accuracy throughout the chip, and the third performs efficient full-chip analysis for the case where different regions of the chip may require different levels of accuracy. Each of these operates under simplifying assumptions: that the heat sources are in a plane at the top of the chip, and that the chip has layerwise uniform thermal conductivities. The first algorithm is suited for limited or incremental computations, the second to full-chip analysis, and the third to full-chip analysis where the required accuracy in different parts is different.

#### 2.3.3.1 Algorithm I: Localized Analysis

For localized analysis, the algorithm proceeds by first establishing a few look-up tables using the Green function coefficients, $C_{mn}$, in Equation (2.32). This step can be performed efficiently with the help of the discrete cosine transform (DCT) [102]. After the look-up tables have been established, the calculation of the temperature rise in a rectangular shaped field region due to the power generated in a rectangular shaped source region is reduced to a few table look-ups, which is significantly faster than truncating and evaluating the Green function directly and then using Equation (2.33) to calculate the temperature rise.

Since on-chip geometries can typically be decomposed into combinations of rectangles, we only focus on rectangular-shaped source and field regions in the following analysis. Figure 2.4 shows a schematic of a source and a field region. Note that the two regions can have different $z$ coordinates if the field plane does not coincide with the source plane.

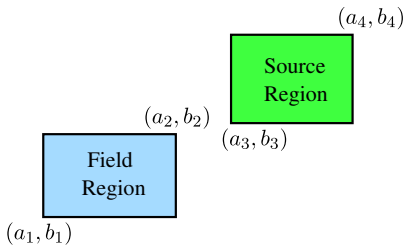



Fig. 2.4 Source and field regions for computing the temperature distribution.

Our objective here is to calculate the average temperature $\overline{T_f}$ of the field region efficiently given the power density $P_d$ of the source region. To simplify the analysis, it is assumed that $P_d$ is a constant within the source region. This is not a very restrictive assumption, since if the power density is not uniformly distributed in the source region, the source region may be divided into smaller rectangular-shaped sub-regions such that the power density is uniform within each sub-region.

The average temperature in the field region can be computed using

$$\overline{T_f} = \frac{1}{(a_2 - a_1)(b_2 - b_1)} \int_{a_1}^{a_2} dx \int_{b_1}^{b_2} T(x, y) dy. \tag{2.34}$$

Substituting Equations (2.32) and (2.33) into the above equation and performing some involved, but not difficult, algebra, it can be verified that the following equation is obtained:

$$\begin{aligned} \overline{T_f} &= T_a + \frac{P_d}{(a_2 - a_1)(b_2 - b_1)} \\ &\quad \times \int_{a_1}^{a_2} dx \int_{b_1}^{b_2} dy \int_{a_3}^{a_4} dx' \int_{b_3}^{b_4} dy' G(x, y, x', y') \\ &= T_a + \Sigma_0 + \Sigma_1 + \Sigma_2 + \Sigma_3, \end{aligned} \tag{2.35}$$

where

$$\Sigma_0 = C_{00} P_d (a_4 - a_3)(b_4 - b_3) \tag{2.36}$$

$$\begin{aligned} \Sigma_1 = &\left\{ \frac{P_d (b_4 - b_3)}{(a_2 - a_1)} \sum_{m=0}^{\infty} D_{m0} \left[ \sin\left(\frac{m\pi a_2}{a}\right) - \sin\left(\frac{m\pi a_1}{a}\right) \right] \right. \\ &\left. \times \left[ \sin\left(\frac{m\pi a_4}{a}\right) - \sin\left(\frac{m\pi a_3}{a}\right) \right] \right\} \end{aligned} \tag{2.37}$$

$$\begin{aligned} \Sigma_2 = &\left\{ \frac{P_d (a_4 - a_3)}{(b_2 - b_1)} \sum_{n=0}^{\infty} E_{0n} \left[ \sin\left(\frac{n\pi b_2}{b}\right) - \sin\left(\frac{n\pi b_1}{b}\right) \right] \right. \\ &\left. \times \left[ \sin\left(\frac{n\pi b_4}{b}\right) - \sin\left(\frac{n\pi b_3}{b}\right) \right] \right\} \end{aligned} \tag{2.38}$$

$$\Sigma_3 = \Bigg\{\frac{P_d}{(a_2-a_1)(b_2-b_1)}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}F_{mn}\left[\sin\left(\frac{m\pi a_2}{a}\right)-\sin\left(\frac{m\pi a_1}{a}\right)\right] \times\left[\sin\left(\frac{m\pi a_4}{a}\right)-\sin\left(\frac{m\pi a_3}{a}\right)\right]\left[\sin\left(\frac{n\pi b_2}{b}\right)-\sin\left(\frac{n\pi b_1}{b}\right)\right] \times\left[\sin\left(\frac{n\pi b_4}{b}\right)-\sin\left(\frac{n\pi b_3}{b}\right)\right]\Bigg\}. \tag{2.39}$$

$$D_{m0} = \begin{cases} C_{m0}\left(\frac{a}{m\pi}\right)^2 & \text{if } m \neq 0, \\ 0 & \text{if } m = 0. \end{cases} \tag{2.40}$$

$$E_{n0} = \begin{cases} C_{n0}\left(\frac{b}{n\pi}\right)^2 & \text{if } n \neq 0, \\ 0 & \text{if } n = 0. \end{cases} \tag{2.41}$$

$$F_{mn} = \begin{cases} C_{mn}\left(\frac{a}{m\pi}\right)^2\left(\frac{b}{n\pi}\right)^2 & \text{if } m \neq 0, \quad n \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.42}$$

Although the above expressions look rather involved, the key realization here is that there are a number of terms in $\Sigma_1$, $\Sigma_2$, and $\Sigma_3$ that are a product of two sines. To see how these can be mapped to a DCT, we begin with the standard trigonometric identity

$$\sin\theta_1 \sin\theta_2 = \frac{1}{2}(\cos(\theta_1 - \theta_2) - \cos(\theta_1 + \theta_2)).$$

$\Sigma_1$ can be rewritten as a sum of eight terms in the form:

$$\pm\frac{1}{2}\sum_{m=0}^{\infty} D_{m0}\cos\left(\frac{m\pi(a_i \pm a_j)}{a}\right), \tag{2.43}$$

where $i = 1, 2$ and $j = 3, 4$.

To utilize the DCT, the source and field planes are first discretized into $M$ equal divisions along the $x$ direction and $N$ equal divisions along the $y$ direction and form the grids (criteria for selecting $M$ and $N$ are detailed in [166]). Next, the summation in Equation (2.43) is truncated at index $M$; the indices $M$ and $N$ are determined by the considerations of both the resolution of thermal analysis and the convergence of the Green function. Assuming that all the vertices of the field and source

regions are located on grid points, i.e., $\frac{a_i}{a} = \frac{k_i}{M}$, $\frac{a_j}{a} = \frac{k_j}{M}$, where $k_i$ and $k_j$ are integers, and $0 \le k_i \le M$, $0 \le k_j \le M$, (2.43) can be rewritten as

$$\pm\frac{1}{2}\sum_{m=0}^{M} D_{m0}\cos\left(\frac{m\pi(k_i \pm k_j)}{M}\right). \tag{2.44}$$

Let

$$k = \begin{cases} k_i \pm k_j & \text{if } 0 \le k_i \pm k_j \le M \\ -(k_i \pm k_j) & \text{if } k_i \pm k_j < 0 \\ 2M - (k_i \pm k_j) & \text{if } k_i \pm k_j > M. \end{cases} \tag{2.45}$$

Then $0 \le k \le M$ and Equation (2.44) can be rewritten as

$$\pm\frac{1}{2}\sum_{m=0}^{M} D_{m0}\cos\left(\frac{m\pi k}{M}\right). \tag{2.46}$$

This is precisely one term in the type-I DCT of the sequence $D_{m0}$, and the DCT sequence can be computed efficiently using the fast Fourier transform (FFT) in $O(M \log M)$ time [102]. After the DCT sequence is obtained, it can be stored in a vector and used many times in future temperature calculations. As a result, the computation of $\Sigma_1$ is reduced to eight look-ups in the DCT vector in constant time and then adding up the look-up results. The summation $\Sigma_2$ in Equation (2.38) can be similarly computed in an efficient manner, using the DCT and table look-ups.

The double summation $\Sigma_3$ in Equation (2.37) can be rewritten as a sum of 64 terms in the form:

$$\pm\frac{1}{4}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} F_{mn}\cos\left(\frac{m\pi(a_i \pm a_j)}{a}\right)\cos\left(\frac{n\pi(b_p \pm b_q)}{b}\right), \tag{2.47}$$

where $i = 1, 2$, $j = 3, 4$, $p = 1, 2$, and $q = 3, 4$. Using a similar approach as before, $\Sigma_2$ can be cast into

$$\pm\frac{1}{4}\sum_{m=0}^{M}\sum_{n=0}^{N} F_{mn}\cos\left(\frac{m\pi k}{M}\right)\cos\left(\frac{n\pi l}{N}\right), \tag{2.48}$$

where $0 \le k \le M$ and $0 \le l \le N$. This is one term in the 2D type-I DCT of the matrix $F_{mn}$. The 2D DCT matrix can be computed

using the FFT in $O((M{\cdot}N) \times \log(M{\cdot}N))$ time, and after the 2D DCT table is obtained, the double summation reduces to 64 table look-ups in constant time and then adding up the look up results.

Note that when multiple heat sources are present, their effects on the average temperature rise above $T_a$ in the field region, i.e., the integral term in Equation (2.33), can be superposed to obtain the total average temperature rise.

#### 2.3.3.2 Algorithm II: Full-Chip Thermal Simulation Using Spectral Domain Computations

Algorithm II for full-chip analysis is based by noting that Equation (2.33) is essentially the convolution of the Green function and the power density distribution, which can be performed efficiently in the spectral domain. The algorithm proceeds as follows:

- It first calculates the spectral response of a linear system determined by the Green function using the coefficients $C_{mn}$'s in Equation (2.32).
- Next, it obtains the spectral domain representation of the power density distribution with the help of the DCT and computes the spectral domain representation of the temperature distribution through the point-wise multiplications of the spectral components of the power density and the corresponding spectral response of the linear system.
- Finally, the space domain temperature distribution is calculated using the inverse DCT of its spectral domain representation. The DCT operations can be achieved in $O(n \log n)$ time, where $n$ is the number of grid cells on the source and field planes, which ensures the efficiency of the thermal analysis algorithms.

An advantage of the method is that the spectral responses of the linear system determined by the Green function only depend on the chip geometry and material properties, and are independent of the placement and power dissipation of modules. Therefore, they can be calculated once and used many times in thermally aware physical design, as

the spatial configuration of the design is iteratively altered to achieve an optimal layout.

The first step of the algorithm is to obtain the spectral domain representation of the power density map in the form:

$$P_d(x',y') = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{ij}\phi_{ij}(x',y'), \tag{2.49}$$

$$\text{where } \phi_{ij}(x,y) = \cos\left(\frac{i\pi x}{a}\right)\cos\left(\frac{j\pi y}{b}\right), \tag{2.50}$$

where $a$ and $b$ are, respectively, the dimension of the chip in the $x$ and $y$ direction.

Using simple algebra, it is easy to verify that $\phi_{ij}(x,y)$ satisfies the equation

$$\lambda_{ij}\phi_{ij}(x,y) = \int_0^a dx' \int_0^b dy' G(x,y,x',y')\phi_{ij}(x',y'), \tag{2.51}$$

where

$$\lambda_{ij} = \begin{cases} abC_{ij} & \text{if } i=j=0 \\ \frac{1}{2}abC_{ij} & \text{if } i=0,\ j\neq 0 \quad \text{or } i\neq 0,\ j=0 \\ \frac{1}{4}abC_{ij} & \text{if } i\neq 0,\ j\neq 0. \end{cases} \tag{2.52}$$

Here, $\lambda_{ij}$ is the response of the linear system to the spectral component $\phi_{ij}(x,y)$ [36]. After the spectral domain representation of the power density distribution in the source plane is obtained, the temperature distribution in the field plane can be calculated easily by

$$T(x,y) = T_a + \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \lambda_{ij}a_{ij}\phi_{ij}(x,y). \tag{2.53}$$

As will be shown next, both the spectral decomposition in Equation (2.49) and the double-summation in Equation (2.53) can be calculated efficiently using the DCT and IDCT through the FFT.

Next, assume that the source plane is divided into $M_s \times N_s$ rectangular grid cells of equal size as shown in Figure 2.5; a criterion for selecting $M_s$ and $N_s$ based on a power density criterion is described in [166]. The power density in each grid cell on the source plane is assumed

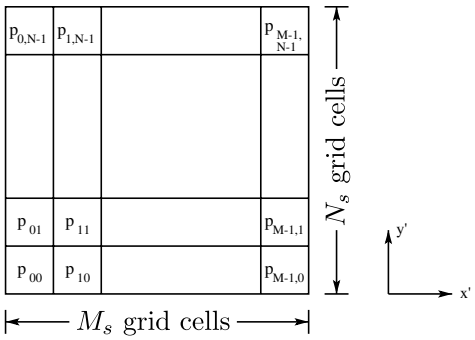



Fig. 2.5 The arrangement of the $M_s \times N_s$ grid cells on the source plane.

to be uniform, i.e., the power density distribution can be written in the piecewise constant form:

$$P_d(x', y') = \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn} \Theta \left( x' - \left( m + \frac{1}{2} \right) \Delta x_s, y' - \left( n + \frac{1}{2} \right) \Delta y_s \right) \tag{2.54}$$

where

$$\Theta(x', y') = \begin{cases} 1 & \text{if } |x'| \leq \frac{1}{2}\Delta x_s \quad \text{and} \quad |y'| \leq \frac{1}{2}\Delta y_s \\ 0 & \text{otherwise} \end{cases} \tag{2.55}$$

$\Delta x_s = \frac{a}{M_s}, \Delta y_s = \frac{b}{N_s}$, and $P_{mn}$ is the power density of the $mn$th grid cell.

The substitution of Equation (2.54) into Equation (2.49) and the use of the orthogonality property of the cosine functions in the integral sense yields

$$a_{ij} = A_{ij} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn} \cos \left( \frac{i\pi(2m+1)}{2M_s} \right) \cos \left( \frac{j\pi(2n+1)}{2N_s} \right), \tag{2.56}$$

where

$$A_{ij} = \begin{cases} \frac{1}{M_s N_s} & \text{if } i = j = 0 \\ \frac{4}{iN_s\pi} \sin \left( \frac{i\pi}{2M_s} \right) & \text{if } i \neq 0,\ j = 0 \\ \frac{4}{M_s j\pi} \sin \left( \frac{j\pi}{2N_s} \right) & \text{if } i = 0,\ j \neq 0 \\ \frac{16}{ij\pi^2} \sin \left( \frac{i\pi}{2M_s} \right) \sin \left( \frac{j\pi}{2N_s} \right) & \text{if } i \neq 0,\ j \neq 0. \end{cases} \tag{2.57}$$

As before, we see that for $0 \le i < M_s$ and $0 \le j < N_s$, the double summation in Equation (2.56) can be considered as a term in the 2D type-II DCT [102] of the power density matrix $P$. For $i \ge M_s$ or $j \ge N_s$, we can always find integers $s_1$ and $s_2$ such that $i = 2s_1M_s \pm \hat{i}$ and $j = 2s_2N_s \pm \hat{j}$, where $0 \le \hat{i} < M_s$ and $0 \le \hat{j} < N_s$.[1] Hence, for any $i$ and $j$, we always have

$$a_{ij} = \pm A_{ij}\widetilde{P}_{\hat{i}\hat{j}}, \tag{2.58}$$

where

$$\widetilde{P}_{\hat{i}\hat{j}} = \sum_{m=0}^{M_s-1}\sum_{n=0}^{N_s-1} P_{mn}\cos\left(\frac{\hat{i}\pi(2m+1)}{2M_s}\right)\cos\left(\frac{\hat{j}\pi(2n+1)}{2N_s}\right) \tag{2.59}$$

with $0 \le \hat{i} < M_s$ and $0 \le \hat{j} < N_s$ is the 2D type-II DCT of the $P$ matrix and the sign of the right-hand side of Equation (2.58) is determined by whether $s_1$ and $s_2$ are even or odd numbers [36]. Equation (2.59) can be calculated efficiently using the 2D FFT in $O((M_s{\cdot}N_s){\times}\log(M_s{\cdot}N_s))$ time. After the 2D DCT matrix $\widetilde{P}$ is obtained, the calculation of $a_{ij}$ simply involves computing the coefficient $A_{ij}$ and finding the corresponding term $\widetilde{P}_{\hat{i}\hat{j}}$.

From Equations (2.50) and (2.53), under the discretization, the temperature distribution $T(x,y)$ can now be written as

$$T(x,y) = T_a + \sum_{i=0}^{M'-1}\sum_{j=0}^{N'-1}\lambda_{ij}a_{ij}\cos\left(\frac{i\pi x}{a}\right)\cos\left(\frac{j\pi y}{b}\right). \tag{2.60}$$

If we assume that the temperature field plane is divided into $M_f{\times}N_f$ rectangular grid cells of equal size, then the average temperature of the $mn$th grid cell can be obtained by

$$\begin{aligned} T_{mn} &= \frac{1}{\Delta x_f \Delta y_f}\int_{m\Delta x_f}^{(m+1)\Delta x_f} dx \int_{n\Delta y_f}^{(n+1)\Delta y_f} dy T(x,y) \\ &= T_a + \sum_{i=0}^{M'-1}\sum_{j=0}^{N'-1} B_{ij}\cos\left(\frac{i\pi(2m+1)}{2M_f}\right)\cos\left(\frac{j\pi(2n+1)}{2N_f}\right), \end{aligned} \tag{2.61}$$

[1] If $i$ equals an odd multiple of $M_s$, we will not be able to write $i$ as $i = 2s_1M_s \pm \hat{i}$. However, for this kind of $i$, it can be easily shown that $a_{ij} = 0$ because $\cos\left(\frac{i\pi(2m+1)}{2M_s}\right) = 0$. Similarly, we know that $a_{ij} = 0$ if $j$ equals an odd multiple of $N_s$.

where $\Delta x_f = \frac{a}{M_f}$, $\Delta y_f = \frac{b}{N_f}$, and

$$B_{ij} = \begin{cases} \lambda_{ij} a_{ij} & \text{if } i = j = 0 \\ 2\lambda_{ij} a_{ij} \frac{M_f}{i\pi} \sin\left(\frac{i\pi}{2M_f}\right) & \text{if } i \neq 0,\ j = 0 \\ 2\lambda_{ij} a_{ij} \frac{N_f}{j\pi} \sin\left(\frac{j\pi}{2N_f}\right) & \text{if } i = 0,\ j \neq 0 \\ 4\lambda_{ij} a_{ij} \frac{M_f N_f}{ij\pi^2} \sin\left(\frac{i\pi}{2M_f}\right) \sin\left(\frac{j\pi}{2N_f}\right) & \text{if } i \neq 0,\ j \neq 0. \end{cases} \tag{2.62}$$

Similar to the analysis shown previously, any $i \geq M_f$ and $j \geq N_f$ can be written as $i = 2s_3 M_f \pm \hat{\hat{i}}$ and $j = 2s_4 N_f \pm \hat{\hat{j}}$ such that $0 \leq \hat{\hat{i}} < M_f$, $0 \leq \hat{\hat{j}} < N_f$, and $s_3$ and $s_4$ are integers. Using the periodicity of the cosine function, we can finally cast $T_{mn}$ into the form:

$$T_{mn} = T_a + \sum_{\hat{\hat{i}}=0}^{M_f-1} \sum_{\hat{\hat{j}}=0}^{N_f-1} L_{\hat{\hat{i}}\hat{\hat{j}}} \cos\left(\frac{\hat{\hat{i}}\pi(2m+1)}{2M_f}\right) \cos\left(\frac{\hat{\hat{j}}\pi(2n+1)}{2N_f}\right), \tag{2.63}$$

where

$$L_{\hat{\hat{i}}\hat{\hat{j}}} = \begin{cases} B_{00} & \text{if } \hat{\hat{i}} = \hat{\hat{j}} = 0 \\ \sum\limits_{\substack{i < M' \\ i = 2s_3 M_f \pm \hat{\hat{i}}}} \pm\, B_{i0} & \text{if } \hat{\hat{i}} \neq 0,\ \hat{\hat{j}} = 0 \\ \sum\limits_{\substack{j < N' \\ j = 2s_4 N_f \pm \hat{\hat{j}}}} \pm\, B_{0j} & \text{if } \hat{\hat{i}} = 0,\ \hat{\hat{j}} \neq 0 \\ \sum\limits_{\substack{i < M' \\ i = 2s_3 M_f \pm \hat{\hat{i}}}} \sum\limits_{\substack{j < N' \\ j = 2s_4 N_f \pm \hat{\hat{j}}}} \pm\, B_{ij} & \text{if } \hat{\hat{i}} \neq 0,\ \hat{\hat{j}} \neq 0 \end{cases} \tag{2.64}$$

and the signs of the $B's$ in Equation (2.64) are determined by whether $s_3$ and $s_4$ are even or odd numbers. After the matrix $L$ is obtained, the double summation in Equation (2.63) can be calculated efficiently using the 2D IDCT.

