

# Temperature-Aware Placement for SOCs

Charlie Chung-Ping Chen\*, Jeng-Liang Tsai<sup>¶</sup>, Guoqiang Chen<sup>†</sup>, Brent Goplen<sup>†</sup>, Haifeng Qian<sup>†</sup>, Yong Zhan<sup>†</sup>,  
Sung-Mo (Steve) Kang<sup>‡</sup>, Martin D. F. Wong<sup>§</sup>, and Sachin S. Sapatnekar<sup>†</sup>

\* National Taiwan University

<sup>¶</sup> University of Wisconsin-Madison

<sup>†</sup> University of Minnesota

<sup>‡</sup> University of California Santa Cruz

<sup>§</sup> University of Illinois at Urbana-Champaign

**Abstract**—Dramatic rises in the power consumption and integration density of contemporary systems-on-chip (SoCs) have led to the need for careful attention to chip-level thermal integrity. High temperatures or uneven temperature distributions may result not only in reliability issues, but also timing failures, due to the temperature-dependent nature of chip time-to-failure and delay, respectively. To resolve these issues, high quality, accurate thermal modeling and analysis, and thermally-oriented placement optimizations, are essential prior to tapeout. This paper first presents an overview of thermal modeling and simulation methods such as finite-difference time domain, finite element, model reduction, random walk, and Green-function based algorithms, that are appropriate for use in placement algorithms. Next, 2D and 3D thermal-aware placement algorithms such as matrix-synthesis, simulated annealing, partition-driven, and force directed are presented. Finally, future trends and challenges are described.

## I. INTRODUCTION AND PRELIMINARIES

Technology scaling has brought about the ability to operate circuits with millions of transistors at gigahertz frequencies. A side-effect of this is the large amount of power that is dissipated on-chip, which manifests itself physically as heat, leading to elevated temperatures. These thermal effects can have unfortunate results on the operation of the chip, and can result in both catastrophic and parametric failures. Large temperature gradients can reduce the lifetime of a chip, and in extreme cases, can cause material stresses and crack the chip. In addition, nonuniform temperatures can induce timing variations, since both transistor performance and metal conductivity are temperature-dependent, which may result in logic errors [1].

An example temperature profile for an industrial chip is shown in Fig. 1: an examination of the thermal contours shows that the temperature at the hot spots can exceed 100°C. With the growing realization of the impact of thermal effects on circuit performance and reliability, thermal issues have started to affect on chip architectures, and methods for alleviating thermal effects have become extremely important. For example, mobile devices now typically have multiple power modes to improve their energy efficiency without sacrificing

This work was supported in part by the National Science Foundation under Awards CCR-0205227, CCF-0541367, CCR-0093309, and CCR-0204468, National Science Council of Taiwan, R.O.C. under grant NSC 92-2218-E-002-030, and by the Semiconductor Research Corporation under Grant 2003-TJ-1092.

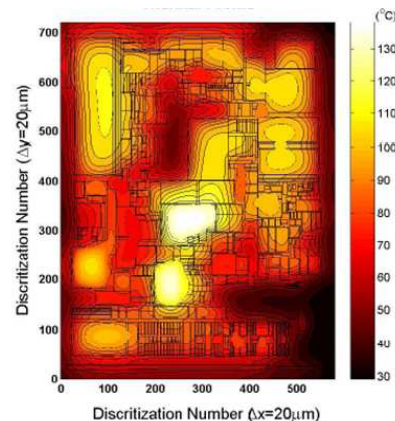


Fig. 1. The temperature profile of an industrial chip.