Input:

- Chip geometry and physical properties of the material layers.
- Power density map − matrix $P$.

Output: Temperature distribution map − matrix $T$.
Algorithm:

(1) Calculate the Green function coefficients $C_{ij}{}'s$;
(2) Calculate the spectral responses of the system $\lambda_{ij}{}'s$;
(3) Calculate the type-II 2D DCT of the power density matrix $\widetilde{P} = \text{2DDCT}(P)$;
(4) $TSE = \frac{1}{M_s N_s} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn}^2$;
(5) $M' = M_s,\ N' = N_s$;
$ASE = \sum_{i=0}^{M'-1} \sum_{j=0}^{N'-1} s_{ij} a_{ij}^2$;
*while* $(ASE < \eta \times TSE)$
$M' = M' + M_s,\ N' = N' + N_s$;
Update $ASE$;
*end while*;
(6) Calculate the matrix $L$;
(7) Calculate the temperature distribution map using the type-II 2D IDCT $T = T_a + \text{2DIDCT}(L)$;

Fig. 2.6 Thermal simulation algorithm using the Green function method, the DCT, and the spectral domain computations.

The complete thermal simulation algorithm using the Green function method, the DCT, and the spectral domain computations is shown in Figure 2.6. The asymptotic time complexity of the algorithm is $O(\mathcal{N}_{gs} \times \log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf} \times \log(\mathcal{N}_{gf}))$, where $\mathcal{N}_{gs} = M_s \cdot N_s$ is the total number of grid cells in the power density map, and $\mathcal{N}_{gf} = M_f \cdot N_f$ is the total number of grid cells in the resulting temperature profile. This is a significant improvement over the $O(\mathcal{N}_{gs} \cdot \mathcal{N}_{gf})$ complexity of Algorithm I for full-chip thermal simulations.

#### 2.3.3.3 Algorithm III: Thermal Simulation with Local High Accuracy Requirements

In some cases, a design may have different requirements on the accuracy of the thermal simulation over different parts of the same chip. For example, in mixed signal designs where analog circuits are fabricated on the same chip as digital circuits, the analog blocks often have more stringent accuracy requirements on the thermal simulation because the operations of the analog circuits are more sensitive to temperature. For full-chip analysis, Algorithm I will be too slow, and Algorithm II will constrain the size of each grid cell to be small enough to satisfy the highest accuracy requirements, resulting in wasted computation.

The key idea of Algorithm III is to use coarse grids to divide the source and field planes, so that the size of each grid cell in the field plane satisfies the accuracy requirements of the digital circuits. A temperature analysis is performed using Algorithm II to perform thermal analysis at a level of accuracy that is sufficient for most blocks (e.g., for all the digital blocks in a design). Finally, for each region or unit on the field plane whose temperature is to be calculated more accurately (e.g., the analog blocks), we use Algorithm I to compute the contributions to its temperature rise from the nearby logic gates and analog function units on the source plane, and use this result to correct the temperature obtained by Algorithm II over the coarse grid cell.

### 2.3.4 Comparison Between Steady-State Thermal Analysis Algorithms

We have presented three different classes of steady-state thermal analysis algorithms, and they each have their own advantages and disadvantages. The FDM and FEM are more generic, and they can handle complicated on-chip geometries such as nonuniform wiring structures. Therefore, they can achieve very high accuracy in thermal analysis. However, the direct application of these two methods usually involves meshing the entire substrate, which may lead to large problem sizes and relatively long runtimes. Using the macromodeling techniques such as that presented in [139], it is possible to abstract away the nodes that the user of the thermal simulator is not interested in, and therefore,

reduce the problem sizes in the finite difference and finite element analysis. However, building a macromodel still involves considerable effort. Therefore, the macromodeling approach is most effective under the situation where the chip geometry does not change but the thermal analysis needs to be performed multiple times, such as in the fixed-die thermal aware floorplanning and placement, because the time it takes to build the macromodel can be amortized.

In Green function-based methods, only the layers where the temperature distribution is to be calculated and the layers where the power is generated are meshed. Therefore, the resulting problem size is relatively small and the efficiency of thermal analysis is rather high. However, in a Green function-based thermal analysis, it is often assumed that the chip materials are layer-wise uniform, which may be too restrictive. As a result, these algorithms are usually used in early stages of physical design, where the accuracy requirement on thermal analysis is moderate but the efficiency requirement is high.

## 2.4 Solving the Linear Equations

The FEM and FDM methods both lead to problem formulations that require the solution of large systems of linear equations. The matrices that describe these equations are typically sparse (more so for the FDM than the FEM, as can be seen from the individual element stamps) and positive definite.

There are many different ways of solving these equations. Direct methods typically use variants of Gaussian elimination, such as LU factorization, to first factor the matrices, and then solve the system through forward and backward substitution. The cost of LU factorization is $O(n^3)$ for a dense $n \times n$ matrix, but is just slightly superlinear in practice for sparse systems. This step is followed by forward/backward substitution, which can be performed in $O(n)$ time for a sparse system where the number of entries per row is bounded by a constant. If a system is to be evaluated for a large number of right-hand side vectors, corresponding to different power vectors, LU factorization only needs to be performed once and its cost may be amortized over the solution for multiple input vectors.

Iterative methods are seen to be very effective for large sparse positive definite matrices. This class of techniques includes more classical methods such as Gauss–Jacobi, Gauss–Seidel, and successive overrelaxation, as well as more contemporary approaches based on the conjugate gradient method or GMRES. The idea here is to begin with an initial guess solution, and to successively refine it to achieve convergence. Under certain circumstances, it is possible to guarantee this convergence: in particular, FDM matrices have a structure that guarantees this property. For further details on standard techniques used in direct and iterative solvers, the reader is referred to a standard text on the topic, such as [47].

In this section, we will focus on describing two methods that are especially useful in solving thermal systems, namely, the multigrid approach and the random walk method. Our exposition will describe both of these in the context of the FDM. It should be noted that these methods are also useful in solving transient analysis problems under time-stepping, as outlined in Section 2.5.

### 2.4.1 The Multigrid Method

The multigrid algorithm was successfully used in [83, 84] to solve the thermal analysis problem using the FDM matrices. The multigrid method follows a hierarchical approach to solve the thermal problem and has been used in the solution of a number of areas where the underlying problem is described by a PDE or is equivalent to its discretization (e.g., in the solution of on-chip power grids). An excellent tutorial on various aspects of the multigrid method can be found in [15]. The essence of the approach is based on the observation that an iterative solver is usually more effective in removing high frequency solution errors in an FDM mesh than low frequency errors. Therefore, the method constructs a hierarchy of FDM meshes corresponding to the thermal problem, with each lower level mesh being coarser than the adjacent higher level mesh in the hierarchy. The solver starts with iterating over the finest mesh, and once it detects that the speed toward convergence is slowed down due to low frequency solution errors, the iteration is changed to the coarser grid that is

one level below, since solution errors appear to have higher frequencies in a coarser grid than in a finer grid. Once the corrections to the solution is obtained for the coarser grid, it is mapped back to the finer grid to generate the final solution. This procedure is often called the V-Cycle in the literature. The overall runtime of the algorithm is observed to be linear in terms of the number of nodes in the finest FDM mesh.

An outline of the algorithm used in [84] is shown in Figure 2.7. The three key operations in a multigrid method are:

- *Smoothing*: This operation, referred to as *smooth* in Figure 2.7, is carried out by simultaneously updating the

Multigrid cycle $u_k = MG(k, u_k^0, A_k, b_k, m, n_1, n_2)$

(1) Presmoothing: Iterate $n_1$ times on the $k$th grid to compute an approximate solution $v_k$ with initial guess $u_k^0$, i.e., $v_k = smooth^{n_1}(u_k^0, A_k, b_k)$.
(2) Coarse grid correction:
  (a) Compute the residual $r_k = b_k - A_k v_k$.
  (b) Approximate the residual on the $(k+1)$th grid using a restriction operator $r_{k+1} = R_k^{k+1} r_k$.
  (c) Solve the problem in the coarse grid $A_{k+1} e_{k+1} = r_{k+1}$. If the coarsest level has been reached, solve the equations directly, otherwise, apply the multigrid cycle $m$ times with a zero initial guess at the entrance to the first multigrid cycle, i.e., $e_{k+1} = MG^m(k+1, 0, A_{k+1}, r_{k+1}, m, n_1, n_2)$.
  (d) Map the correction to the $k$th grid by interpolation and correct the solution $u_k$, i.e., $e_k = P_{k+1}^k e_{k+1}$ and $u_k = v_k + e_k$.
(3) Post-smoothing: Iterate $n_2$ times with initial guess $u_k$, i.e., $u_k = smooth^{n_2}(u_k, A_k, b_k)$.

Fig. 2.7 The multigrid algorithm.

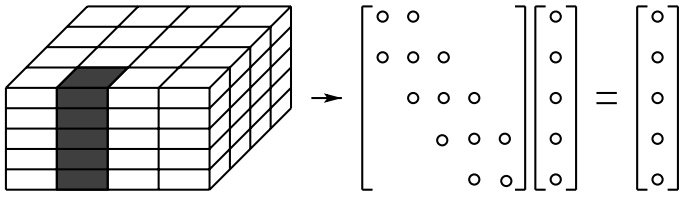


Fig. 2.8 One step in a smoothing iteration which involves updating the temperature corresponding to one column of the FDM cells.

values of the unknowns corresponding to a vertical column of nodes in the FDM mesh, as shown in Figure 2.8, in each step of a smoothing iteration. A conventional smoothing operation could use a few Gauss–Seidel iterations, for example, but this approach takes advantage of the fact that heat flow in the $z$ direction dominates flow in the $x$ or $y$ dimensions under typical discretization schemes. The modified smoothing operation involves the solution of a tridiagonal linear system (as each node is linked to the node above and below it), which has a linear runtime in terms of the dimension of the system.

- *Restriction*: The restriction operator, $R^{k+1}$, maps a finer grid at level $k$ to a coarser grid at level $k+1$. Mathematically, this involves finding the residual $r_{k+1}$ in the $(k+1)$th grid given the residual $r_k$ in the $k$th grid, and in [84], the component of the residual at a particular node in the $(k+1)$th grid is obtained by taking the weighted average of the residual components corresponding to the same node and its neighboring nodes in the $k$th grid, where the weight of each point is the absolute value of its coefficient in the finite difference equations corresponding to the $k$th grid.
- *Interpolation*: The role of the interpolation operator, $P^{k+1}$, is to map a coarser grid at level $k+1$ to a finer grid at level $k$. It finds the error correction term $e_k$ in the fine grid, given the correction term $e_{k+1}$ in the coarse grid, and in [84], this

is achieved by setting

$$e^k_{p,q,r} = \frac{1}{A}\big(a_{p-1,q,r}e^k_{p-1,q,r} + a_{p+1,q,r}e^k_{p+1,q,r} + a_{p,q-1,r}e^k_{p,q-1,r} + a_{p,q+1,r}e^k_{p,q+1,r} + a_{p,q,r-1}e^k_{p,q,r-1} + a_{p,q,r+1}e^k_{p,q,r+1}\big), \tag{2.65}$$

where $e^k_{p,q,r}$ is the component of $e_k$ that corresponds to the $(p,q,r)$th node in the FDM mesh, $a_{p,q,r}$ is the coefficient of the FDM equations that is determined by the material properties of the chip and the discretization, and $A = a_{p-1,q,r} + a_{p+1,q,r} + a_{p,q-1,r} + a_{p,q+1,r} + a_{p,q,r-1} + a_{p,q,r+1}$ Here, $p$, $q$, and $r$ are the indices of the node in the $x$, $y$, and $z$ directions, respectively. Note that the weighted averaging using the $a_{i,j,k}$ values, rather than a simple averaging, serve to capture effects related to anisotropies in thermal conductivity.

### 2.4.2 The Random Walk Method

The random walk method has been used in the literature to solve the on-chip power grid analysis problem, which is structurally identical to the thermal analysis problem using finite differences. It involves the solution of a network of resistors, constant current sources, and constant voltage sources, and computes the voltage throughout this circuit, which corresponds to temperature under the thermal–electrical analogy, defined in Section 2.3.1. The random walk method has been used successfully in the solution of such large networks. These techniques perform very well when the temperatures at one node, or a small number of nodes, must be calculated. Therefore, a major benefit of these methods is in their ability to perform incremental analysis rapidly and efficiently, and they are good for computing the effects of a small design change that requires temperature changes in only a small region of the chip.

To outline the method, we will consider the solution of a resistive network for the voltages; the thermal analog, of course, is that the voltages in the resistive network are the temperatures, and the current sources are the power values.

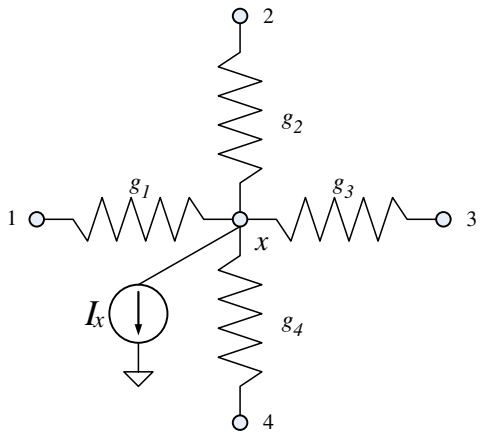



Fig. 2.9 A representative node in the resistive network.

For the DC analysis of a resistive network with constant current and voltage sources, let us look at a single node $x$ in the circuit, as illustrated in Figure 2.9. The application of Kirchoff's Current Law, Kirchoff's Voltage Law, and the device equations for the conductances, yields the following equation:

$$\sum_{i=1}^{\text{degree}(x)} g_i(V_i - V_x) = I_x, \tag{2.66}$$

where the nodes adjacent to $x$ are labeled $1, 2, \ldots,$ degree$(x)$, $V_x$ is the voltage at node $x$, $V_i$ is the voltage at node $i$, $g_i$ is the conductance between node $i$ and node $x$, and $I_x$ is the current load connected to node $x$. Equation (2.66) can be reformulated as follows:

$$V_x = \sum_{i=1}^{\text{degree}(x)} \frac{g_i}{\sum_{j=1}^{\text{degree}(x)} g_j} V_i - \frac{I_x}{\sum_{j=1}^{\text{degree}(x)} g_j}. \tag{2.67}$$

This implies that the voltage at any node is a linear function of the voltages at its neighbors. We also observe that the sum of the linear coefficients associated with the $V_i$'s is 1. For a resistive network with $N$ nodes at nonfixed voltage values we have $N$ linear equations similar to the one above, one for each node. Solving this set of equations,

along with the condition that the voltage at a fixed node $h$ is the constant value $V_h$, provides the exact solution. In thermal analysis, the fixed node could correspond to the ambient, which is at a fixed temperature. Alternatively, if a package model is provided, it can be appended to the FDM circuit, with some node(s) of the package model being connected to the ambient, and the combined circuit can be solved to obtain temperatures within the structure.

Having considered the resistive circuit problem, let us construct a random walk "game," given a finite undirected connected graph (for example, Figure 2.10) representing a street map. A walker starts from one of the nodes, and goes to an adjacent node $i$ every day with probability $p_{x,i}$ for $i = 1, 2, \ldots, \text{degree}(x)$, where $x$ is the current node, and degree$(x)$ is the number of edges connected to node $x$. These probabilities satisfy the following relationship:

$$\sum_{i=1}^{\text{degree}(x)} p_{x,i} = 1. \tag{2.68}$$

The walker pays an amount $m_x$ to a motel for lodging everyday, until he/she reaches one of the homes, which are a subset of the nodes. If the walker reaches the home $h$, he/she will stay there and be awarded a certain amount of money, $m_{0h}$; note that this value can be different at different homes. We will consider the problem of calculating the

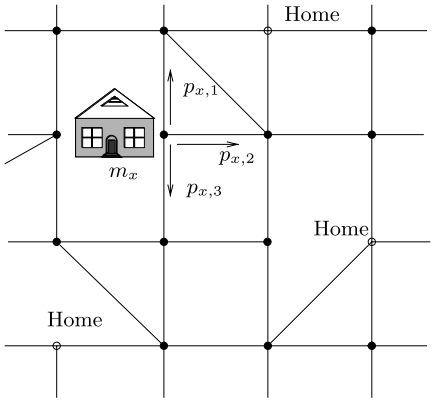



Fig. 2.10 An instance of a random walk "game."

expected amount of money that the walker has accumulated at the end of the walk, as a function of the starting node, assuming he/she starts with nothing. The gain function for the walk is therefore defined as

$$f(x) = E[\text{total money earned}|\text{walk starts at node } x]. \tag{2.69}$$

It is obvious that

$$f(\text{one of the homes}) = m_{0h}. \tag{2.70}$$

For a nonhome node $x$, assuming that the nodes adjacent to $x$ are labeled $1, 2, \ldots, \text{degree}(x)$, the $f$ variables satisfy

$$f(x) = \sum_{i=1}^{\text{degree}(x)} p_{x,i} f(i) - m_x. \tag{2.71}$$

For a random-walk problem with $N$ nonhome nodes, there are $N$ linear equations similar to the one above, and the solution to this set of equations will give the exact values of $f$ at all nodes.

It is easy to draw a parallel between this problem and that of resistive network analysis. Equation (2.71) becomes identical to (2.67), and Equation (2.70) reduces to the condition of constant voltage sources.

$$\begin{aligned} p_{x,i} &= \frac{g_i}{\sum_{j=1}^{\text{degree}(x)} g_j} \quad i = 1, 2, \ldots, \text{degree}(x) \\ m_x &= \frac{I_x}{\sum_{j=1}^{\text{degree}(x)} g_j} \quad m_{0h} = V_h, \quad f(x) = V_x. \end{aligned} \tag{2.72}$$

In other words, for any resistive network problem, we can construct a random walk problem that is mathematically equivalent, i.e., characterized by the same set of equations. It can be proven that such an equation set has and only has one unique solution [41]. Therefore, if we find an approximated solution for the random walk, it is also an approximated solution for the resistive network.

A natural way to approach the random walk problem is to perform a certain number of experiments and use the average money left in those experiments as the approximated solution. If this amount is averaged over a sufficiently large number of walks by playing the "game" a sufficiently large number of times, then by the law of large numbers, an

acceptably accurate solution can be obtained, and the error can be estimated using the Central Limit Theorem [164].

A desirable feature of the proposed algorithm is that it localizes the computation, i.e., it can calculate a single node voltage without having to solve the whole circuit. As compared to a conventional approach that must solve the full set of matrix equations to find the voltage at any one node, the computational advantage of this method could be tremendous. Numerous efficiency enhancing techniques are available for this approach, and are described in further detail in [112, 114].