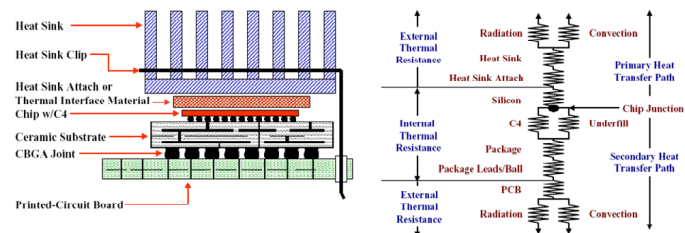


Fig. 2. Heat dissipation paths of a chip in a system.

performance [2]. On-chip power mode controllers can also force a chip to enter low power mode if on-chip temperature sensors detect an excessively high temperature. While these are important strategies for thermal mitigation, more effort must be invested at the design stage to alleviate thermal problems.

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 thermal system. Fig. 2 shows a chip in a CBGA packaging and its surrounding environment, including the heat sink and the underlying printed circuit board. A simplified thermal model of the packaging and surrounding environment is also shown in the figure. The heat generated on chip can be dissipated through the packaging to the heat sink, and then to the ambient. A small portion of the heat can also be dissipated through the packaging to the printed circuit board.

Transistor performance and metal conductivity are both tem-

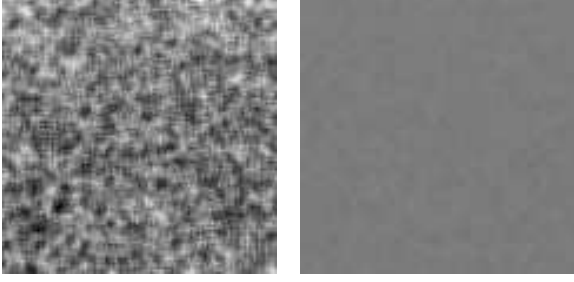


Fig. 3. Temperature profiles with (left) random placement and (right) thermal placement.

perature dependent. Therefore, on-chip thermal modeling must achieve micrometer resolution. For on-chip thermal modeling, several approaches can be used to model the heat transfer in the substrate. Finite-difference time domain [3], finite element [4], model reduction [5], random walk [6], [7], and Green-function [8] based algorithms are appropriate for on-chip thermal modeling and analysis.

An excellent way to address temperature issues during design is to ensure that circuit blocks are placed in such a way that they even out the thermal profile: in other words, using temperature-aware placement [4], [9]–[11]. Simplistically, if we spread high-power cells across the chip “evenly,” the temperature profile will be flat and we can avoid hot-spot related thermal issues. In reality, thermal placement is more complex, and a uniform distribution of power sources does not lead to a uniform temperature; it is also essential to consider heat sink characteristics and edge effects. Fig. 3 shows the temperature profiles of a test circuit using random placement and thermally-driven placement, and illustrates graphically that hot spots can be significantly reduced by thermal placement.

The remainder of this paper is organized as follows. In Section II, we briefly review thermal modeling techniques and methods for calculating steady state temperature profile for placement. In Section III, we review three major 2D and 3D thermal placement algorithms. Finally, we point out future research directions in Section IV and conclude the paper.

## II. OVERVIEW OF THERMAL MODELING FOR PLACEMENT

### A. Fundamentals of Heat Transfer and Thermal Modeling

Before discussing thermal placement techniques, it is important to understand the mechanisms of heat transfer. There are three means for heat transfer: *conduction*, *convection* and *radiation*. Conduction is heat transfer through molecular interactions within a material without any movement of the material. The rate of conduction heat transfer through a barrier is as follows.

$$\frac{Q}{t} = \frac{\kappa A (T_{hot} - T_{cold})}{d}. \quad (1)$$

Here,  $Q$  is the heat transferred in time  $t$ ,  $\kappa$  is the thermal conductivity of the barrier,  $A$  is the area,  $T$  is the temperature and  $d$  is the thickness of the barrier. The mechanism for convection is heat transfer by mass motion of a fluid or a gas. The volume of a fluid or a gas usually expands or contracts with a change on the temperature, thereby causing convection

currents and speeding up the heat transfer. Convection can also be generated intentionally. For example, microprocessors usually need cooling fans to generate air flow through the heat sinks to speedup heat dissipation. The phenomenon of radiation corresponds to heat transfer by electromagnetic wave emission which carries energy away from the emitting object.

Conduction is the major heat transfer mechanism within VLSI chips. As dielectric materials are poor heat conductors, conduction largely occurs within the substrate. The heat conduction in the chip substrate is governed by the following partial differential equation of heat conduction from the law of energy conservation [12]

$$\rho c_p \frac{\partial T(\vec{r}, t)}{\partial t} = \nabla \cdot [\kappa(\vec{r}, t) \nabla T(\vec{r}, t)] + g(\vec{r}, t) \quad (2)$$

subject to the following thermal boundary condition

$$\kappa(\vec{r}, T) \frac{\partial T(\vec{r}, t)}{\partial n_i} + h_i T(\vec{r}, t) = f_i(r_{s_i}, t) \quad (3)$$

where  $T$  is the time-dependent temperature at any  $\vec{r}$ ,  $\rho$  is the density of the material,  $c_p$  is the specific heat,  $\kappa$  is the thermal conductivity,  $g$  is the heat energy generation rate,  $n_i$  is the heat-transfer coefficient on the boundary surface of the chip,  $f_i(r_{s_i}, t)$  is an arbitrary function on the boundary surface  $s_i$ , and  $\partial/\partial n_i$  is the differentiation along the outward direction normal to the boundary surface  $s_i$ . Fig. 4 illustrates conservation of energy and the heat conduction equation.

In general, thermal conductivity  $\kappa(\vec{r}, T)$  is position- and temperature-dependent. However, thermal conductivity variations are usually not significant, and for practical purposes, the substrate can be treated as a homogeneous material with a constant  $\kappa$ . The heat-generation rate  $g(\vec{r}, t)$  arises from the power consumption in cells and interconnects. The power consumption of cells is caused by dynamic, short-circuit and leakage currents, and the power consumption of interconnects is generated by charging and discharging interconnect capacitances. Since the charging and discharging currents of an interconnect need to go through its driving cell, whose driving resistance is usually much larger than the interconnect resistance, the power consumption tend to concentrate on the cell. Therefore, cells are treated as the only heat sources in a chip. Although a small portion of the interconnect power consumption does dissipate on the metal wire and increase its temperature, this self-heating effect is usually more related to the electromigration and lifetime of the wire than the temperature profile of a chip. Since electromigration is dependent on both current density and temperature, electromigration analysis must be performed after placement, and this topic is therefore beyond the scope of this paper.

The time constant of on-chip heat conduction is much larger than the clock periods used in current technologies. This means that transient currents with short time constants do not have significant effects on the temperature profile once it reaches a steady state, and average power consumption can be used to obtain the steady state temperature profile. In effect, the thermal network acts as a low pass filter that filters out the effect of fast transients. However, significant changes, such as changes in the power mode of a processor

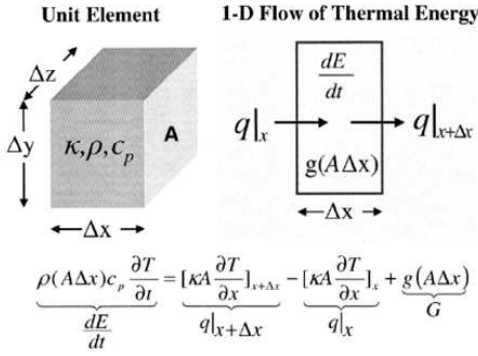


Fig. 4. Conservation of energy and the heat conduction equation.

(note that the frequency of occurrence of these is larger than the time constant of the thermal network), require a transient analysis, or as an approximation, steady state analysis in multiple power modes. In the following subsections, we review various methods for temperature profile analysis.