Another version of this approach uses the random walk solver to obtain a high-quality preconditioner for an iterative method. The essential idea is that the information gathered during the random walks, when organized well, can be used to generate approximate LU factors at almost no additional cost. These approximate LU factors may then be used to generate a preconditioner, whose quality is shown to be superior to many existing methods [111, 113].

## 2.5 Transient Thermal Analysis

In this section, we will provide an outline of approaches that may be used for transient analysis. Our description will primarily focus on FDM-based formulations, since these are most widely used in the IC world today. However, it is understood that equivalent approaches may also be used under other methods, typically using a timestepping-based approach. Alternatively, it is possible to use frequency-domain techniques [163], as is common in the case of analyzing transients in electrical circuits.

For transient thermal analysis, the time-dependent left-hand side term in Equation (2.3) is nonzero. Using a similar finite differencing strategy as in Section 2.3.1, the equation may be discretized in the space domain as

$$\rho c_p \frac{\partial T_{i,j,k}}{\partial t} = k_t \left[ \frac{\delta_x^2 T_{i,j,k}^{n+1} + \delta_x^2 T_{i,j,k}^{n}}{2(\Delta x)^2} + \frac{\delta_y^2 T_{i,j,k}^{n+1} + \delta_y^2 T_{i,j,k}^{n}}{2(\Delta y)^2} + \frac{\delta_z^2 T_{i,j,k}^{n+1} + \delta_z^2 T_{i,j,k}^{n}}{2(\Delta z)^2} \right] + \frac{g_{i,j,k}^{n}}{\rho c_p}. \tag{2.73}$$

Again, using similar substitutions as in Section 2.3.1, the time-independent terms on the right-hand side can again be considered to be the currents per unit volume through thermal resistors and thermal current sources. The left-hand side, on the other hand, represents a current source of value $\rho c_p \frac{\partial T_{i,j,k}}{\partial t}$. Recalling that in the thermal–electrical analogy, the temperatures correspond to voltages, it is easy to see that we can represent the left-hand side by a *thermal capacitor* of value $\rho c_p$ per unit volume.

Given this mapping, transient thermal analysis can be performed by creating the equivalent network consisting of resistors, current sources, and capacitors, and using routine electrical techniques for transient analysis.

### 2.5.1 The ADI Method

An alternative iterative approach that is often used in the heat transfer context is the so-called alternating-direction-implicit (ADI) method. This method is described in standard texts on heat transfer, and was employed in [151] for on-chip transient thermal analysis. The left-hand side of Equation (2.73) is discretized with respect to the time variable $t$ to obtain

$$\frac{T_{i,j,k}^{n+1} - T_{i,j,k}^{n}}{\Delta t}. \tag{2.74}$$

The work in [151] adopted the Crank–Nicholson method for discretizing Equation (2.3), which uses the centered finite difference in space and the trapezoidal rule in time (see Figure 2.11).

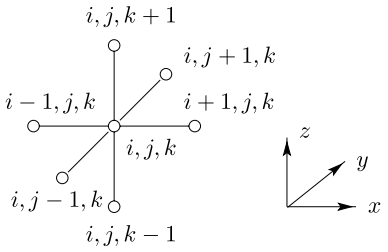
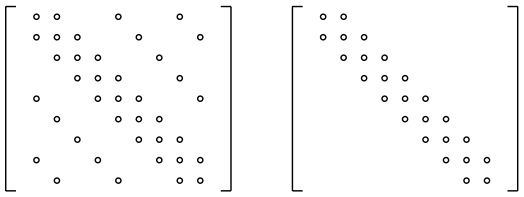



Fig. 2.11 The node indexing scheme for the spatial discretization.



(a) (b)

Fig. 2.12 The nonzero patterns of the coefficient matrices. (a) The coefficient matrix for solving the discretized PDE directly. (b) The coefficient matrix in each sub-step of the ADI algorithm.

The objective is to solve for $T_{i,j,k}^{n+1}$ for all $i, j, k$ assuming that $T_{i,j,k}^{n}$'s are known. Then through time marching, the evolution of the temperature distribution as a function of time can be obtained. It is not difficult to see that if we number the nodes in a lexicographical order and collect the linear equations associated with each node, the resulting coefficient matrix of the equations will be sparse and banded, as shown in Figure 2.12(a). However, solving this system of linear equations directly takes super-linear time. To improve the overall efficiency of thermal analysis, the authors of [151] adopted an ADI algorithm to solve Equation (2.73). Specifically, each time step is split into three sub-steps, and in each sub-step, a system of finite difference equations that is implicit in only one spatial direction is constructed, as shown below.

Step I:

$$T_{i,j,k}^{n+(1/3)} - T_{i,j,k}^{n} = \frac{r_x \delta_x^2}{2}\left(T_{i,j,k}^{n+(1/3)} + T_{i,j,k}^{n}\right) + r_y \delta_y^2 T_{i,j,k}^{n} + r_z \delta_z^2 T_{i,j,k}^{n} + \frac{\Delta t}{\rho c_p} g_{i,j,k}^{n} \quad (2.75)$$

Step II:

$$T_{i,j,k}^{n+(2/3)} - T_{i,j,k}^{n} = \frac{r_x \delta_x^2}{2}\left(T_{i,j,k}^{n+(1/3)} + T_{i,j,k}^{n}\right) + \frac{r_y \delta_y^2}{2}\left(T_{i,j,k}^{n+(2/3)} + T_{i,j,k}^{n}\right) + r_z \delta_z^2 T_{i,j,k}^{n} + \frac{\Delta t}{\rho c_p} g_{i,j,k}^{n} \quad (2.76)$$

Step III:

$$T_{i,j,k}^{n+1} - T_{i,j,k}^{n} = \frac{r_x \delta_x^2}{2}(T_{i,j,k}^{n+(1/3)} + T_{i,j,k}^{n}) + \frac{r_y \delta_y^2}{2}\left(T_{i,j,k}^{n+(2/3)} + T_{i,j,k}^{n}\right) + \frac{r_z \delta_z^2}{2}\left(T_{i,j,k}^{n+1} + T_{i,j,k}^{n}\right) + \frac{\Delta t}{\rho c_p} g_{i,j,k}^{n}, \quad (2.77)$$

where

$$r_x = \frac{k_t \Delta t}{\rho c_p}\frac{1}{\Delta x^2}, \quad r_y = \frac{k_t \Delta t}{\rho c_p}\frac{1}{\Delta y^2}, \quad r_z = \frac{k_t \Delta t}{\rho c_p}\frac{1}{\Delta z^2}. \quad (2.78)$$

The unknowns in the three sub-steps are $T_{i,j,k}^{n+(1+3)}$'s, $T_{i,j,k}^{n+(2/3)}$'s, and $T_{i,j,k}^{n+1}$'s, respectively. It can be seen that the coefficient matrix of the finite difference equations in each of the sub-steps is tridiagonal as shown in Figure 2.12(b), and tridiagonal linear systems can be solved in linear time. As a result, the overall runtime of performing the transient thermal analysis via the ADI approach is $O(ns)$, where $n$ is the total number of nodes in the finite difference mesh and $s$ is the number of time steps.

### 2.5.2 HotSpot: An Approach for Architectural-Level Transient Analysis

The work in [135] first proposed HotSpot, a technique for architectural-level thermal analysis that can be coupled to architectural simulators. Subsequently, several updates and elaborations on the approach have been outlined in [60, 61, 62, 63, 64, 134], and a public-domain release of the tools is available [1] and widely used. The goal of this approach is to obtain a coarsely discretized FDM solution to the on-chip thermal problem, with an FDM/FEM-like model for the package components. This coarse granularity is appropriate at the architectural level, where the power of large functional blocks can be estimated, but details of the design are as yet unknown. Thermal analysis at this stage can be linked to a cycle-accurate architectural power estimation tool such as Wattch [16]. Such an approach can be extremely useful in making thermally conscious decisions at the architectural level.

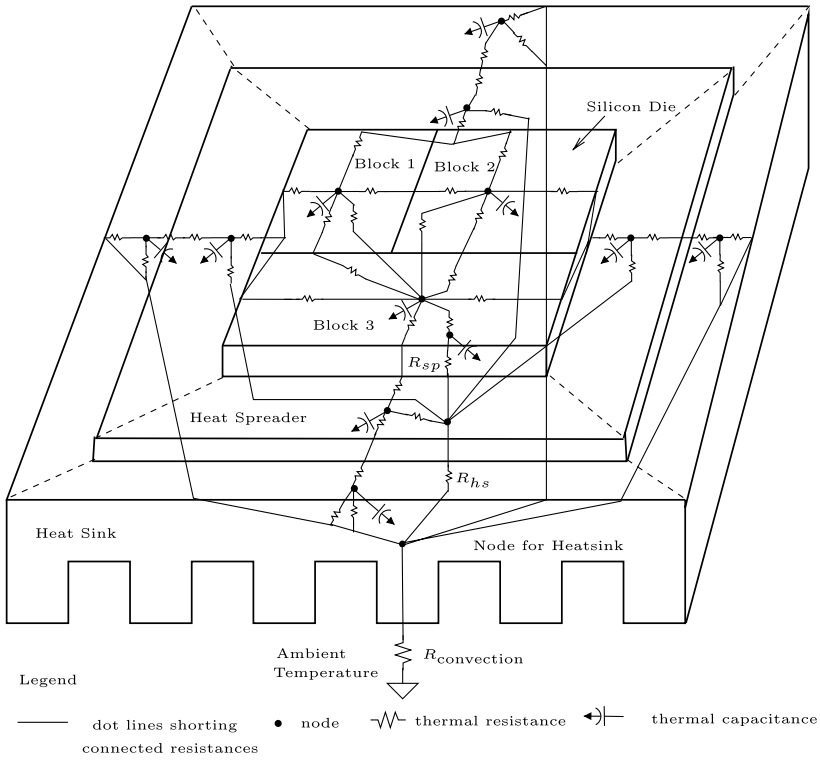



Fig. 2.13 An RC model of the chip, used in HotSpot [134].

The thermal model used in HotSpot is an RC model, illustrated in Figure 2.13. Depending on the accuracy desired, the chip area may be modeled using an FDM approach where the discretization corresponds either to (i) a grid of regions, the center of each of which is a node in the thermal network, such that the power dissipation of each block is assumed to be uniformly distributed over the grid points that it covers, or (ii) a set of rectangular regions corresponding to functional blocks, where the center of each functional block corresponds to a node (in subsequent versions of HotSpot, blocks with skewed aspect ratios are divided into subblocks whose aspect ratios are closer to 1:1). The heat spreader is divided into five blocks, one directly below the die, and four trapezoids corresponding to the peripheral region that is not covered by the die; likewise, the heat sink is also divided into five blocks,

corresponding to the area under the spreader and four similarly constructed external trapezoids. Within each layer, lateral conduction is captured by resistors between adjacent blocks, and vertical conduction is modeled by a resistance between each block and the block(s) adjacent to it in an adjoining layer. A thermal capacitance is connected to each node in the circuit, the heat sink is assumed to be connected to the ambient, and the power sources correspond to the power associated with each functional block on the die. Both the primary heat flow path through the heat sink, and the secondary heat flow path, through the CBGA grid and the PCB, are modeled.

Interconnect effects are captured by modeling the self-heating effects of wires, and using Rent's rule-based wire density predictions on each layer to obtain an effective thermal conductivity for the region around a wire. The composite thermal RC system is solved using a fourth-order Runge–Kutta method at each time step, to obtain the result of a transient analysis.

# 3

# Effects on Circuit Performance

In this survey, we will consider the effect of elevated on-chip temperatures on circuit performance. Specifically, we will address the issue of thermal effects on circuit delay, power, and reliability.

## 3.1 Circuit Delay as a Function of Temperature

Increases in temperature can affect the parameters of both transistors and on-chip interconnects, thus affecting the circuit delay. Thermal variations can affect transistor behavior in two ways:

- The mobility, $\mu$, of charge carriers in a transistor reduces with increasing temperature, $T$, according to the equation

$$\mu(T) = \mu(T_0) \left( \frac{T}{T_0} \right)^{-m}, \tag{3.1}$$

  where $T_0$ is room temperature (typically, 300 K), and $m > 0$ is the mobility temperature exponent, with a typical value of 1.7 in highly doped silicon, and 1.4 in nanometer-thin silicon layers, where boundary scattering becomes important [109]. This reduction in the mobility lowers the drive current of

a transistor, leading to a tendency toward increased delays with increasing temperatures.

- The threshold voltage of a transistor, $V_{\text{th}}$, decreases with increasing temperature along the trajectory

$$V_{\text{th}}(T) = V_{th}(T_0) - \kappa(T - T_0), \tag{3.2}$$

where $\kappa > 0$ is the threshold voltage temperature coefficient with a typical value of 2.5 mV/K [74]. This trend makes it easier for a transistor to switch on as temperatures rise, and implies a tendency for an increased drive current. Therefore, this effect results in a reduction in the circuit delay with increasing temperatures.

The two effects above are in contradiction, and depending on which of the two is more dominant, one may see either *negative* temperature dependence, where delays increase with temperature, or *positive* temperature dependence, where delays decrease with temperature. It is also possible that neither effect will dominate over the entire range of temperatures, implying that there may be nonmonotonic behavior in the delay-temperature curve: this corresponds to *mixed* temperature dependence. Issues related to mixed/inverted temperature dependence have been addressed in [39, 45, 74].

Interconnect parameters are also affected by thermal effects. The resistance, $R$, of a wire segment increases with temperature as follows:

$$R = R_0(1 + \beta(T - T_0)), \tag{3.3}$$

where $R_0$ is the resistance of the wire at room temperature, $T_0$. The value of $\beta$ is typically positive, and this implies that the RC delay of a wire typically increases with temperature. Moreover, since wires often travel through large lengths of insulating media, the heat generated in a wire can result in localized self-heating, or Joule-heating, effects [9].

## 3.2 The Leakage–Temperature Relationship

The power dissipated in a circuit consists of three components: dynamic power due to the charging/discharging of parasitic capacitances,

short-circuit power due to crowbar current from supply to ground during switching, and static power due to leakage when a transistor is supposedly off. The first two of these are less significantly impacted by temperature, but leakage power has a large temperature dependence. Leakage is now one of the dominant components of the total on-chip power: reported results show cases where more than half of the total power dissipation is due to leakage, and therefore, this is a serious issue. Moreover, leakage is susceptible to process variations [20] and temperature variations [37], which are magnified by the exponential terms associated with various components of leakage, as we will see below.

There are two major constituents of leakage current: subthreshold leakage current and gate tunneling current [122]. The subthreshold leakage current is the leakage current between the drain and source node when the device is in the "off" state (the voltage between the gate and source terminal is zero). Historically, in 0.25 $\mu$m and higher technology nodes, the subthreshold leakage was small enough to be negligible (several orders of magnitude smaller than the on-current). However, the traditional scaling requires the reduction of supply voltage $V_{DD}$, along with the reduction of the channel length. As a consequence, the threshold voltage must be scaled accordingly in order to maintain the driving capability of the MOSFET device. Even though this scaling has not been linear, because of leakage considerations, the scaled threshold voltage increases the proportion of the subthreshold leakage current to the total current. Therefore, subthreshold leakage is a significant factor in nanometer technologies.

The second component of the static power is the gate tunneling current, which is also a consequence of scaling. As the device dimensions are reduced, the gate oxide thickness also must be reduced. An unwanted consequence of thinner gate oxide thickness is the increased gate tunneling leakage current. Compared to the exponential dependence of subthreshold leakage on temperature, gate leakage is less affected by temperature. Moreover, the recent introduction of high-k gate dielectrics is likely to control the gate leakage problem, at least for a few generations.

An expression for the subthreshold leakage current density, i.e., the current per unit transistor area, is given by Mukhopadhyay et al. [95]:

$$J_{\text{sub}} = \frac{W}{L_{\text{eff}}}\mu\sqrt{\frac{q\epsilon_{si}N_{cheff}}{2\phi_s}}v_T^2 \exp\left(\frac{V_{gs} - V_{\text{th}}}{\eta v_T}\right)\left[1 - \exp\left(\frac{-V_{\text{ds}}}{v_T}\right)\right]. \tag{3.4}$$

The details of the parameters in the above equation can be found in [95], but it is important to make a few observations:

- The term $v_T = kT/q$ is the *thermal voltage*, where $k$ is the Boltzmann's constant and $q$ is the electrical charge, and $T$ is the junction temperature. From the equation, we can see that the leakage is an exponential function of the junction temperature $T$.
- The symbol $V_{\text{th}}$ represents the threshold voltage. It can be shown that for a given technology, $V_{\text{th}}$ is a function of the effective channel length $L_{\text{eff}}$. Therefore, subthreshold leakage is also an exponential function of effective channel length.
- The drain-to-source voltage, $V_{\text{ds}}$, is closely related to supply voltage $V_{DD}$, and has the same range in static CMOS circuits. Therefore, subthreshold leakage is an exponential function of the supply voltage.
- The threshold voltage $V_{\text{th}}$ is also affected by the body bias $V_{BS}$. In a bulk CMOS technology, since the body node is always tied to ground for NMOS and $V_{DD}$ for PMOS, the body bias conditions for stacked devices are different, depending on the location of the "off" device on a stack (e.g., top of the stack or bottom of the stack). As a result, the subthreshold leakage current can vary quite significantly when different input vectors are applied to a gate with stacks.

For the gate tunneling current, a widely used model is the one provided in [14]:

$$J_{\text{tunnel}} = \frac{4\pi m^* q}{h^3}(kT)^2\left(1 + \frac{\gamma kT}{2\sqrt{E_B}}\right)\exp\left(\frac{E_{F0,Si/SiO_2}}{kT}\right)\exp(-\gamma\sqrt{E_B}), \tag{3.5}$$

where $T$ is the operating temperature, $E_{F0,Si/SiO_2}$ is the Fermi level at the $Si/SiO_2$ interface and $m^*$ depends on the underlying tunneling mechanism. Parameters $k$ and $q$ are defined as above, and $h$ is Planck's constant: all of these are physical constants. The term $\gamma = 4\pi t_{\text{ox}}\sqrt{2m_{\text{OX}}}/h$, where $t_{\text{ox}}$ is the oxide thickness, and $m_{\text{ox}}$ is the effective electron mass in the oxide. The parameter $E_B$ is the height of the energy barrier, given by

$$E_B = q\left(\xi - \frac{V_{\text{ox}}}{2}\right), \tag{3.6}$$

where $\xi$ is the modified electron affinity in silicon, and $V_{\text{ox}}$ is the applied voltage across the oxide.

Besides physical constants and many technology-dependent parameters, it is quite clear that the gate-tunneling leakage depends on the gate oxide thickness and the operating temperature. The former is a strong dependence, but the latter is more complex: over normal ranges of operating temperature, the variations in gate leakage are roughly linear. In comparison with subthreshold leakage, which shows exponential changes with temperature, these gate leakage variations are often much lower. More details about this model can be found in [14].

One possible solution to mitigate the negative impact of gate current is to use material with higher dielectric constants (i.e., high-k material) in conjunction with metal gates [81]. In many current technologies, the gate leakage component is non-negligible. Recently some progress has been reported on the development of high-k material. If successfully deployed, the new technology can reduce gate tunneling leakage by at least an order of magnitude, and at least postpone the point at which gate leakage becomes significant.

Due to the power consumption limit dictated by the air-cooling technique widely accepted by the industry and market, power consumption, especially static power, has become a major design constraint. Besides the advancements in manufacturing technology and material science, several circuit level power reduction techniques also have implications on the physical design flow. They include power gating, $V_{\text{th}}$ (or effective channel length) assignment, input vector assignment or

any combination of these methods. More details on these topics can be found in [82, 96, 122].