### B. Finite Difference Time Domain Approach

The two-dimensional heat conduction equation can be rewritten as

$$\frac{\partial T(x, y, t)}{\partial t} = \frac{\kappa}{\rho c_p} \frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\kappa}{\rho c_p} \frac{\partial^2 T(x, y, t)}{\partial y^2} + \frac{1}{\rho c_p} g(x, y, t). \quad (4)$$

This is a second order parabolic partial differential equation and can be rewritten as a difference equation by space and time discretization. After discretization, the temperature  $T(x, y, t)$  at each discrete point is replaced by  $T(i\Delta x, j\Delta y, n\Delta t)$ . The first-order partial derivative of  $T$  with respect to  $x$  can be approximated by the forward-difference approximation

$$\left. \frac{\partial T}{\partial x} \right|_{i,j}^n = \frac{T_{i+1,j}^n - T_{i,j}^n}{\Delta x} + O(\Delta x) \approx \frac{T_{i+1,j}^n - T_{i,j}^n}{\Delta x} \quad (5)$$

where the truncation error is  $O(\Delta x)$ . Similarly, the second-order partial derivative of  $T$  with respect to  $x$  can be approximated by the central-difference approximation

$$\begin{aligned} \left. \frac{\partial^2 T}{\partial x^2} \right|_{i,j}^n &= \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} + O((\Delta x)^2) \\ &\approx \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} = \frac{\delta_x^2 T^n}{(\Delta x)^2} \end{aligned} \quad (6)$$

where  $\delta_x^2 T^n = T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n$  and the truncation error is  $O((\Delta x)^2)$ . By applying the explicit update on the right hand side of (4), we have

$$\frac{T^{n+1} - T^n}{\Delta t} = \frac{\kappa}{\rho c_p} \left[ \frac{\delta_x^2 T^n}{(\Delta x)^2} + \frac{\delta_y^2 T^n}{(\Delta y)^2} \right] + \frac{g}{\rho c_p}. \quad (7)$$

Note that in (7),  $T^{n+1}$  only depends on  $T^n$  and can be solved directly without matrix inversion. This explicit method has second-order accuracy in space and first-order accuracy in time. Alternatively, implicit method, Crank-Nicholson method and alternating direction implicit (ADI) method can be used [3].

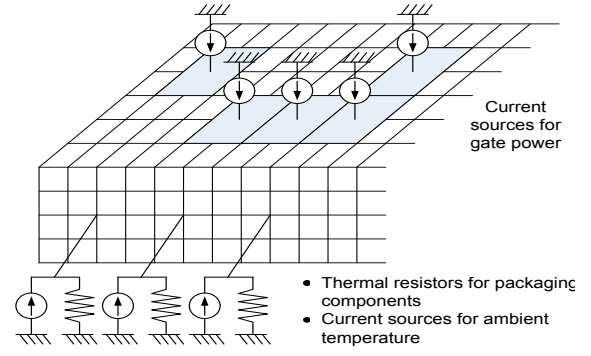


Fig. 5. The substrate, cells and boundary conditions and their equivalent models.

### C. RC Equivalent Network Approach

At steady state, (2) can be simplified as

$$\kappa \nabla^2 T + g(\vec{r}) = 0 \quad (8)$$

which is in the same form as the Poisson equation describing the relation between charge density and electrical potential:

$$\epsilon_0 \epsilon_s \nabla^2 V + \rho(\vec{r}) = 0. \quad (9)$$

Therefore, the substrate can be modeled as an RC equivalent circuit, a heat source can be modeled as a current source, boundary condition with constant temperature can be modeled as a voltage source, and nodal temperature can be modeled as the nodal voltage in the RC equivalent circuit. Fig. 5 shows the substrate, cells and boundary conditions and their equivalent models. The advantage of using RC equivalent models is that well-developed circuit analysis techniques are readily available for thermal analysis. However, if we discretize the substrate uniformly, we need to use a fine grid to ensure enough resolution on any part of the chip. This can result in a large equivalent circuit and long simulation time.