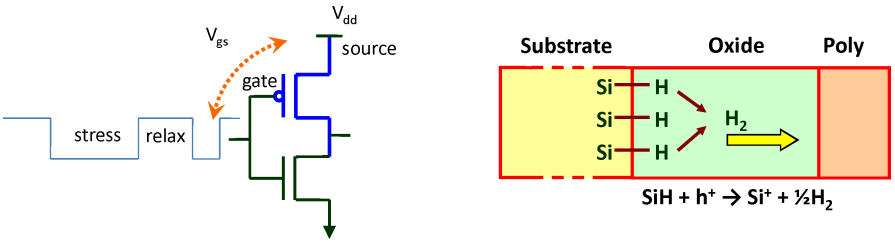
## 3.3 Reliability and Aging Effects

Thermal effects can cause a circuit to age prematurely. In this section, we present an outline of a few reliability effects that are exacerbated by thermal stress, namely, bias temperature instability, oxide breakdown, and electromigration.

### 3.3.1 Bias Temperature Instability

Bias temperature instability is a phenomenon that causes threshold voltage shifts over long periods of time, eventually causing the circuit to fail to meet its specifications. The word "bias" refers to the fact that this degradation is heightened by the application of a bias on the gate node of a transistor.

The phenomenon of negative bias temperature instability (NBTI) can be illustrated with the help of a simple circuit, an inverter, illustrated in Figure 3.1(a). When a PMOS transistor is biased in inversion ($V_{gs} = -V_{dd}$) (for example, when the input of the inverter is at logic 0), interface traps are generated due to the dissociation of $Si$–$H$ bonds along the substrate-oxide interface, as illustrated in Figure 3.1(b). The connection of this mechanism to thermal effects is that the rate of generation of these traps is accelerated by elevated temperatures, and




(a) (b)

Fig. 3.1 (a) An inverter whose PMOS device is alternately subjected to NBTI stress and relax phases. (b) An illustration of the phenomenon of NBTI.

therefore, increased on-chip temperatures can directly affect the lifetime of a chip. The time for which the transistor is stressed is another factor that increases the level of degradation. These traps cause an increase in the threshold voltage ($V_{\text{th}}$), and a reduction in the saturation current ($I_{\text{dsat}}$) of the PMOS transistors. This effect, known as NBTI, has become a significant reliability issue in high-performance digital IC design, especially in sub-130 nm technologies [5, 19, 75, 91, 116, 128]. An increase in $V_{\text{th}}$ causes the circuit delay to degrade, and when this degradation exceeds a certain amount, the circuit may fail to meet its timing specifications. The rate constants of the reactions that define NBTI are dependent on temperature, and are worsened at elevated temperatures.

A corresponding and dual effect, known as Positive Bias Temperature Instability (PBTI) can be seen for NMOS devices, for example, when the input to an inverter is at logic 1, and a positive bias stress is applied across the gate oxide of the NMOS device. Although the impact of a stressing bias on PBTI is lower than NBTI [116], PBTI is becoming increasingly important in its own right. Moreover, techniques are developed to reduce NBTI can contribute to increasing PBTI. For example, for the example of the inverter listed earlier, if the input is biased so that it is more likely to be at logic 1 than logic 0, the NBTI stress on the PMOS device is reduced since the $V_{gs}$ bias becomes zero; however, this now places a bias on the NMOS device, which now has a nonzero $V_{gs}$ value.

The degradation in threshold voltages is at least partially reversible. Experiments have shown that the application of a negative bias ($V_{gs} = -V_{dd}$) on a PMOS transistor leads to the generation of interface traps, while removal of the bias ($V_{gs} = 0$) causes a reduction in the number of interface traps due to annealing [5, 6, 7, 19, 22, 43, 75, 128, 170]. Thus, the impact of NBTI on the PMOS transistor depends on the sequence of stress and relaxation applied to the gate. Since a digital circuit consists of millions of nodes with differing signal probabilities and activity factors, asymmetric levels of degradation are experienced by various timing paths. The exact amount of degradation must be determined using a model that estimates the amount of NBTI-induced shift in the various parameters of the circuit that affect the delay. This metric can then be used to design circuits with appropriate guard-bands, such

that they remain reliable over the desired lifetime of operation, despite temporal degradation.

Over the past few years, there have been many attempts to model the NBTI effect, based on several theories, such as the classical Reaction–Diffusion, dispersive diffusion, hole trapping. Reaction–Diffusion (R–D) theory [71, 100] has been commonly used to model NBTI, leading to various long-term models for circuit degradation [5, 12, 77, 146]. An alternative school of thought relates to the inability of the R–D model to explain some key phenomena, as detailed in [53, 118, 119, 131, 137], leading to models such as [53, 65, 66, 73, 104, 165], as well as efforts to resolve the controversy between the R–D model theory and the hole trapping theory [67, 68, 89].

At the circuit level, the effect of bias temperature instability (BTI) is in alterations of the transistor threshold voltages. Under DC stress, the threshold voltage of a PMOS transistor degrades with time, $t$, at a rate given by

$$\Delta V_{\text{th}} \propto t^n, \tag{3.7}$$

where $n$ is a constant that is theoretically derived to be 1/6, under, for example, the reaction–diffusion model. This has been confirmed by empirical experimental evidence.

However, in general, transistors in a circuit are not continuously stressed, but a sequence of alternating 0s and 1s is applied at their gate nodes. When the stress is removed, the threshold voltage is seen to recover toward its original value, according to the trend shown in Figure 3.2 for a PMOS transistor. Analytical models for the change in threshold voltage are provided in [5] for a single cycle, and extended to multiple cycles in [77].

The degradation in threshold voltage shows a property known as frequency-independence, demonstrated over a wide range of frequencies [5, 21]: in other words, if a pattern of signals is applied to a transistor over time, the degradation depends only on the total fraction of time for which the transistor was stressed, and not on the frequency of the signal, or the distribution of stress/relax times. Accordingly, if one defines a *signal probability* of the signal value at the gate node of the transistor, corresponding to the proportion of time that the transistor

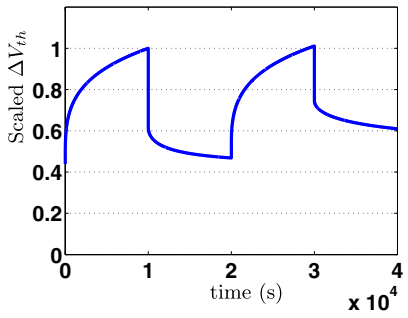



Fig. 3.2 Plot of the first two stress and recovery phases for a transistor with a 12 $\mathring{A}$ thick gate oxide.

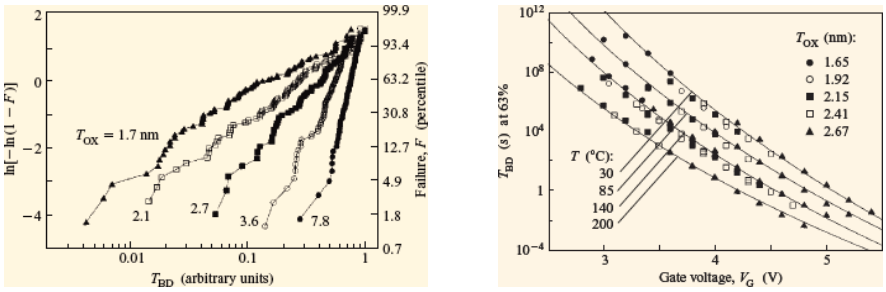
is likely to be under stress, the threshold voltage degradation is only a function of this signal probability, and a one-dimensional look-up table can be used to store this degradation.

### 3.3.2 Oxide Breakdown

Time-dependent dielectric breakdown (TDDB) is a reliability phenomenon in gate oxides that results in a sudden discontinuous increase in the conductance of the gate oxide at the point of breakdown, as a result of which the current through the gate insulator increases significantly. This phenomenon is of particular concern as gate oxide thicknesses become thinner with technology scaling, and gates become more susceptible to breakdown. Various models for explaining TDDB have been put forth, including the hydrogen model, the anode-hole injection model, the thermochemical model (also known as the $E$ model, where $E$ is the electric field across the oxide), and the percolation model: for a survey, the reader is referred to [136, 156]. Unlike BTI, this mechanism is not known to be reversible, and any damage caused can be assumed to be permanent.

The time to breakdown, $T_{\mathrm{BD}}$, can be modeled statistically using a Weibull distribution, whose cumulative density function (CDF) is given by

$$\mathrm{CDF}(T_{\mathrm{BD}}) = 1 - \exp\left(\left[-\left(\frac{T_{\mathrm{BD}}}{\alpha}\right)^{\beta}\right]\right). \tag{3.8}$$




(a) (b)

Fig. 3.3 (a) Normalized cumulative density function for the time to breakdown, $T_{\mathrm{BD}}$, for a range of oxide thicknesses, plotted against the Weibull scale on the left axis, corresponding to the percentiles on the right axis [156, 157]. (b) The breakdown time, $T_{\mathrm{BD}}$, as a function of the gate voltage, $V_G$, at various temperatures [155, 156].

The parameter $\alpha$ corresponds to the time-to-breakdown at about the 63rd percentile, and $\beta$ is the Weibull slope. Representative distributions are shown in Figure 3.3(a). Generally speaking, an increased electric field (i.e., an increased voltage across the gate) accelerates breakdown. The thermal connection corresponds to the fact that at a fixed gate voltage, elevated temperatures lead to faster breakdown, as illustrated in Figure 3.3(b).

Currently, there are few approaches to addressing oxide breakdown issues at the circuit level, although this is likely to change in the coming years.

### 3.3.3 Electromigration

When a current pattern is applied on an on-chip wire for a long period of time, it is seen to cause a physical migration of atoms in the wire, particularly in regions where the current density is high. This can cause the wire to become less wide at best, and to completely open-circuit at worst, and is therefore a serious reliability problem. This problem is witnessed most notably in supply (power and ground) wires [13, 40], where the flow of current is mostly unidirectional, but AC electromigration is also seen in signal wires [138]. Amelioration strategies for electromigration are primarily built in by ensuring that the current density on a wire never exceeds a specified threshold. For the same

current, the use of wider wires results in lower current densities, and therefore, wire-widening is a potent tool for overcoming electromigration, with its accompanying overheads in taking up additional routing area and potentially on signal lines, increased power.

The mean time to failure (MTTF) of a wire under electromigration is described by Black's equation:

$$\text{MTTF} = AJ^{-n}e^{Q/kT}, \tag{3.9}$$

where $J$ is the average current density in the wire, $n$ is an empirical exponent whose value is about 2, $Q$ is the activation energy for grain-boundary diffusion, equal to about 0.7 eV for Al–Cu, $k$ is Boltzmann's constant, and $T$ is the temperature of the wire.

The role of temperature is clear from the above equation: elevated temperatures cause the MTTF to reduce, i.e., degrade the lifetime of the wire, and hence, the chip. It is important to note that the temperature of the wire is affected not only by the heat produced by the devices in the circuit, but also by Joule heating effects within the wire.

# 4

# Thermal Optimization

On-chip thermal effects can be overcome through a number of strategies, which can be classified as follows:

- One class of methods attempts to reduce the root cause of these problems, namely, the amount of power dissipated on-chip. *Low-power design* has been the subject of considerable research for over a decade, and is not addressed in this survey.
- Assuming that the units that are designed dissipate as little power as is reasonably possible, a second approach comes into play, using *thermal management strategies* to control the temperature profile across the chip. This primarily involves altering the distribution of the heat sources, and where applicable, improving the heat sinking pathways on the chip. While off-chip heat sinking can also be improved, it is not addressed in this survey.
- Even with assiduous application of the above optimizations, the temperature on a chip will certainly rise above the ambient temperature under normal use patterns. This implies that the chip will see corresponding degradations in performance

and reliability over time, and therefore, a third class of approaches uses *mitigative methods* to ensure that the circuit behaves correctly over its predicted lifetime.

These optimizations may be performed at all levels of design, ranging from the architectural level to the logic and transistor level. At the architectural level, the best estimate of the layout corresponds to a floorplan: the details of the design are far from being finalized, and only coarse estimates of the power or performance metrics of the circuit are available. However, this early stage of design provides a great deal of flexibility, and decisions made at this point can have a large impact on the behavior of the circuit. At the other end of the spectrum, at the logic/transistor level, a great deal of design detail is available, in that the power consumptions of individual macro cells or blocks are all well known. Therefore, circuit power and performance metrics can be measured accurately, and any promised changes through circuit optimization can more easily be delivered. However, at this stage, the level of flexibility for design changes is low. Therefore, optimizations at all stages of design are important, but those at higher levels of abstraction must be accompanied by guarantees that they will percolate through to the end of the design.

This survey provides a set of example techniques that may be used to perform thermal optimization at all levels of design. We begin with an overview of microarchitectural optimizations, then move to techniques for thermally aware physical design in 3D circuits, and conclude with mitigation techiques using adaptive body biases, adaptive supply voltages, and guardbanding.

## 4.1 Microarchitecture-Driven Floorplanning

At early stages of design, there is a strong coupling between physical layout and temperature. The floorplan of the chip determines the spatial distribution of the power sources within the layout, and therefore, plays a key role in determining the temperature. Coarsely speaking, if the high-power modules are placed close to each other, the thermal profile will be worse than the case where they are spread apart. However, there are also performance implications to moving the modules apart.

For instance, as interconnect effects have grown in importance over the years, the delays of global signals have come to exceed a single clock cycle [127], even when the corresponding wires are optimally designed. This requires the use of wire-pipelining [29, 58], whereby latencies are added on to wires to capture their multicycle delays, in order to support high operating frequencies.

As a result, in the nanometer regime, the choice of a floorplan can significantly affect the performance of a processor design, measured in terms of the number of instructions per cycle (IPC) [42, 69, 87, 98]. The choice of floorplan may have an impact on the spatiotemporal distribution of power, and consequently, affects the thermal profile of the circuit. Figure 4.1 shows the profile of power dissipated in a specific floorplan block, as a function of time, for two different floorplans. The two different layouts have different distributions for the multicycle global wires that must be pipelined, and therefore, correspond to different values for the CPI. The changes in the communication latencies causes a difference in the times the blocks are activated, and hence, in the temporal distribution of power within the block between the two floorplans: it can be seen that one of the profiles has larger peaks and values than the other, which is relatively smoother, and this is entirely due to the sequential latency of interconnect buses.

At the architectural level, therefore, it is essential to consider such interaction between IPC and power (and hence temperature) and jointly optimize both the performance and temperature objectives through floorplanning. The topic of microarchitecturally

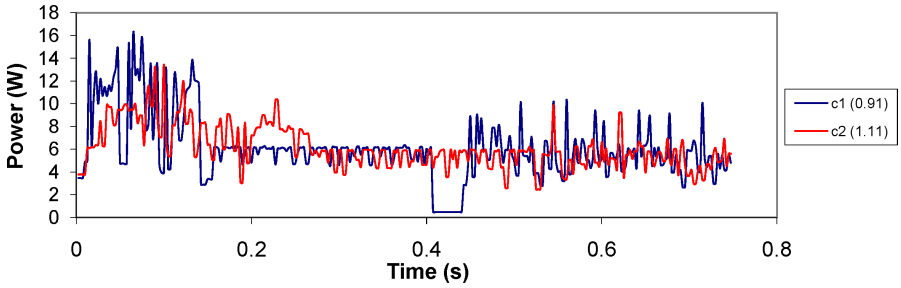



Fig. 4.1 A transient thermal simulation for benchmark gcc on the Alpha architecture, for two sets of wire latencies, c1 and c2.

aware floorplanning has attracted some attention in the last few years, and some work has been carried out in the area of temperature-aware microarchitecture floorplanning [56, 59, 99, 124, 158]. In this section, we will focus our description on the work in [99], whose objective differs from the others in that it performs thermal optimization under transient analysis (as against steady-state analysis in the other approaches), which can capture the effects shown in Figure 4.1, and works not only on optimizing the peak temperature, but also the temporal average of the temperature, which is a key factor that influences a number of aging and reliability mechanisms.

### 4.1.1 Overview of the Microarchitecture

The IPC and power data is computed using Wattch [16], based on `sim-outorder` [17] simulator. The microarchitecture that employed in this survey is based on the DLX architecture [17] and resembles a real processor, Alpha 21362 [10]. The configuration and the corresponding functional blocks are shown in Table 4.1 and Figure 4.2, respectively. The instruction fetch and decode blocks are

Table 4.1 Block configuration of the processor.

| Parameter | Value |
|---|---|
| Fetch width | 8 instrs/cycle |
| Issue width | 8 instrs/cycle |
| Commit width | 8 instrs/cycle |
| RUU entries | 128 |
| LSQ entries | 64 |
| IFQ entries | 16 |
| Branch pred | comb, 4K table<br>2-lev 2K table, 11-bit<br>2K BHT |
| BTB | 512 sets, 4-way |
| IL1 | 64K, 64B, 2-way<br>LRU, latency: 1 |
| DL1 | 32K, 32B, 2-way<br>LRU, latency: 1 |
| L2 | 2M, 128B, 4-way<br>latency: 12 |
| ITLB, DTLB | 128 entries<br>Miss latency: 200 |

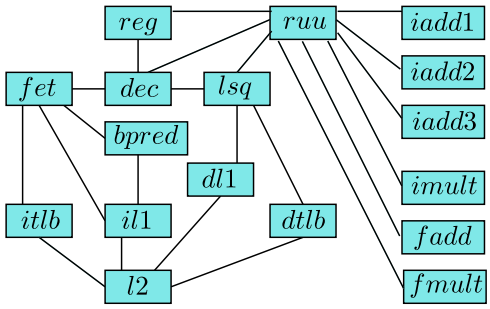



Fig. 4.2 Microarchitecture and buses.

shown as $fet$ and $dec$, respectively, while $il1$ and $dl1$ are the level-1 instruction and data caches, respectively. The instruction and data translation look-aside buffers (TLB) are indicated as $itlb$ and $dtlb$, respectively, while $l2$ represents the unified level-2 cache. The block $ruu$ is the register update unit, which contains the reservation stations and instruction issue logic, while the block $lsq$ represents the load store queue. The system register file is represented by $reg$, whereas $bpred$ consists of the branch predictor and the target buffer (BTB), which predict the direction and target address for a branch instruction, respectively. The blocks $iadd1$, $iadd2$, $iadd3$, $imult$, $fadd$, and $fmult$ are the functional units that execute arithmetic and logic instructions.

Figure 4.2 also shows the 22 system buses that can impact the performance (IPC) and block power densities of the processor, when pipelined. The impact of the bus latencies is modeled as dummy pipeline stages in the simulator and latencies are made configurable. For instance, extra flip–flops inserted on, say, the bus between $ruu$ and $fadd$ units shown in Figure 4.2 can be modeled as an increase in the latency of a floating-point add instruction. To achieve this, the 22 buses are grouped into 19 factors that can be made configurable in the chosen simulator. Each buses, for the most part, maps on to a single factor. The first exception is the factor $extra\_fet$, which represents the sum of the latencies of three buses. This factor corresponds to the number of extra stages to be inserted in the *fetch* stage of the pipeline of the processor. The second and last exception is the factor $max\_lsq\_ruu$, which models

the maximum of the latencies of the buses $dec\_ruu$ and $dec\_lsq$. Interactions are modeled through additional factors: for example, the bus between the $ruu$ and $fadd$ units of Figure 4.2 is indicated as $ruu\_fadd$ in the table.

### 4.1.2 Simulation Strategy

The simulations required in this method involve techniques to measure the temperature and the throughput. At the architectural level, this method uses HotSpot [60] for thermal analysis. In HotSpot, the nodes of the multi-layered thermal network are the centers of the blocks of the microarchitecture. HotSpot is used in transient mode, where it accepts a floorplan, the length of the timestep for transient analysis, and the block power dissipations averaged over each transient timestep as inputs. The heat equation is solved for each timestep to estimate the new set of temperatures (with the initial conditions being those of the previous timestep). The leakage power component of the succeeding timestep can then be updated from the new set of temperatures, and the process repeats until the end of the simulation period. The timestep for simulation is chosen so that it is sufficiently smaller than the thermal time constant of the system, which is of the order of tens of milliseconds.