### D. Model Order Reduction Approach

Wang *et al.* [5] use an adaptive approach to keep the problem size small. The basic idea is to use a coarse grid to form an RC equivalent circuit and find the rough temperature profile. The regions with high temperature gradients are discretized further and the system is simulated again. After modeling the substrate and interconnects, the equivalent thermal circuit can be expressed by the time-domain Modified Nodal Analysis (MNA) equation

$$Gx + C\dot{x} = Bu, \quad (10)$$

where  $G$  and  $C$  represent the conductance and capacitance matrices,  $x$  is the vector of node voltages,  $u$  is the vector of independent current sources,  $B$  is the input adjacency matrix mapping the sources to the internal states.

To further speedup the simulation, a model order reduction technique is used to further reduce the model size. Model order reduction generates an analytic model that is a compact representation of original circuits by matching their moments or poles. These methods typically operate in the frequency

domain, and to outline the procedure, we begin with the Laplace transformation of (10):

$$GX + sCX = BU. \quad (11)$$

Applying the Taylor series expansion at zero frequency on both sides of (11), we have

$$(G + sC)(m_0 + m_1s + m_2s^2 + \dots) = B(u_0 + u_1s + u_2s^2 + \dots) \quad (12)$$

where  $m_i$  and  $u_i$ , the coefficients of the  $i^{\text{th}}$  term in the Taylor series, are known as the  $i^{\text{th}}$  moment of  $x$  and  $u$ , respectively. Moment matching is a method that represents the finite unknown moments of the left hand side of (12) in terms of the known moments of the right hand side. In the basic moment-matching method based on AWE [13], as well as in improved methods like PRIMA [14], the sources are set to impulses in order to compute the transfer function. The impulse sources are constant in the frequency domain and contribute only to the initial vector. Equation (12) can be rewritten as

$$(G + sC)(m_0 + m_1s + m_2s^2 + \dots) = Bu_0. \quad (13)$$

This results in an iterative relationship between the moments:

$$Gm_0 = Bu_0 \quad (14)$$

$$Gm_i + Cm_{i-1} = 0 \quad (15)$$

However, there are numerical stability problems in this basic-moment matching method, especially for higher order moment computations. To avoid the numerical errors, an orthogonal basis  $V$  is determined. This is constructed from the Krylov subspace,  $Kr(A, R, q) = \text{colsp}(R, AR, A^2R, \dots, A^{q-1}R)$  (where  $A = -G^{-1}C$  and  $R = G^{-1}B$ ), over the subspace spanned by finite moments of  $x(s)$ . The order-reduced model can be obtained by projecting the original system onto the Krylov subspace using a congruent transform, which results in the system of equations

$$\tilde{G}\tilde{x} + s\tilde{C}\tilde{x} = \tilde{B}u \quad (16)$$

where  $\tilde{G} = V^T G V$ ,  $\tilde{C} = V^T C V$ , and  $\tilde{B} = V^T B$ . However, the bottleneck in this method is the number of sources in the input vector  $u$ , since the large size of  $B$  results in a high dimension Krylov subspace. This problem is especially acute while solving the equivalent thermal circuit, which has a large number of independent current sources.

Wang *et al.* use the Improved Extended Krylov Subspace (IEKS) method [15], inspired by the EKS method [16], to resolve the issue caused by large number of independent current sources. Unlike PRIMA, whose runtime depends strongly on the number of ports, the runtime of EKS is independent of this parameter. EKS models the piece-wise-linear (PWL) independent sources as a sum of delayed ramps in the frequency domain

$$u(s) = \frac{1}{s^2} \sum_{i=1}^N r_i \exp(-\beta_i s). \quad (17)$$

This expression contains  $\frac{1}{s}$  and  $\frac{1}{s^2}$  terms. While traditional Krylov subspace methods typically begin moment matching

from the 0th moment, EKS extends the Krylov subspace by shifting the moments in the frequency spectrum. However, moment shifting in EKS is tedious and error-prone, and an improved moment calculation method, IEKS, which ensures the  $-1^{\text{st}}$  and  $-2^{\text{nd}}$  order moments are zero for arbitrary finite time PWL waveforms, without moment shifting, is adopted for thermal analysis.