To measure the throughput, statistical design of experiments, an approach that characterizes the response of a system in terms of changes in the factors which influence the system, is employed. The basic idea is to conduct a set of experiments, in which all factors are varied systematically over a specified range of acceptable values, such that the experiments provide an appropriate sampling of the entire search space. The subsequent analysis of the resulting experimental data will identify the critical factors, the presence of interactions between the factors, etc. In this survey, the system is a microarchitecture, such as that shown in Figure 4.2, the response is the IPC/power, and the factors that influence the IPC of the microarchitecture are the latencies of the buses of the microarchitecture. Since it is impractical to fully explore the exponential search space, even when the number of factors (buses) is small ($N = 22$), a *fractional factorial* design is employed to

reduce the number of simulations. However, such designs are only valid when some or all of the interactions between the factors are negligible.

The potentially significant interactions that are incorporated into the DOE framework are shown below:

- Functional unit scheduling: The number of latencies inserted on the three buses between the register update unit and the three integer adders can be different, and while issuing an integer add instruction, of all the available units, the one with the least latency is chosen. This indicates possible significant (two and three factor) interactions, which need to be estimated.
- Decode stage: The number of extra pipeline stages to be inserted is modeled as a maximum function of three of the latencies of the buses *dec–reg*, *dec–ruu*, and *ruu–reg* (refer to 4.2). Such a nonlinear function can result in significant (two and three) factor interactions among these three factors.

To control the number of simulations, each factor is restricted to have two values: the minimum and the maximum possible values for the factor. The idea is that, by stimulating the system with inputs at their extreme values, the greatest response is provoked for each input. The assumption is that the system response is a monotone function of changes in the inputs (factor levels). Since the factor levels represent bus latencies, the extreme (high and low) values can be obtained by assuming worst-case and best-case scenarios for the corresponding wire lengths.

These assumptions allow the use of a 2-level resolution III fractional factorial design [94]. For $N$ factors, the number of experiments required is equal to the nearest highest power of 2, which turns out to be 32 for our work, since $N = 22$. We refer the reader to [94] for more details of the factorial design methodology.

However, cycle-accurate simulations are inherently slow and most SPEC benchmarks with `reference` input sets, when simulated to completion can take days to complete. Therefore, although the resolution III design strategy of Section 4.1.2 requires a small number of simulations, the run time of each simulation is still an issue. The simulations

are sped up through the use of SMARTS [159], a periodic sampling technique, which works well both for throughput (IPC) and power/energy, particularly for the SPEC benchmarks.

The SMARTS technique that is used to speed up the simulations involves fastforwarding program segments between successive samples chosen for detailed simulation. However, the transient modeling that is used to estimate the thermal metrics requires that the block power densities be collected periodically for every timestep. For this, the power data collected for each sample is extrapolated for the succeeding fastforwarded portion.

The total execution time obtained from a simulation is then segmented into slots of size equal to the transient analysis timestep. In other words, the data collected from the simulation can be arranged as an array $P$ indexed by the timestep and the block number, i.e., the entry $P(a,b)$ of the array corresponds to the power consumption of block $b$ (one of the 17 blocks of Figure 4.2) during timestep $a$. Since 32 simulations are performed (per benchmark), there are 32 such tables. For each entry $P(a,b)$ (per benchmark), a regression model is constructed from the 32 values [94], where the variables are the bus latencies. Equation (4.1) shows one such model, constructed to estimate the power dissipation at entry $P(a,b)$, where $\beta_i$s represent the regression coefficients computed from the 32 values obtained for the correspond entry $(a,b)$. Each $x$ variable in the model, say $x_i$, represents an encoding of the latency of bus $i$, $l_i$, where the minimum and the maximum latencies are coded as $-1$ and $+1$, respectively, and $\mathcal{I}$ is the set of interaction terms described above. In other words,

$$x_i = -1 + \left(\frac{2 \cdot l_i}{\min(i) + \max(i)}\right), \quad 1 \leq i \leq 19$$

$$P(a,b) = \beta_0 + \sum_{i=1}^{19} \beta_i \cdot x_i + \sum_{(ij)\in\mathcal{I}} \beta_{ij} \cdot x_i \cdot x_j + \sum_{(ijk)\in\mathcal{I}} \beta_{ijk} \cdot x_i \cdot x_j \cdot x_k. \tag{4.1}$$

An IPC regression model is similarly constructed from the statistics gathered from the 32 simulations for each benchmark.

### 4.1.3 Floorplanning Flow

Figure 4.3 shows the flow of the proposed temperature-aware microarchitecture floorplanning methodology. The approach accepts a microarchitecture block configuration, a set of buses, benchmarks and a target frequency as inputs and generates a floorplan of the blocks that is optimal in both IPC and temperature.

The primary issue of the design flow is estimating the IPC and the block power dissipations required to generate the temperature distribution of the microarchitecture layout. Specifically, the number of pipelined latencies required by each bus of the microarchitecture is proportional to its length, and therefore for every floorplan, there is a corresponding bus-latency configuration, and consequently a corresponding IPC and power and temperature distribution. However, the large search space explored during floorplanning optimization makes it virtually impossible to use simulations for each floorplan that is to be evaluated. Specifically, if each of $n$ wires on a layout can have $k$ possible latencies, then the cycle-accurate simulator may have to perform up to $n^k$ simulations to fully explore the search space. The work in [99] uses a simulation strategy adapted from [98], based on statistical design of experiments (DOE) to limit the number of cycle-accurate simulations to a practical level. This approach, which reduces the number of simulations to a linear function of $n$, forms the preprocessing step of the flow. The objective of this step is to encapsulate the IPC and power

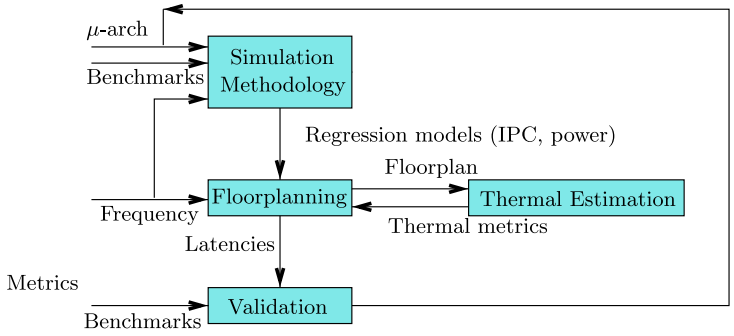



Fig. 4.3 Thermal aware floorplanning: Design flow.

dependence on bus latencies in the form of regression models, which are used by the floorplanner and the thermal simulator.

The floorplanner is based on a simulated annealing (SA) framework and uses the regression models to optimize a cost function, which is a weighted sum of traditional objectives such as the chip area ($Area$) and the aspect ratio ($AR$), as well as thermally related cost components related to the average ($T_{\text{avg}}$) and the peak ($T_{\text{peak}}$) transient temperatures, as shown below:

$$\text{Cost} = W_1 \cdot \text{Area} + W_2 \cdot AR + W_3 \cdot \frac{1}{CPI} + W_4 \cdot (T_{\text{avg}} + T_{\text{peak}}), \tag{4.2}$$

where the $W$s represent the relative weights of the optimization terms.

After every SA move, the floorplanner constructs the block power densities from the regression models derived in the DOE procedure, and passes the data along with the corresponding layout to the thermal simulator, which in turn returns the thermal metrics that are used in the floorplanning cost function. The performance and thermal profile of the resultant layout can then be estimated from cycle-accurate simulations. In addition, the entire design flow of Figure 4.3 may be repeated for several microarchitectural block configurations to identify the optimal configuration [31].

The application of this approach shows significant improvements over methods that are not thermally aware, with peak temperature reductions of up to about 20%, and average temperature reductions of up to about 15%.

## 4.2 Thermally driven Placement and Routing for 3D Circuits

Our second case study on thermally driven optimization operates at a lower level of abstraction than the first, and involves the problem of thermally driven physical design for 3D circuits, which are liable to have acute thermal problems, as described in Section 1.2.2. Physical design provides a great deal of flexibility in rearranging the locations of the heat sources, and in improving the conductivity of thermal pathways in a circuit.

A typical ASIC-like physical design flow begins with a floorplanning step, followed by detailed placement of the cells in the layout, and then a routing step in which the cells are interconnected under the available routing resources. In the 3D context, 3D-specific geometrical considerations must be used, for example, for wire length metrics; temperature considerations must be treated directly; and the scarce inter-tier via resources must be carefully managed.

It is instructive to view the result of a typical 3D thermally aware placement [49]: a layout for the benchmark circuit, ibm01, in a four-tier 3D process, is displayed in Figure 4.4. The cells are positioned in ordered rows on each tier, and the layout in each individual tier looks similar to a 2D standard cell layout. The heat sink is placed at the bottom of the 3D chip, and the lighter shaded regions are hotter than the darker shaded regions. The coolest cells are those in the bottom tier, next to the heat sink, and the temperature increases as we move to higher tiers. The thermal placement method consciously mitigates the temperature by making the upper tiers sparser, in terms of the percentage of area populated by the cells, than the lower tiers.

In addition to spreading out the heat sources through placement, another thermal optimization enhances heat removal in 3D circuits by the judicious insertion of *thermal vias* within the layout. These

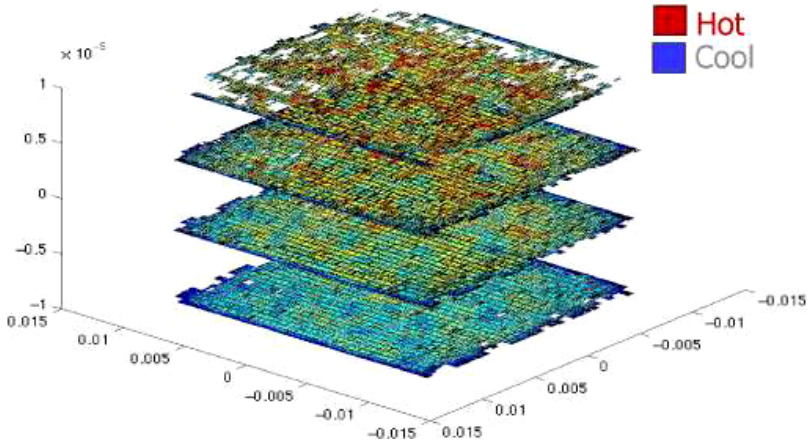



Fig. 4.4 A placement for the benchmark ibm01 in a four-tier 3D technology [2].

vias correspond to inter-tier metal connections that have no electrical function, but instead, constitute a passive cooling technology that draws heat from the problem areas to the heat sink. Thermal via insertion can be built into floorplanning, placement and routing, or performed as an independent post-processing step, depending on the design methodology.

### 4.2.1 Thermal Vias

While silicon is a good thermal conductor, with half or more of the conductivity of typical metals, many of the materials used in 3D technologies are strong insulators that place severe restrictions on the amount of heat that can be removed, even under the best placement solution. The materials include epoxy bonding materials used to attach 3D tiers, or field oxide, or the insulator in an SOI technology. Therefore, the use of deliberate metal lines that serve as heat removing channels, called "thermal vias," are an important ingredient of the total thermal solution. The second step in the flow determines the optimal positions of thermal vias in the placement that provides an overall improvement in the temperature distribution. In realistic 3D technologies, the footprints of these inter-tier vias are of the order of $5\,\mu\text{m} \times 5\,\mu\text{m}$.

In principle, the problem of placing thermal vias can be viewed as one of determining one of two conductivities (corresponding to the presence or absence of metal) at every candidate point where a thermal via may be placed in the chip. However, in practice, it is easy to see that such an approach could lead to an extremely large search space that is exponential in the number of possible positions; note that the set of possible positions in itself is extremely large.

Quite apart from the size of the search space, such an approach is unrealistic for several other reasons. First, the wanton addition of thermal vias in any arbitrary region of the layout would lead to nightmares for a router, which would have to navigate around these blockages. Second, from a practical standpoint, it is unreasonable to perform full-chip thermal analysis, particularly in the inner loop of an optimizer, at the granularity of individual thermal vias. At this level of detail, individual elements would have to correspond to the size of a thermal via,

and the size of the thermal simulation matrix would become extremely large.

Fortunately, there are reasonable ways to overcome each of these issues. The blockage problem may be controlled by enforcing discipline within the design, designating a specific set of areas within the chip as potential thermal via sites. These could be chosen as specific inter-row regions in the cell-based layout, and the optimizer would determine the density with which these are filled with thermal vias. The advantage to the router is obvious, since only these regions are potential blockages, which is much easier to handle. One could work with a two-level scheme with relatively large elements, where the average thermal conductivity of each region is a design variable. Once this average conductivity is chosen, it could be translated back into a precise distribution of thermal vias within the element that achieves that average conductivity.

Various published methods take different approaches to thermal via insertion. We will now describe an algorithm to post-facto thermal via insertion [50]; other procedures perform thermal via insertion during floorplanning, placement or routing, and will be discussed in the appropriate sections.

For a given placed 3D circuit, an iterative method was developed in [50] which, during each iteration, the thermal conductivities of certain FEA elements (thermal via regions) are incrementally modified so that thermal problems are reduced or eliminated. Thermal vias are generically added to elements to achieve the desired thermal conductivities. The goal of this method is to satisfy given thermal requirements using as few thermal vias as possible, i.e., keeping the thermal conductivities as low as possible.

The approach uses the finite element equations to determine a target thermal conductivity. A key observation in this survey is that the insertion of thermal vias is most useful in areas with a high *thermal gradient*, rather than areas with a high temperature. Effectively, the thermal via acts as a pipe that allows the heat to be conducted from the higher temperature region to the lower temperature region; this, in turn, leads to temperature reductions in areas of high temperature.

This is illustrated in Figure 4.5, which shows the 3D layout of the benchmark `struct`, before and after the addition of thermal vias,

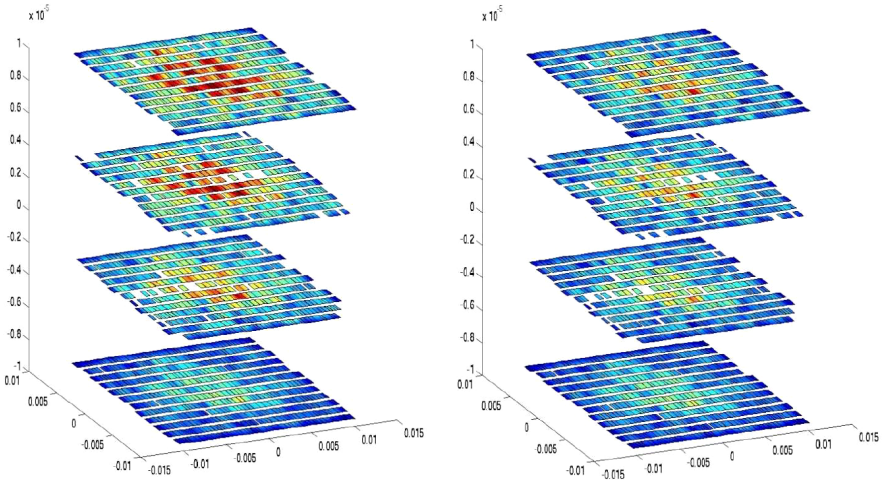


Fig. 4.5 Thermal profile of `struct` before (left) and after (right) thermal via insertion. The top four layers of the figure at right correspond to the four layers in the figure at left [51].

respectively. The hottest region is the center of the uppermost tier, and a major reason for its elevated temperature is because the tier below it is hot. Adding thermal vias to remove heat from the second tier, therefore, effectively also significantly reduces the temperature of the top tier. For this reason, the regions where the insertion of thermal vias is most effective are those that have high thermal gradients.

Therefore the method in [50] employs an iterative update formula of the type

$$K_i^{\text{new}} = K_i^{\text{old}} \left( \frac{\mid g_i^{\text{old}} \mid}{g_{i,\text{ideal}}} \right) \quad i = x, y, z, \tag{4.3}$$

where $K_i^{\text{new}}$ and $K_i^{\text{old}}$ are, respectively, the old and new thermal conductivities in each direction, before and after each iteration, $g_i^{\text{old}}$ is the old thermal gradient, and $g_{i,\text{ideal}}$ is a heuristically selected ideal thermal gradient.

Each iteration begins with a distribution of the thermal vias; this distribution is corrected using the above update formula, and the $K_i^{\text{new}}$ value is then translated to a thermal via density, and then a precise layout of thermal vias, using precharacterization. The iterations end when the desired temperature profile is achieved. This essential iterative

idea has also been used in other methods for thermal via insertion steps that are integrated within floorplanning, placement and routing, as described in succeeding sections. This general framework has been used in several other published techniques that insert thermal vias either concurrently during another optimization, or as an independent step.

### 4.2.2 3D Floorplanning

The 3D floorplanning problem attempts to find the optimal positions, and possibly, aspect ratios, for a set of large blocks in a layout. Typical cost functions include a mix of the conventional wirelength and total area costs, and the temperature and the number of inter-tier vias.

The approach in [33] presented one of the first approaches to 3D floorplanning, and used the TCG representation [85] for each tier, and a bucket structure for the third dimension. Each bucket represents a 2D region over all tiers, and stores, for each tier, the indices of the blocks that intersect that bucket. In other words, the TCG and this bucket structure can quickly determine any adjacency information. A simulated annealing engine is then utilized, with the moves corresponding to perturbations within a tier and across tiers; in each such case, the corresponding TCG(s) and buckets are updated, as necessary.

A simple thermal analysis procedure is built into this solution, using a finite difference approximation of the thermal network to build an RC thermal network. Under the assumption that heat flows purely in the $z$ direction and there is no lateral heat conduction, the RC model obtained from a finite difference approximation has a tree structure, and Elmore-delay like computations [125] can be performed to determine the temperature. The optimization heuristically attempts to make this a self-fulfilling assumption, by discouraging lateral heat conduction, introducing a cost function parameter that discourages strong horizontal gradients. A hybrid approach performs an exact thermal analysis once every 20 iterations or so and uses the approximate approach for the other iterations.

The work in [154] expands the idea of thermally driven floorplanning by integrating thermal via insertion into the simulated annealing procedure. A thermal analysis procedure based on random walks

[115] is built into the method, and an iterative formula, similar to [50], is used in a thermal via insertion step between successive simulated annealing iterations. The approach in [169] uses a technique based on force-directed methods for floorplanning, using legalization techniques to translate the continuous solution to a discrete, layered solution.

### 4.2.3 3D Placement

In the placement step, the precise positions of cells in a layout are determined, and they are arranged in rows within the tiers of the 3D circuit. Since thermal considerations are particularly important in 3D cell-based circuits, this procedure must spread the cells to achieve a reasonable temperature distribution, while also capturing traditional placement requirements [140].

Several approaches to 3D placement have been proposed in the literature. The work in [38] embeds the netlist hypergraph into the layout area. A recursive bipartitioning procedure is used to assign nodes of the hypergraph to partitions, using mincut as the primary objective and under partition capacity constraints. Partitioning in the $z$ direction corresponds to tier assignment, and $xy$ partitions to assigning standard cells to rows. No thermal considerations are taken into account.

The procedure in [49] presents a 3D-specific force-directed placer that incorporates thermal objectives directly into the placer. Instead of the finite difference method that is used in many floorplanners, this approach employs FEA, as described in Section 2.3.2. The placement engine is based on a force-directed approach: attractive forces are created between interconnected cells, and weighted together so that the constants of proportionality are chosen to be higher in the z direction to discourage inter-tier vias; repulsive forces are based on factors such as the cell overlap and thermal criteria based on the temperature gradient. An equilibrium point is found where these forces balance each other.