For a given finite-time PWL source, the moment representation of IEKS with the  $-1^{\text{st}}$  and  $-2^{\text{nd}}$  order moments are zero. Let the given finite-time PWL source  $u(t)$  be represented as

$$u(t) = \sum_{i=0}^N \{ [a_i + \gamma_i(t - t_i)] E_{t-t_i} - [a_{i+1} + \gamma_i(t - t_{i+1})] E_{t-t_{i+1}} \}, \quad (18)$$

where  $\gamma_i = (a_{i+1} - a_i)/(t_{i+1} - t_i)$  and  $E_{t-t_i}$  is the unit-step function with a delay of  $t_i$ . By taking the Laplace transform and Taylor expansion of (18), we have

$$u(s) = \frac{1}{s^2} \sum_{i=0}^N \left[ a_i s \sum_{l=0}^{\infty} (-1)^l \frac{t_i^l}{l!} s^l + \gamma_i \sum_{l=0}^{\infty} (-1)^l \frac{t_i^l}{l!} s^l - a_{i+1} s \sum_{l=0}^{\infty} (-1)^l \frac{t_{i+1}^l}{l!} s^l - \gamma_i \sum_{l=0}^{\infty} (-1)^l \frac{t_{i+1}^l}{l!} s^l \right]. \quad (19)$$

If  $\tilde{u}_i$  denotes the coefficient of the  $s^i$  term, then (19) can be simplified as

$$\mathcal{L}(u(t)) = \{ \tilde{u}_{-2} s^{-2} + \tilde{u}_{-1} s^{-1} + \tilde{u}_0 + \tilde{u}_1 s + \tilde{u}_2 s^2 + \dots + \tilde{u}_m s^m + \dots \}. \quad (20)$$

After the detailed calculation, the first two coefficients,  $\tilde{u}_{-2}$  and  $\tilde{u}_{-1}$  are zero. The process of generating an orthogonal basis  $V$  for the corresponding moments is similar to [16].

### E. Finite Element Analysis

In finite element analysis (FEA), the design space is first discretized or meshed into elements. Different element shapes can be used such as tetrahedra and hexahedra. A four-node tetrahedral element is the simplest possible three dimensional element, but it does not simulate heat conduction in rectangular structures well. An rectangular prism can simulate heat conduction in lateral directions without aberrations in the prime directions.

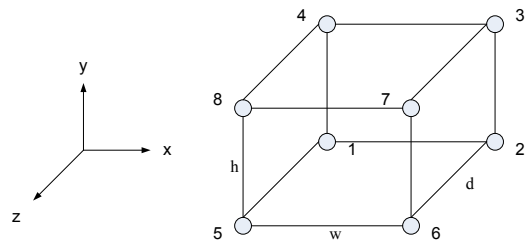


Fig. 6. An eight-node rectangular prism element for FEA.

In FEA, the temperatures are calculated at discrete points, the nodes of the elements, and the temperatures elsewhere within the elements are interpolated using a weighted average of the temperatures at the nodes. In deriving the finite element

equations, the differential equation describing heat conduction is approximated within the elements using this interpolation. For an 8-node hexahedral element shown in Fig. 6, a trilinear interpolation function is used to describe the temperature within each element based on the nodal temperatures:

$$T(x, y, z) = \mathbf{N}^T \mathbf{T} \quad (21)$$

where  $\mathbf{N} = [N_1 N_2 \cdots N_8]^T$  and  $\mathbf{T} = [T_1 T_2 \cdots T_8]^T$ , where  $T_i$  is the temperature at the  $i^{\text{th}}$  of eight vertices of the rectangular prism, and  $N_i$  is the shape function for the  $i^{\text{th}}$  vertex. The shape functions are determined by the coordinates of element's center,  $(x_c, y_c, z_c)$ , the coordinates at the nodes,  $(x_i, y_i, z_i)$ , the width,  $w$ , height,  $h$ , and depth,  $d$ , of the element.