Once the entire system of attractive and repulsive forces is generated, repulsive forces are added, the system is solved for the minimum energy state, i.e., the equilibrium location. Ideally, this minimizes the wire lengths while at the same time satisfying the other design criteria

such as the temperature distribution. The iterative force-directed approach follows the following steps in the main loop. Initially, forces are updated based on the previous placement. Using these new forces, the cell positions are then calculated. These two steps of calculating forces and finding cell positions are repeated until the exit criteria are satisfied. The specifics of the force-directed approach to thermal placement, including the mathematical details, are presented in [49]. Once the iterations converge, a final postprocessing step is used to legalize the placement. Even though forces have been added to discourage overlaps, the force-directed engine solves the problem in the continuous domain, and the task of legalization is to align cells to tiers, and to rows within each tier.

Another method in [32] maps an existing 2D placement to a 3D placement through transformations based on dividing the layout into $2^k$ regions, for integer values of $k$, and then defining local transformations to heuristically refine the layout.

More recent work in [52] observes that since 3D layouts have very limited flexibility in the third dimension (with a small number of layers and a fixed set of discrete locations), partitioning works better than a force-directed method. Accordingly, this work performs global placement using recursive bisectioning. Thermal effects are incorporated through *thermal resistance reduction nets*, which are attractive forces that induce high power nets to remain close to the heat sink. The global placement step is followed by coarse legalization, in which a novel cell-shifting approach is proposed. This generalizes the methods in FastPlace [147] by allowing shift moves to adjust the boundaries of both sparsely and densely populated cells using a computationally simple method. Finally, detailed legalization generates a final nonoverlapping layout. The approach is shown to provide excellent tradeoffs between parameters such as the number of interlayer vias, wire length, temperature.

### 4.2.4 Routing Algorithms

During routing, several objectives and constraints must be taken into consideration, including avoiding blockages due to areas occupied by

thermal vias, incorporating the effect of temperature on the delays of the routed wires, and of course, traditional objectives such as wire length, timing, congestion, and routing completion. In the 2D arena, results on thermally aware routing have been reported, for example, in [3, 4].

In the 3D context, once the cells have been placed and the locations of the thermal vias determined, the routing stage finds the optimal interconnections between the wires. As in 2D routing, it is important to optimize the wire length, the delay, and the congestion. In addition, several 3D-specific issues come into play. First, the delay of a wire increases with its temperature, so that more critical wires should avoid the hottest regions, as far as possible. Second, inter-tier vias are a valuable resource that must be optimally allocated among the nets. Third, congestion management and blockage avoidance is more complex with the addition of a third dimension. For instance, a signal via or thermal via that spans two or more tiers constitutes a blockage that wires must navigate around.

Consider the problem of routing in a three-tier technology, as illustrated in Figure 4.6. The layout is gridded into rectangular tiles, each with a horizontal and vertical capacity that determines the number of wires that can traverse the tile, and an inter-tier via capacity that determines the number of free vias available in that tile. These capacities account for the resources allocated for nonsignal wires (e.g., power and clock wires) as well as the resources used by thermal vias. For a single net, as shown in the figure, the degrees of freedom that are available are in choosing the locations of the inter-tier vias, and selecting

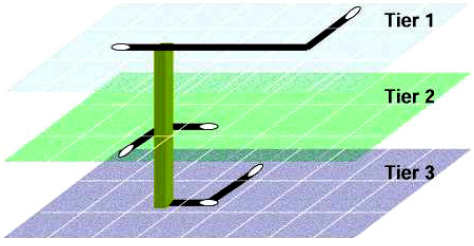



Fig. 4.6 An example route for a net in a three-tier 3D technology [2].

the precise routes within each tier. The locations of inter-tier vias will depend on the resource contention for vias within each grid. Moreover, critical wires should avoid the high-temperature tiles, as far as possible.

The work in [35] presents a thermally conscious router, using a multilevel routing paradigm similar to [30, 34], with integrated inter-tier via planning and incorporating thermal considerations. An initial routing solution is constructed by building a 3D minimum spanning tree (MST) for each multipin net, and using maze routing to avoid obstacles.

At each level of the multilevel scheme, the inter-tier via planning problem assigns vias in a given region at level $k-1$ of the multilevel hierarchy to tiles at level $k$. The problem is formulated as a min-cost maxflow problem, which has the form of a transportation problem. The flow graph is constructed as follows:

- The source node of the flow graph is connected through directed edges to a set of nodes $v_i$, representing candidate thermal vias; the edges have capacity 1 and cost 0.
- Directed edges connect a second set of nodes, $T_j$, from each tile to the sink node, with capacity equaling the number of vias that the tile can contain, and cost zero. The capacity is computed using a heuristic approach that takes into account the temperature difference between the tile and the one directly in the tier below it (under the assumption that heat flows downwards toward the sink); the thermal analysis is based on a commercial FEA solver.
- The source and sink both have cost $m$, which equals the number of inter-tier vias in the entire region.
- Finally, a node $v_i$ is connected to a tile $T_j$ through an arc with infinite capacity and cost equaling the estimated wirelength of assigning an inter-tier via $v_i$ to tile $T_j$.

Another approach to 3D routing, presented in [168], combines the problem of 3D routing with heat removal by inserting thermal vias in the $z$ direction, and introduces the concept of *thermal wires*. Like a thermal via, a thermal wire is a dummy object: it has no electrical

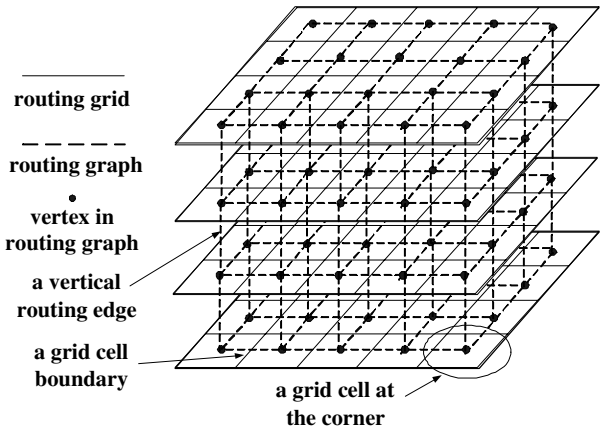



Fig. 4.7 Routing grid and routing graph for a four-tier 3D circuit [168].

function, but is used to spread heat in the lateral direction. Each tier is tiled into a set of regions, as shown in Figure 4.7.

The global routing scheme goes through two phases. In Phase I, an initial routing solution is constructed. A 3D MST is built for each multipin net, and based on the corresponding two-pin decomposition, the routing congestion is statistically estimated over each lateral routing edge using the method in [153]. This congestion model is extended to 3D by assuming that a two-pin net with pins on different tiers has an equal probability of utilizing any inter-tier via position within the bounding box defined by the pins.

A recursive bipartitioning scheme is then used to assign inter-tier vias. This is also formulated as a transportation problem, but the formulation is different from the multilevel method described above. Signal inter-tier via assignment is then performed across the cut in each recursive bipartition. Figure 4.8(a) shows an example of signal inter-tier via assignment for a decomposed two-pin signal net in a four-tier circuit with two levels of hierarchy. The signal inter-tier via assignment is first performed at the boundary between regions of group 0 and group 1 at topmost level, and then it is processed for tier boundary within each group. At each level of the hierarchy, the problem of signal inter-tier via assignment is formulated as a min-cost network flow.

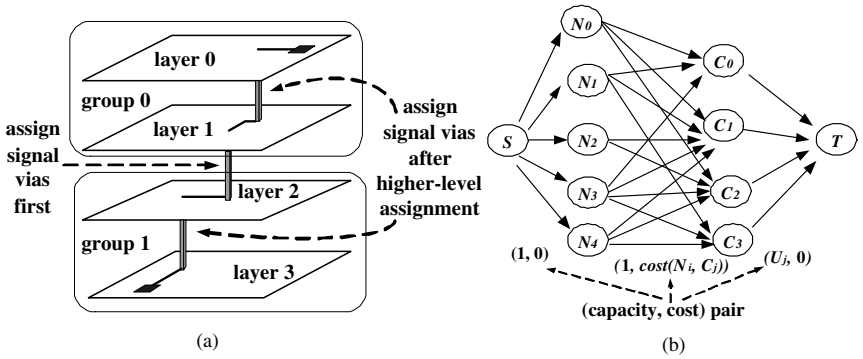



Fig. 4.8 (a) Example of hierarchical signal via assignment for a four-tier circuit. (b) Example of min-cost network flow heuristics to solve signal via assignment problem at each level of hierarchy [168].

Figure 4.8(b) shows the network flow graph for assigning signal inter-tier vias of five inter-tier nets to four possible inter-tier via positions. The idea is to assign each net that crosses the cut to an inter-tier via. Each inter-tier net is represented by a node $N_i$ in the network flow graph; each possible inter-tier via position is indicated by a node $C_j$. If $C_j$ is within the bounding box of the two-pin inter-tier net $N_i$, we build a directed edge from $N_i$ to $C_j$, and set the capacity to be 1, the cost of the edge to be cost$(N_i, C_j)$. The cost$(N_i, C_j)$ is evaluated as the shortest path cost for assigning inter-tier via position $C_j$ to net $N_i$ when both pins of $N_i$ are on the two neighboring tiers; otherwise it is evaluated as the average shortest path cost over all possible unassigned signal inter-tier via positions in lower levels of the hierarchy. The shortest path cost is obtained with Dijkstra's algorithm in the 2D congestion map generated from the previous estimation step, and the cost function for crossing a lateral routing edge is a combination of edge length and an overflow cost function similar to that in [55]. The supply at the source, equaling the demand at the sinks, is $N$, the number of nets.

Finally, once the inter-tier vias are fixed, the problem reduces to a 2D routing problem in each tier, and maze routing is used to route the design.

Next, in Phase II, a linear programming approach is used to assign thermal vias and thermal wires. A thermal analysis is performed, and

fast sensitivity analysis is carried out using the adjoint network method, which has the cost of a single thermal simulation. The benefit of adding thermal vias, for relatively small perturbations in the via density, is given by a product of the sensitivity and the via density, a linear function. The objective function is a sum of via densities and is also linear. Additional constraints are added in the formulation to permit overflows, and a sequence of linear programs is solved to arrive at the solution.

## 4.3 Recovering from Runtime Shifts

In this section, we will discuss examples of techniques that can be used to recover from the degradation caused by thermal effects. Specifically, we will begin with a description of methods that use adaptive body biases to recover performance and leakage power, then overview adaptive supply voltage based methods, and finally, discuss thermal guardbanding methods to overcome aging effects, using NBTI degradation as an example. The adaptive techniques, in particular, are examples of methods that can dynamically change the behavior of the circuit based on information provided by on-chip sensors. On the other hand, guardbanding methods are purely static design-time approaches that are used to build sufficient margins to ensure that the circuit functions correctly in the presence of stress.

### 4.3.1 Using Adaptive Body Biases

#### 4.3.1.1 Overview

The fourth terminal of a transistor, after the source, drain and gate, is the body, and it is typically tied to ground for an NMOS device and the supply for a PMOS device. Body biasing alters the voltage on this terminal to achieve threshold voltage modulation, providing an effective knob to alter the delay and leakage of the circuit. Typically, the body of each transistor is tied to its corresponding well (or substrate), implying that the same altered body bias must be applied to all transistors that share the well.

Traditionally, this method has been used in two different operational scenarios [93]. The first method, known as static body biasing

uses reverse body biasing when the microprocessor is in a stand-by state, with the aim of reducing the subthreshold leakage current. Algorithms to determine the optimal configuration that achieves the lowest leakage in the presence of latency constraints, have been described in [8, 90, 160, 161, 162]. Such schemes have been used in low power and embedded systems, where leakage power minimization is the main objective. The second scheme, known as adaptive body bias (ABB), involves recovering dies impacted by process variations through post-silicon tuning. Adaptive body bias is a dynamic control technique, used to tighten the distribution of the maximum operational frequency and the maximum leakage power, in the presence of within-die (WID) thermal or process variations. It was proposed in [152] and was further explored [80] during the design of a DSP processor. The main goal of this scheme is to ensure that maximum number of dies operate in the highest frequency bin, thereby increasing the yield of the fabrication process [141, 142].

Bidirectional adaptive body bias has been shown to reduce the impact of D2D and WID parameter variations on microprocessor frequency and leakage in [25, 141, 142, 144, 145]. Typically, devices that are slow but do not leak too much can be Forward Body Biased (FBB) to improve the speed, whereas devices that are fast and leaky can be Reverse Body Biased (RBB) to meet the leakage budget. The work in [97, 142] performs process variation-based ABB, and divides the die into a set of WID-variational regions. In each region, test structures that are replicas of the critical path, are built. The delay and leakage values of these test structures are measured, and are used to determine the exact body bias values that are required to counter process variations at room temperature. The application of a WID–ABB technique for one-time compensation during the test phase, in [142], shows that 100% of the dies can be salvaged, while 99% of them operate at frequencies within the fastest bin.

#### 4.3.1.2 Implementing ABB

If the foundry is capable of supporting a triple well process, the N-well(s) and the P-well(s) can be independently biased. Techniques for

clustering gates, into wells that can be separately biased, are presented in [76]. The body bias to be applied is denoted by the tuple, $(v_{bn}, v_{bp})$ where in general, each element of the tuple can be a vector, corresponding to wells that can be separately biased. Note that the actual voltage applied to the body of the PMOS transistors is $(V_{dd} - v_{bp})$, where $V_{dd}$ is the supply voltage. The range of operating temperatures, and the extent of process variations, over which thermal variations can be compensated for, each depends on the minimum and maximum limits imposed on the body bias voltages, due to device physics restrictions. It is important to note that if the applied body bias is too large, it may exceed the cut-in voltage of $p$–$n$ junctions associated with the structure of a transistor. Additionally, the maximum amount of body bias is also constrained by the permissible leakage budget of the circuit block, since FBB reduces the delay at the expense of an increase in the leakage. The exact resolution of bias voltages is primarily determined by constraints on generating and routing these voltages to every biasable well in the circuit.

The control mechanism necessary to ensure that the requisite voltages are selected can either be built using a critical path replica based control system or a look-up table based control system. The hardware on-chip control set-up, as built in [93, 142], requires a test structure that replicates the critical path (assuming there is only one such path), which is expected to accurately reflect the behavior of the entire circuit, and the impact on delay and leakage due to on-chip variations. The control circuit consists of a delay monitor, phase comparator, decoder, digital–analog converter (DAC), and such other precision hardware to automatically select the bias pair, $(v_{bn}, v_{bp})$. Although such schemes are self-adapting, and require minimal post-silicon testing, a few sample critical path replicas might be unable to reflect the exact nature of process and thermal variations on the actual circuit, which consists of millions of paths. Experimental results in [142] indicate that a minimum of 14 critical path replicas per test-chip are required to accurately determine the die frequency of microprocessors, for a 130 nm based process. The increased impact of process variations in sub-100 nm technologies is likely to require a larger number of critical path replicas to be fabricated per test-chip to ensure a high level of confidence in the frequency

measurements for a 65 nm or a 45 nm based design. This may lead to a substantial area overhead. Further, if the test circuits are large, they measure their own variations, which may not be the same as that of the actual circuit. Thus, the additional area overhead imposed by the number of critical path replicas and their inaccuracies, coupled with the need for PVT (process, voltage, and temperature) invariant hardware, call for better control mechanisms.

A viable alternative to the critical path replica based control system is the look-up table based control system. In this case, every block is equipped with a look-up table [93, 101] that can store the bias values $(v_{bn}, v_{bp})$. These are the precomputed optimal values that can compensate for thermal and process parametric variations. Each entry in the look-up table corresponds to a different temperature point. These entries are calibrated off-line through post-silicon measurements, with the aid of an efficient algorithm, i.e., using software. The look-up table is assumed to be built using a simple ROM like structure, and is populated during post-silicon testing. When the circuit is in operation, the entries in the look-up table are keyed based on the operating temperature, which is measured by a temperature sensor, as shown in [101]. The output of the table is fed to the body bias network to generate and route the appropriate voltages, thereby providing run-time compensation.

The look-up table based control system eliminates the various issues associated with using critical path replicas as test structures, to capture the effect of process and thermal variations, on the entire chip. Since the body bias voltages are already precomputed, they may be immediately applied to the entire chip, to compensate for on-chip temperature variations, without affecting the run-time operation. An overall architectural implementation of this control scheme is explained in the next subsection.

Further, the effect of voltage variations, as well as aging, can be incorporated by adding appropriate sensors, and introducing an additional entry, i.e., supply voltage $(V_{dd})$, along with $v_{bn}$ and $v_{bp}$, to the look-up table. The algorithms can be modified accordingly, to determine the optimal body bias and supply voltage configuration, to overcome the effects of process and thermal variations, and

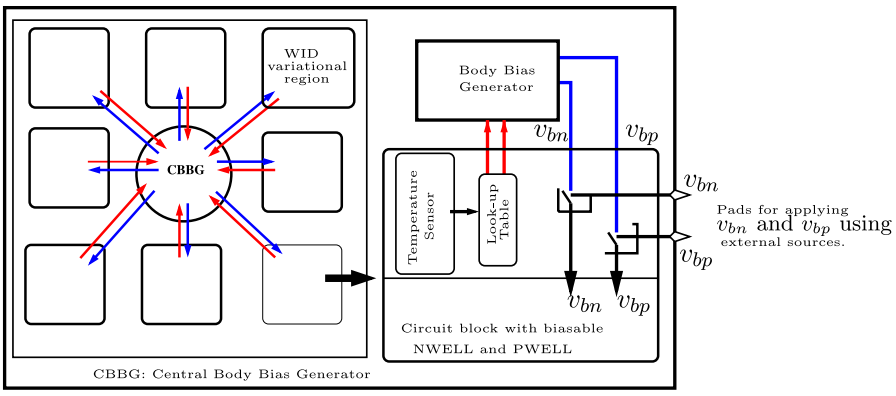



Fig. 4.9 A generic ABB implementation architecture showing the structure of the WID-variational regions.

temporal degradation. A practical example of a system that uses the above scheme, and compensates for PVT variations, as well as aging, is seen in a 90 nm-based design in [143].

An implementation for the look-up table control scheme based body bias compensation network is shown in Figure 4.9. The chip is partitioned into several WID-variational regions, each of which must be compensated independently. The implementation assumes a central body bias network capable of generating the requisite voltage to each block. Alternatively, each block may have its own body bias generation and distribution network. Each WID-variational region is equipped with a temperature sensor that is capable of tracking variations in on-chip operating temperature. The temperature sensor references a ROM, that stores the ($v_{bn}$, $v_{bp}$) values for each compensating temperature, in the form of a look-up table. The output of the look-up table feeds the central (or local) body bias generator, and accordingly generates the required voltages. These voltages are then routed to the corresponding N and P wells. The NMOS and PMOS body bias voltages may be applied by external sources during testing. Once the final voltages are determined, and the look-up table has been populated, the switches can be closed and the requisite voltages required for compensation are supplied from the on-chip body bias generation network.

#### 4.3.1.3 Using ABB for Thermal Compensation

ABB has been used to compensate for process [142, 144, 145] as well as temperature [78, 101] variations. The work in [78] applies a combination of temperature-based ABB, and a process-based ABB to permit the circuit to recover from changes due to both temperature and process variations. In order to be able to adaptively body bias the dies at all operating temperatures, an efficient self-adjusting mechanism is employed, which can sense the operating temperature, and thereby dynamically regulate the voltages that must be applied to the body of the devices to meet the performance constraints.

Computational techniques for efficiently determining the exact amount of bias required to achieve process and temperature compensation, to populate the look-up table such that the time spent on the tester is minimized, are proposed in [78]. Two methods for computing the final body bias values are proposed: the PTABB (Process and Temperature Adaptive Body Bias) algorithm and the PABB–TABB (Process Adaptive Body Bias–Temperature Adaptive Body Bias) algorithm. Both methods use mathematical models to express the delay and leakage as functions of the NMOS and the PMOS transistor body bias voltages. A two variable nonlinear optimization problem is formulated and an optimizer is used to determine the configuration that meets the delay requirement, and thereby minimizes the overall leakage.

While the PTABB algorithm involves measuring the delay and leakage at sample points for each individual die or WID-variational region, at each compensating temperature, the PABB–TABB algorithm involves measurements only at the nominal operating temperature. The PABB–TABB algorithm splits the original problem into two sub-problems, namely compensating for process variations at nominal temperature (PABB), and compensating for thermal variations under ideal process conditions (TABB). The final set of bias voltages is simply a combination of the PABB and TABB voltages. Thus, this scheme minimizes the number of tester measurements, and eliminates the need to test at each operating temperature. Experimental results on a 65 nm and a 45 nm technology demonstrate the ability of the PTABB and the PABB–TABB algorithms to closely predict the body bias voltages.

### 4.3.2 Using Adaptive Supply Voltages and Frequencies

As in the case of body biases, adaptive supply voltages can be used to recover performance in the presence of thermal variations. The key observation here is that elevated temperatures are caused by increased power dissipation, and the quadratic dependence between the dynamic power and the supply voltage, $V_{dd}$, implies that changing the value of $V_{dd}$ is an effective knob for controlling on-chip thermal effects. Similarly, slowing down the processor by reducing the chip frequency also provides reductions in the power dissipation, and hence the temperature. Cumulatively, scaling both the supply voltage and frequency allows a potentially cubic reduction in power.

The idea of throttling the processor was proposed in [123] for a PowerPC processor by adaptively altering its frequency, based on feedback from a thermal assist unit (TAU) that monitors junction temperatures of devices on the chip, using a temperature sensor that uses a differential scheme, based on the voltage across a pair of diodes. The TAU compares the temperature against one or two user-programmed thresholds, and generates a thermal management interrupt if the threshold is exceeded. The approaches in [24, 144] show how sensor-based approaches may be used to provide adaptive feedback to alter the supply voltage in response to thermal changes. A control-theoretic framework for this purpose is described in [133], where a simple proportional controller is used to update the supply voltage.

Voltage and frequency scaling techniques have also been incorporated into the design of an Itanium-family processor, codenamed the Montecito, as described in [44, 92, 107], and illustrated in Figure 4.10. Four on-die thermal sensors are used, and are sampled at low frequencies (with periods of the order of 20 ms). The voltage regulator provides three possible supply voltages to the processor, one to the core, one to the cache, and a fixed, nonvarying voltage for special circuitry. The closed loop thermal management control system senses the dissipated power, and generates an error signal that shows the difference between the actual power and the power limit. This signal goes through an infinite impulse response (IIR) filter, and then to saturating modules that ensure that the supply voltage remains with the desired range of

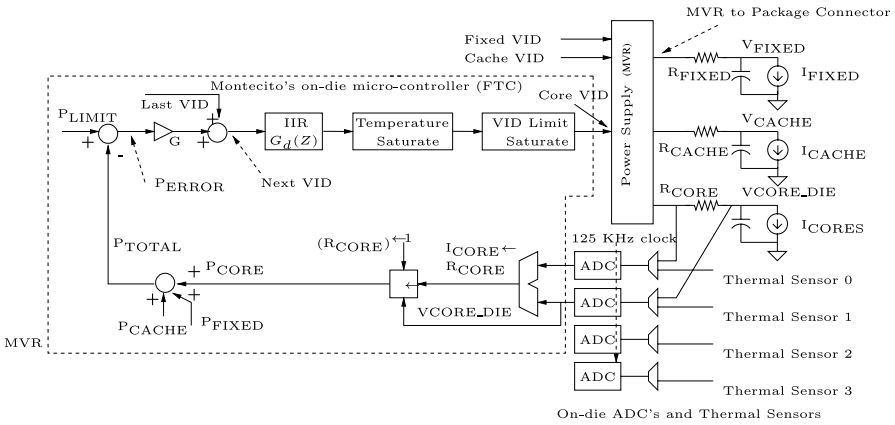



Fig. 4.10 Control system for a 90 nm Itanium processor, taking in inputs from thermal sensors, and adaptively changing the processor supply voltage and frequency [92].

0.9–1.25 V, and generates a signal that determines the voltage to be generated by the regulator. The variable frequency clock system reacts to changes in the voltage and adapts the frequency to be appropriate for the applied voltage. The control system is designed to have a transient response of 250 $\mu$s.

An approach to solving this problem for embedded applications was presented in [167]. Since the functionalities for embedded applications are well characterized, it is possible to perform static scheduling at design time in this domain. The work assumes a periodic sequence of tasks to be scheduled, and uses a polynomial-time approximation algorithm to solve an offline dynamic voltage and frequency scaling (DVFS) formulation.

### 4.3.3 NBTI Guardbanding

The physical explanations for the NBTI mechanism, which correspond to various models [5, 6, 7, 12, 19, 77, 146], are unanimous in agreeing that the PMOS transistor threshold voltage rises logarithmically with time. This rise could lead to an increase of up to 25%–30% in the value of the threshold voltage after 10 years, depending on the process. The extent of threshold voltage degradation is strongly dependent on the amount of time for which the device has been stressed and relaxed,

since the generation of traps under negative bias stress is followed by annealing of the traps during recovery, (i.e., when the stress is relaxed). Circuit simulations using these models have shown that NBTI causes the delay of digital circuits to worsen by about 10% [77, 105, 106].

A general solution to maintaining optimal performance under the influence of NBTI has been to reduce the delay of the critical paths through the use of gate sizing [106, 146]. The work in [106] formulates a nonlinear optimization problem to determine the optimal set of gate sizes required to ensure that the circuit runs at its delay specification, after 10 years of operation. The work is based on a model for NBTI, that ignores the effect of recovery, in computing the threshold voltage degradation. The model cumulatively adds the time for which the gates are stressed during their entire lifetime, and estimates the threshold voltage degradation, assuming that the gates are continuously stressed for that duration. Hence, their results show that the increase in the circuit area is rather weakly dependent on the signal probabilities of the nodes, and assuming that all gates in the circuit are always NBTI affected (worst case design) does not significantly affect the final solution. The authors consider the gate sizes to be continuous, and show that an increase in area of about 8.7%, as compared to a design that ignores NBTI effects, is required to meet the target delay.

The above idea can be readily used in other transforms, such as technology mapping, by replacing the nominal value of the delays of the gates in the standard cell library, with the delay under worst case NBTI. The target frequency is given to the synthesis tool, and technology mapping can be performed using these NBTI-affected library cells to produce a circuit, which is structurally different from that obtained using the sizing algorithm in [106], but is functionally equivalent, and meets the timing requirement.

However, the worst-case assumption that ignores the effect of signal probabilities on the delay can be pessimistic, since the annealing or healing process on the removal of NBTI stress can partially reverse the threshold voltage degradation due to NBTI. This happens frequently in a circuit: for example, when the input signal to a CMOS inverter changes from logic 0 to logic 1, the $V_{gs}$ stress is relaxed from $-V_{dd}$ to zero. The recovery in threshold voltage on removing the applied

stress, can be explained by physical mechanisms related to annealing of interface traps, and reformation of $Si$–$H$ bonds. Experiments in [5, 19], and subsequently the models in [12, 77, 146], have shown that considering the effect of annealing and recovery has a significant bearing on the overall NBTI impact.

The work in [79] considers the effect of NBTI, including recovery effects, during the technology mapping process. Since a digital circuit consists of millions of nodes with differing signal probabilities, it is essential to estimate the delay of the gates in the library based on their input node signal probabilities, and use these delay values during technology mapping. The mapping process can be used to "hide" high probability nodes within a gate, and reduce the potential for NBTI degradation. The essential idea of the approach is to modify the process of technology mapping, using the signal probability (defined as the probability that the signal is low), denoted by SP, of the nodes in the circuit. The SP values of the primary inputs are determined, based on RTL-level simulations and statistical estimates. The SP values at every other node in the subject graph are calculated accordingly, and this information is used to choose the best gate to meet the timing at each node. Experimental results indicate that an average 10% recovery in area can be obtained using the SP-based method, as opposed to the worst case NBTI based method.

# 5

# Conclusion

The objective of this survey has been to provide a meaningful overview of techniques for thermal analysis and electrothermal optimization. It should be noted that the predecessors of the work described herein lie in early work on transistor level thermal simulation, thermal simulation of microwave ICs, package-level simulation, etc., but to maintain the focus of this survey, we have restricted our descriptions to the on-chip context.

Techniques for on-chip simulation are now available at varying levels of accuracy and computational efficiency, as described in the chapter on *Thermal Analysis Optimizations*. These may apply to thermal simulation and optimization in much the same way that circuit simulators or analyzers are used: the more accurate and detailed versions are employed in verification, while the more approximate approaches are adequate for optimization, as long as they show sufficient fidelity. Electrothermal analysis techniques, which closely couple thermal analysis with circuit performance, are extremely useful in on-chip thermal optimization. These methods must include mechanisms that ensure self-consistency, so that the impact of electrical effects on thermal effects, and vice versa, are taken into consideration.

Finally, thermal optimization and mitigation form a key part of any thermally conscious design strategy. To the extent possible, it is desirable to control the on-chip temperature and prevent it from rising, in order to control both near-term degradations in performance metrics such as power and delay, due to elevated temperatures, as well as long-term reliability-related problems. At the point where temperature levels can no longer be controlled, or where control comes at an unacceptable cost, it is essential to mitigate any effects through adaptive approaches, or through guardbanding.

# Acknowledgments

The authors gratefully acknowledge the role of their past collaborations with (alphabetically) Charlie Chung-Ping Chen, Brent Goplen, David J. Lilja, Sani R. Nassif, Vidyasagar Nookala, Haifeng Qian, and Tianpei Zhang.

# References


[1] "HotSpot," available at: http://lava.cs.virginia.edu/HotSpot/index.htm.

[2] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, and S. Sapatnekar, "Placement and routing in 3D integrated circuits," *IEEE Design and Test*, vol. 22, no. 6, pp. 520–531, November–December 2005.

[3] A. H. Ajami, K. Banerjee, and M. Pedram, "Analysis of substrate thermal gradient effects on optimal buffer insertion," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 44–48, 2001.

[4] A. H. Ajami, K. Banerjee, and M. Pedram, "Modeling and analysis of nonuniform substrate temperature effects on global ULSI interconnects," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 24, no. 6, pp. 849–861, June 2005.

[5] M. A. Alam, "A critical examination of the mechanics of dynamic NBTI for pMOSFETs," in *IEEE International Electronic Devices Meeting*, pp. 14.4.1–14.4.4, 2003.

[6] M. A. Alam, "On the reliability of micro-electronic devices: An introductory lecture on negative bias temperature instability," Nanotechnology 501 Lecture Series; available at http://www.nanohub.org/resources/?id=193, 2005.

[7] M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," *Journal of Microelectronics Reliability*, vol. 45, pp. 71–81, August 2004.

[8] A. Andrei, M. Schmitz, P. Eles, Z. Peng, and B. M. Al-Hashimi, "Overhead-conscious voltage selection for dynamic and leakage energy reduction of time-constrained systems," in *Proceedings of the Design, Automation and Test in Europe*, pp. 518–523, 2004.

[9] K. Banerjee and A. Mehrotra, "Global (interconnect) warming," *IEEE Circuits and Devices*, pp. 16–32, September 2001.

[10] P. Bannon, "Alpha 21364: A scalable single-chip SMP," available at: http://www.digital.com/alphaoem/micro- processorforum.htm, 1998.

[11] W. Batty, C. E. Christoffersen, A. J. Panks, S. David, C. M. Snowden, and M. B. Steer, "Electrothermal CAD of power devices and circuits with fully physical time-dependent compact thermal modeling of complex nonlinear 3-D systems," *IEEE Transactions on Components and Packaging Technologies*, vol. 34, no. 4, pp. 566–590, December 2001.

[12] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 189–192, 2006.

[13] J. R. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," in *Proceedings of the IEEE*, vol. 57, pp. 1587–1594, September 1969.

[14] K. Bowman, L. Wang, X. Tang, and J. Meindl, "A circuit-level perspective of the optimum gate oxide thickness," *IEEE Transactions on Electron Devices*, vol. 48, no. 8, pp. 1800–1810, August 2001.

[15] W. L. Briggs, "A multigrid tutorial," http://www.llnl.gov/CASC/people/henson/mgtut/ps/mgtut.pdf.

[16] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimization," in *Proceedings of the ACM International Symposium on Computer Architecture*, pp. 83–94, 2000.

[17] D. C. Burger and T. M. Austin, "The SimpleScalar tool set, version 2.0," Technical Report CS-TR-97-1342, The University of Wisconsin, Madison, June 1997.

[18] J. Burns, L. McIlrath, J. Hopwood, C. Keast, D. P. Vu, K.Warner, and P. Wyatt, "An SOI-based three dimensional integrated circuit technology," in *Proceedings of the IEEE International SOI Conference*, pp. 20–21, 2000.

[19] S. Chakravarthi, A. T. Krishnan, V. Reddy, C. Machala, and S. Krishnan, "A comprehensive framework for predictive modeling of negative bias temperature instability," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 273–282, 2004.

[20] H. Chang and S. S. Sapatnekar, "Prediction of leakage power under process uncertainties," in *Proceedings of the ACM Transactions on Design Automation of Electronic Systems*, vol. 12, no. 2, April 2007.

[21] G. Chen, K. Y. Chuah, M. F. Li, D. S. H. Chan, C. H. Ang, J. Z. Cheng, Y. Jin, and D. L. Kwong, "Dynamic NBTI of PMOS transistors and its impact on device lifetime," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 196–200, 2003.

[22] G. Chen, M. F. Li, C. H. Ang, J. Z. Zheng, and D. L. Kwong, "Dynamic NBTI of p-MOS transistors and its impact on MOSFET scaling," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 196–202, 2003.

[23] G. Chen and S. S. Sapatnekar, "Partition-driven standard cell placement," in *Proceedings of the International Symposium on Physical Design*, pp. 75–80, 2003.

[24] Q. Chen, M. Meterelliyoz, and K. Roy, "A CMOS thermal sensor and its applications in temperature adaptive design," in *Proceedings of the IEEE International Symposium on Quality Electronic Design*, 2006.

[25] T. Chen and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *IEEE Transactions on VLSI Systems*, vol. 11, no. 5, pp. 888–899, October 2003.

[26] T.-Y. Chiang, K. Banerjee, and K. C. Saraswat, "Analytical thermal model for multilevel VLSI interconnects incorporating via effect," *IEEE Electron Device Letters*, vol. 23, no. 1, pp. 31–33, January 2002.

[27] R. C. Chu, R. E. Simons, and G. M. Chrysler, "Experimental investigation of an enhanced thermosyphon heat loop for cooling a high performance electronics module," in *Proceedings of the IEEE Semiconductor Thermal Measurement and Management Symposium* (*Semitherm*), pp. 1–9, 1999.

[28] L. O. Chua and P.-M. Lin, *Computed-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[29] P. Cocchini, "Concurrent flip-flop and repeater insertion for high performance integrated circuits," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 268–273, 2002.

[30] J. Cong, J. Fang, and Y. Zhang, "Multilevel approach to full-chip gridless routing," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 234–241, 2001.

[31] J. Cong, A. Jagannathan, G. Reinman, and M. Romesis, "Microarchitecture evaluation with physical planning," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 32–35, 2003.

[32] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 780–785, 2007.

[33] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proceedings of the International Symposium on Physical Design*, pp. 306–313, 2004.

[34] J. Cong, M. Xie, and Y. Zhang, "An enhanced multilevel routing system," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 51–58, 2002.

[35] J. Cong and Y. Zhang, "Thermal-driven multilevel routing for 3-D ICs," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 121–126, 2005.

[36] J. P. Costa, M. Chou, and L. M. Silveira, "Efficient techniques for accurate modeling and simulation of substrate coupling in mixed-signal IC's," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 18, no. 5, pp. 597–607, May 1999.

[37] H. Dadgour, S.-C. Lin, and K. Banerjee, "A statistical framework for estimation of full-chip leakage-power distribution under parameter variations," *IEEE Transactions on Electron Devices*, vol. 54, no. 11, pp. 2930–2945, November 2007.

[38] S. Das, A. Chandrakasan, and R. Reif, "Design tools for 3-D integrated circuits," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 53–56, 2003.

[39] A. Dasdan and I. Hom, "Handling inverted temperature dependence in static timing analysis," in *Proceedings of the ACM Transactions on Design Automation of Electronic Systems*, vol. 11, no. 2, pp. 306–324, April 2006.

[40] F. M. d'Heurle, "Electromigration and failure in electronics: An introduction," in *Proceedings of the IEEE*, vol. 59, pp. 1409–1418, October 1971.

[41] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*. Washington, DC: Mathematical Association of America, 1984.

[42] M. Ekpanyapong, J. R. Minz, T. Watewai, H.-H. S. Lee, and S. K. Lim, "Profile-guided microarchitectural floorplanning for deep submicron processor design," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 634–639, 2004.

[43] M. Ershov, R. Lindley, S. Saxena, A. Shibkov, S. Minehane, J. Babcock, S. Winters, H. Karbasi, T. Yamashita, P. Clifton, and M. Redford, "Transient effects and characterization methodology of negative bias temperature instability in pMOS transistors," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 606–607, 2003.

[44] T. Fischer, F. Anderson, B. Patella, and S. Naffziger, "A 90 nm variable-frequency clock system for a power-managed itanium®-family processor," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 294–299, 599, 2005.

[45] V. Gerousis, "Design and modeling challenges for 90 nm and 50 nm," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 353–360, 2003.

[46] R. Gharpurey and R. G. Meyer, "Modeling and analysis of substrate coupling in integrated circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 344–353, March 1996.

[47] G. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: John Hopkins University Press, third ed., 1996.

[48] B. Goplen, *Advanced Placement Techniques for Future VLSI Circuits*. PhD thesis, Minneapolis, MN: University of Minnesota, 2006.

[49] B. Goplen and S. S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 86–89, November 2003.

[50] B. Goplen and S. S. Sapatnekar, "Thermal via placement in 3D ICs," in *Proceedings of the International Symposium on Physical Design*, pp. 167–174, 2005.

[51] B. Goplen and S. S. Sapatnekar, "Placement of thermal vias in 3-D ICs using various thermal objectives," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 26, no. 4, pp. 692–709, April 2006.

[52] B. Goplen and S. S. Sapatnekar, "Placement of 3D ICs with thermal and interlayer via considerations," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 626–631, 2007.

[53] T. Grasser, W. Gos, V. Sverdlov, and B. Kaczer, "The universality of NBTI relaxation and its implications for modeling and characterization," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 268–280, 2007.

[54] K. W. Guarini, A. W. Topol, M. Leong, R. Yu, L. Shi, M. R. Newport, D. J. Frank, D. V. Singh, G. M. Cohen, S. V. Nitta, D. C. Boyd, P. A. O'Neil, S. L. Tempest, H. B. Pogpe, S. Purushotharnan, and W. E. Haensch, "Electrical integrity of state-of-the-art 0.13 $\mu$m SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication," in *IEEE International Electronic Devices Meeting*, pp. 943–945, 2002.

[55] R. T. Hadsell and P. H. Madden, "Improved global routing through congestion estimation," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 28–34, 2003.

[56] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in *Second Workshop on Temperature-Aware Computing Systems*, 2005.

[57] R. Hannemann, "Thermal control of electronics: Perspectives and prospects," available at http://hdl.handle.net/1721.1/7315.

[58] S. Hassoun, C. J. Alpert, and M. Thiagarajan, "Optimal buffered routing path constructions for single and multiple clock domain systems," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 247–253, 2002.

[59] M. Healy, M. Vittes, M. Ekpanyapong, C. Ballapuram, S. K. Lim, H.-H. S. Lee, and G. H. Loh, "Microarchitectural floorplanning under performance and thermal tradeoff," in *Proceedings of the Design, Automation and Test in Europe*, pp. 1–6, 2006.

[60] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Transactions on VLSI Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[61] W. Huang, K. Sankaranarayanan, R. J. Ribando, M. R. Stan, and K. Skadron, "An improved block-based thermal model in HotSpot 4.0 with granularity considerations," Tech. Rep. CS-2007-07, Department of Computer Science, University of Virginia, Charlottesville, VA, 2007.

[62] W. Huang, K. Sankaranarayanan, R. J. Ribando, M. R. Stan, and K. Skadron, "An improved block-based thermal model in HotSpot 4.0 with granularity considerations," in *Proceedings of the Workshop on Duplicating, Deconstructing, and Debunking*, 2007.

[63] W. Huang, M. R. Stan, and K. Skadron, "Parameterized physical compact thermal modeling," *IEEE Transactions on Components and Packaging Technologies*, vol. 28, no. 4, pp. 615–622, December 2005.

[64] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 878–883, 2004.

[65] V. Huard, M. Denais, and C. Parthasarathy, "NBTI degradation: From physical mechanisms to modeling," *Journal of Microelectronics Reliability*, vol. 46, pp. 1–23, January 2006.

[66] V. Huard, C. R. Parthasarathy, C. Guerin, and M. Denais, "Physical modeling of negative bias temperature instabilities for predictive exploration," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 733–734, 2006.

[67] A. E. Islam, H. Kufluoglu, D. Varghese, and M. A. Alam, "Critical analysis of short-term negative bias temperature instability measurements: Explaining the effect of time-zero delay for on-the-fly measurements," *Applied Physics Letters*, vol. 90, February 2007.

[68] A. E. Islam, H. Kufluoglu, D. Varghese, S. Mahapatra, and M. A.Alam, "Recent issues in negative bias temperature instability: Initial degradation, field dependence of interface trap generation, hole trapping effects, and relaxation," *IEEE Transactions on Electron Devices*, vol. 54, pp. 2143–2154, September 2007.

[69] A. Jagannathan et al., "Microarchitecture evaluation with floorplanning and interconnect pipelining," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 8–15, January 2005.

[70] M. Janicki, G. De Mey, and A. Napieralski, "Transient thermal analysis of multilayered structures using Green's functions," *Microelectronics and Reliability*, vol. 42, pp. 1059–1064, 2002.

[71] K. O. Jeppson and C. M. Svensson, "Negative bias stress of MOS devices at high electric fields and degradation of NMOS devices," *Journal of Applied Physics*, vol. 48, pp. 2004–2014, 1977.

[72] Y. Joshi, "Emerging thermal challenges in electronics driven by performance, reliability and energy efficiency," in *8th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, 2002.

[73] B. Kaczer, V. Arkhipov, R. Degraeve, N. Collaert, G. Groeseneken, and M. Goodwin, "Disorder-controlled kinetics model for NBTI and its experimental verification," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 381–387, 2005.

[74] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai, "Design impact of positive temperature dependence on drain current in Sub-1-V CMOS VLSIs," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 10, pp. 1559–1564, October 2001.

[75] A. T. Krishnan, V. Reddy, S. Chakravarthi, J. Rodriguez, S. John, and S. Krishnan, "NBTI impact on transistor and circuit: Models, mechanisms and scaling effects," in *IEEE International Electronic Devices Meeting*, pp. 14.5.1–14.5.4, 2003.

[76] S. H. Kulkarni, D. Sylvester, and D. Blaauw, "A statistical framework for post-silicon tuning through body bias clustering," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 39–46, 2006.

[77] S. V. Kumar, C. H. Kim, , and S. S. Sapatnekar, "An analytical model for negative bias temperature instability (NBTI)," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 493–496, 2006.

[78] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Mathematically-assisted adaptive body bias (ABB) for temperature compensation in gigascale lsi systems," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 559–564, 2006.

[79] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-aware synthesis of digital circuits," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 370–375, 2007.

[80] T. Kuroda, T. Fujita, S. Mita, T. Nagamatu, S. Yoshioka, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9 V 150 MHz 10 mW 2-D discrete cosine transform core processor with variable-threshold-voltage scheme," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 166–167, 1996.

[81] B. Lee, L. Kang, W. Qi, R. Nieh, Y. Jeon, K. Onishi, and J. Lee, "Ultrathin hafnium oxide with low leakage and excellent reliability for alternative gate dielectric application," *IEEE International Electronic Devices Meeting*, pp. 133–136, 1999.

[82] D. Lee, D. Blaauw, and D. Sylvester, "Static leakage reduction through simultaneous $V_t/T_{ox}$ and state assignment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1014–1029, July 2005.

[83] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "Efficient full-chip thermal modeling and analysis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 319–326, November 2004.

[84] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC thermal simulation and modeling via efficient multigrid-based approaches," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 25, no. 9, pp. 1763–1776, September 2006.

[85] J.-M. Lin and Y.-W. Chang, "TCG: A transitive closure graph based representation for non-slicing floorplans," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 764–769, 2001.

[86] D. L. Logan, *A First Course in the Finite Element Method*. Pacific Grove, CA: Brooks/Cole Publishing Company, third ed., 2002.

[87] C. Long, L. J. Simonson, W. Liao, and L. He, "Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 640–645, 2004.

[88] R. Mahajan, C.-P. Chiu, and G. Chrysler, "Cooling a microprocessor chip," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1476–1486, August 2006.

[89] S. Mahapatra, K. Ahmed, S. Varghese, A. E. Islam, G. Gupta, L. Madhav, D. Saha, and M. A. Alam, "On the physical mechanism of NBTI in silicon oxynitride p-MOSFETs: Can differences in insulator processing conditions resolve the interface trap generation versus hole trapping controversy?," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 1–9, 2007.

[90] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 721–725, 2002.

[91] J. G. Massey, "NBTI: What we know and what we need to know — A tutorial addressing the current understanding and challenges for the future," in *Proceedings of the IEEE International Integrated Reliability Workshop Final Report*, pp. 199–211, 2004.

[92] R. McGowen, C. A. Poirier, C. Bostak, J. Ignowski, M. Millican, W. H. Parks, and S. Naffziger, "Power and temperature control on a 90-nm Itanium family processor," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 229–237, January 2006.

[93] M. Miyazaki, G. Ono, and T. Kawahara, "Optimum threshold-voltage tuning for low-power high-performance microprocessor," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 17–20, 2005.

[94] D. C. Montgomery, *Design and Analysis of Experiments*. New York, NY: John Wiley, 1991.

[95] S. Mukhopadhyay, A. Raychowdury, K. Roy, and C. Kim, "Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 363–381, March 2005.

[96] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, August 1995.

[97] S. Narendra, A. Keshavarzi, B. A. Bloechel, S. Borkar, and V. De, "Forward body bias for microprocessors in 130-nm technology generation and beyond," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 5, pp. 696–701, May 2003.

[98] V. Nookala, Y. Chen, D. J. Lilja, and S. S. Sapatnekar, "Microarchitecture-aware floorplanning using a statistical design of experiments approach," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 579–584, 2005.

[99] V. Nookala, D. J. Lilja, and S. S. Sapatnekar, "Temperature-aware floorplanning of microarchitecture blocks with IPC-power dependence modeling and transient analysis," in *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 298–303, 2006.

[100] S. Ogawa and N. Shiono, "Generalized diffusion-reaction model for the low-field charge-buildup instability at the Si-SiO2 interface," *Journal of Applied Physics*, vol. 51, pp. 4128–4230, February 1995.

[101] G. Ono, M. Miyazaki, H. Tanaka, N. Ohkubo, and T. Kawahara, "Temperature referenced supply voltage and forward-body-bias control (TSFC) architecture for minimum power consumption," in *Proceedings of the European Solid State Circuits Conference*, pp. 391–394, 2004.

[102] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 1999.

[103] M. N. Özişik, *Heat Transfer: A Basic Approach*. New York, NY: McGraw-Hill, 1985.

[104] C. R. Parthasarathy, M. Denais, V. Huard, G. Ribes, E. Vincent, and A. Bravaix, "New insights into recovery characteristics post NBTI stress," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 471–477, 2006.

[105] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," *IEEE Electron Device Letters*, vol. 26, pp. 560–562, August 2003.

[106] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Temporal performance degradation under NBTI: Estimation and design for improved reliability of nanoscale circuits," in *Proceedings of the Design, Automation and Test in Europe*, pp. 1–6, 2006.

[107] C. Piorier, R. McGowen, C. Bostak, and S. Naffziger, "Power and temperature control on an Itanium®-family processor," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 304–305, 2005.

[108] E. Pop, R. W. Dutton, and K. E. Goodson, "Analytic band Monte Carlo model for electron transport in Si including acoustic and optical phonon disperson," *Journal of Applied Physics*, vol. 96, no. 9, pp. 4998–5005, 2004.

[109] E. Pop, S. Sinha, and K. E. Goodson, "Heat generation and transport in nanometer-scale transistors," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1587–1601, August 2006.

[110] R. Prasher, "Thermal interface materials: Historical perspective, status, and future directions," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1571–1586, August 2006.

[111] H. Qian, *Stochastic and Hybrid Linear Equation Solvers and their Applications in VLSI Design Automation*. PhD thesis, Minneapolis, MN: University of Minnesota, 2006.

[112] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Random walks in a supply network," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 93–98, 2003.

[113] H. Qian and S. S. Sapatnekar, "Stochastic preconditioning for iterative linear equation solvers," *SIAM Journal on Scientific Computing*, (to appear).

[114] H. Qian and S. S. Sapatnekar, "Hierarchical random-walk algorithms for power grid analysis," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 499–504, 2004.

[115] H. Qian and S. S. Sapatnekar, "A hybrid linear equation solver and its application in quadratic placement," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 905–909, November 2005.

[116] V. Reddy, A. T. Krishnan, A. Marshall, J. Rodriguez, S. Natarajan, T. Rost, and S. Krishnan, "Impact of negative bias temperature instability on digital circuit reliability," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 248–254, April 2002.

[117] R. Reif, A. Fan, K.-N. Chen, and S. Das, "Fabrication technologies for three-dimensional integrated circuits," in *Proceedings of the IEEE International Symposium on Quality Electronic Design*, pp. 33–37, 2002.

[118] H. Reisenger, O. Blank, W. Heinrigs, W. Gustin, and C. Schlunder, "A comparison of very fast to very slow components in degradation and recovery due to NBTI and bulk hole trapping to existing physical models," *IEEE Transactions on Devices and Materials Reliability*, vol. 7, pp. 119–129, March 2007.

[119] H. Reisenger, O. Blank, W. Heinrigs, A. Muhlhoff, W. Gustin, and C. Schlunder, "Analysis of NBTI degradation and recovery behavior based on ultra fast Vt measurements," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 448–453, 2006.

[120] P. Rodgers, V. Eveloy, and M. G. Pecht, "Extending the limits of air-cooling in microelectronic equipment," in *International Conference on Thermal, Mechanical and Multiphysics Simulation and Experiments in Micro-Electronics and Micro-Systems* (*EumSimE*), pp. 695–702, 2005.

[121] J. Rowlette, E. Pop, S. Sinha, M. Panzer, and K. Goodson, "Thermal simulation techniques for nanoscale transistors," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 225–228, November 2005.

[122] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-micrometer CMOS circuits," in *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, February 2003.

[123] H. Sanchez, B. Kuttanna, T. Olson, M. Alexander, G. Gerosa, R. Philip, and J. Alvarez, "Thermal management system for high performance PowerPC$^{\text{TM}}$ microprocessors," in *Proceedings of the IEEE COMPCON*, pp. 325–330, 1997.

[124] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *The Journal of Instruction-Level Parallelism*, vol. 8, September 2005.

[125] S. S. Sapatnekar, *Timing*. Boston, MA: Springer, 2004.

[126] S. S. Sapatnekar and H. Su, "Analysis and optimization of power grids," *IEEE Design and Test*, vol. 20, no. 3, pp. 7–15, May–June 2002.

[127] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick, "Repeater scaling and its impact on CAD," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 23, no. 4, pp. 451–463, April 2004.

[128] D. K. Schroder, "Negative bias temperature instability: Physics, materials, process, and circuit issues," available at http://www.ewh.ieee.org/r5/denver/sscs/Presentations/2005.08.Schroder.p%df, 2005.

[129] A. Shakouri, "Nanoscale thermal transfer and microrefrigerators on a chip," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1613–1638, August 2006.

[130] J. Sharp, J. Bierschenk, and H. B. Lyon, Jr., "Overview of solid-state thermal microrefrigerators and possible applications to on-chip thermal management," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1602–1612, August 2006.

[131] C. Shen, M. F. Li, C. E. Foo, T. Yang, D. M. Huang, A. Yap, G. S.Samudra, and Y.-C. Yeo, "Characterization and physical origin of fast $V_{th}$ transient in NBTI of pMOSFETs with SiON dielectric," in *IEEE International Electronic Devices Meeting*, pp. 333–336, 2006.

[132] S. Sinha, E. Pop, R. W. Dutton, and K. E. Goodson, "Non-equilibrium phonon distribution in sub 100 nm silicon transistors," *Transactions of the ASME*, vol. 28, pp. 638–647, July 2006.

[133] K. Skadron, T. Abdelzaher, and M. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," in *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture*, pp. 17–28, 2002.

[134] K. Skadron, K. Sankaranarayanan, S. Velusamy, D. Tarjan, M. Stan, and W. Huang, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 94–125, March 2004.

[135] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proceedings of the ACM International Symposium on Computer Architecture*, pp. 2–13, 2003.

[136] J. H. Stathis, "Reliability limites for the gate insulator in CMOS technology," *IBM Journal of Research and Development*, vol. 46, pp. 265–286, March/May 2002.

[137] J. H. Stathis and S. Zafar, "The negative bias temperature instability in MOS devices: A review," *Journal of Microelectronics Reliability*, vol. 46, pp. 270–286, February–April 2006.

[138] L. M. Ting, J. S. May, W. R. Hunter, and J. W. McPherson, "AC electromigration characterization and modeling of multilayered interconnects," in *Proceedings of the IEEE International Reliability Physics Symposium*, pp. 311–316, 1993.

[139] C. H. Tsai and S. M. Kang, "Cell-level placement for improving substrate thermal distribution," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 19, no. 2, pp. 253–266, Feburary 2000.

[140] J.-L. Tsai, C. C.-P. Chen, G. Chen, B. Goplen, H. Qian, Y. Zhan, S.-M. Kang, M. D. F. Wong, and S. S. Sapatnekar, "Temperature-aware placement for SOCs," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1502–1518, August 2006.

[141] J. Tschanz, S. Narendra, A. Keshavarazi, and V. De, "Adaptive circuit techniques to minimize variation impacts on microprocessor performance and power," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 9–12, 2005.

[142] J. W. Tschanz, J. Kao, S. G. Narendra, R. Nair, D. Antoniadis, A. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1396–1402, November 2002.

[143] J. W. Tschanz, N. S. Kim, S. Dighe, J. Howard, G. Ruhl, S. Vangal, S. Narendra, Y. Hoskote, H. Wilson, C. Lam, M. Shuman, C. Tokunaga, D. Somasekhar, S. Tang, D. Flnan, T. Karnik, N. Borkar, N. Kurd, and V. De, "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 292–294, 2007.

[144] J. W. Tschanz, S. G. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 5, pp. 826–829, May 2003.

[145] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1838–1845, November 2003.

[146] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 1047–1052, 2006.

[147] N. Viswanathan and C. C.-N. Chu, "FastPlace: Efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 5, pp. 722–733, May 2005.

[148] B. Wang and P. Mazumder, "Fast thermal analysis for vlsi circuits via semi analytical Greens functions in multi-layer 3-D integrated circuits," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2004.

[149] B. Wang and P. Mazumder, "A logarithmic complexity algorithm for full chip thermal analysis using multi-layer Greens function," in *Proceedings of the Design, Automation and Test in Europe*, 2006.

[150] B. Wang and P. Mazumder, "Accelerated chip-level thermal analysis using multilayer green's function," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 26, no. 2, pp. 325–344, February 2007.

[151] T.-Y. Wang and C. C.-P. Chen, "3-D Thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 21, no. 12, pp. 1434–1445, December 2002.

[152] C. H. Wann, H. Chenming, K. Noda, D. Sinitsky, F. Assaderaghi, and J. Bokor, "Channel doping engineering of MOSFET with adaptable threshold voltage using body effect for low voltage and low power applications," in *Proceedings of the IEEE International Symposium of VLSI Technology*, pp. 159–163, 1995.

[153] J. Westra, C. Bartels, and P. Groeneveld, "Probabilistic congestion prediction," in *Proceedings of the International Symposium on Physical Design*, pp. 204–209, 2004.

[154] E. Wong and S. K. Lim, "3D floorplanning with thermal vias," in *Proceedings of the Design, Automation and Test in Europe*, pp. 878–883, 2006.

[155] E. Wu, E. Nowak, A. Vayshenker, J. McKenna, D. Harmon, and R. Vollertsen, "New global insight in ultra-thin oxide reliability using accurate experimental methodology and comprehensive database," *IEEE Transactions on Devices and Materials Reliability*, vol. 1, pp. 69–80, 2001.

[156] E. Y. Wu, E. J. Nowak, A. Vayshenker, W. L. Lai, and D. L. Harmon, "CMOS scaling beyond the 100-nm node with silicon-dioxide-based gate dielectrics," *IBM Journal of Research and Development*, vol. 46, pp. 287–298, March/May 2002.

[157] E. Y. Wu, J. H. Stathis, and L.-K. Han, "Ultra-thin oxide reliability for ULSI applications," *Semiconductor Science and Technology*, vol. 15, pp. 425–435, 2000.

[158] Y. W. Wu, C.-L. Yang, P.-H. Yuh, and Y.-W. Chang, "Joint exploration of architectural and physical design spaces with thermal consideration," in *Proceedings of the ACM International Symposium on Low Power Electronics and Design*, pp. 123–126, 2005.

[159] R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe, "SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling," in *Proceedings of the ACM International Symposium on Computer Architecture*, pp. 84–97, 2003.

[160] S. Yajuan, W. Zuodong, and W. Shaojun, "Energy-aware Supply and Body Biasing Voltage Scheduling Algorithm," in *Proceedings of the International*

*Conference on Solid State and Integrated Circuits Technology*, pp. 1956–1959, 2004.

[161] L. Yan, J. Luo, and N. K. Jha, "Combined dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 30–37, 2003.

[162] L. Yan, J. Luo, and N. K. Jha, "Joint dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 24, no. 7, pp. 1030–1041, July 2005.

[163] Y. Yang, C. Zhu, Z. Gu, L. Shang, and R. P. Dick, "Adaptive multi-domain thermal modeling and analysis for integrated circuit synthesis and design," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 575–582, 2006.

[164] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. New York, NY: John Wiley and Sons, 1999.

[165] S. Zafar, B. H. Lee, J. Stathis, A. Callegari, and T. Ning, "A model for negative bias temperature instability (NBTI) in oxide and high k pFETs," in *Proceedings of the IEEE Symposium on VLSI Technology*, pp. 208–209, 2004.

[166] Y. Zhan and S. S. Sapatnekar, "High efficiency Green function-based thermal simulation algorithms," *IEEE Transactions on Computer-Aided Design of Integrated*, vol. 26, no. 9, pp. 1661–1675, September 2007.

[167] S. Zhang and K. S. Chatha, "Approximation algorithm for the temperature-aware scheduling problem," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 281–288, 2007.

[168] T. Zhang, Y. Zhan, and S. S. Sapatnekar, "Temperature-aware routing in 3D ICs," in *Proceedings of the Asia-South Pacific Design Automation Conference*, pp. 309–314, 2006.

[169] P. Zhou, Y. Ma, Z. Li, R. P. Dick, L. Shang, H. Zhou, X. Hong, and Q. Zhou, "3D-STAF: Scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 590–597, 2007.

[170] B. Zhu, J. S. Suehle, Y. Chen, and J. B. Bernstein, "Negative bias temperature instability of deep sub-micron p-MOSFETs under pulsed bias stress," in *Proceedings of the IEEE International Integrated Reliability Workshop Final Report*, pp. 125–129, 2002.