As in circuit simulation using the modified nodal formulation (MNA) method [17], stamps are created for each element and are added to the global system of equations. In FEA, these stamps are called element stiffness matrices,  $k$ , and can be derived as follows using the variational method for an arbitrary element type [18]. The heat conduction stamp for the eight-vertex rectangular prism is derived as an  $8 \times 8$  matrix, and the global stiffness matrix,  $K$ , is derived using these as stamps, obtaining a set of equations

$$K\mathbf{T} = \mathbf{P} \quad (22)$$

where  $\mathbf{T}$  is the vector of nodal temperatures and  $\mathbf{P}$  the vector of node powers. Similarly, stamps for convective boundary conditions can be derived. Conductive boundary conditions simply correspond to fixed temperatures; since these parameters are no longer variables, they can be eliminated and moved to the right hand side.

#### F. Random Walk Methods

Random walk methods have been used very successfully for the analysis of large RC networks, in the context of power grids [6], [7]. Such methods can easily be applied to the large resistive networks that appear in steady-state thermal analysis, and the RC networks in transient thermal analysis. Since they perform very well when one node, or a small number of nodes, must be solved for, a major benefit of these methods is their ability to perform incremental analysis rapidly and efficiently. Therefore, they are excellent candidates for use in incremental placement, capturing the effects of a small 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.

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 Fig. 7. The application of Kirchoff's Current Law, Kirchoff's Voltage Law and the device equations for the conductances, yields the following equation:

$$\sum_{i=1}^{\text{degree}(x)} g_i (V_i - V_x) = I_x \quad (23)$$

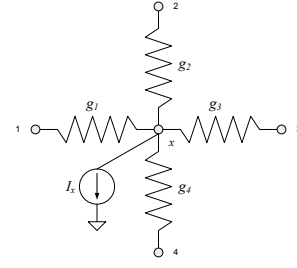


Fig. 7. A representative node in the resistive network.

where the nodes adjacent to  $x$  are labeled  $1, 2, \dots, \text{degree}(x)$ ,  $V_x$  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 (23) can be reformulated as follows:

$$V_x = \sum_{i=1}^{\text{degree}(x)} \frac{g_i}{\sum_{j=1}^{\text{degree}(x)} g_j} V_i - \frac{I_x}{\sum_{j=1}^{\text{degree}(x)} g_j}. \quad (24)$$

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 non-fixed 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.

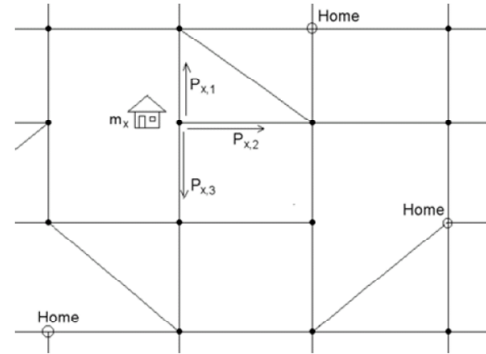


Fig. 8. An instance of a random walk "game."

Now let us look at a random walk "game," given a finite undirected connected graph (for example, Fig. 8) 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  $i = 1, 2, \dots, \text{degree}(x)$ , where  $x$  is the current node, and  $\text{degree}(x)$  is the number of edges connected to node  $x$ . These probabilities satisfy the following relationship:

$$\sum_{i=1}^{\text{degree}(x)} p_{x,i} = 1 \quad (25)$$

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 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

$$f(x) = E[\text{total money earned} \mid \text{walk starts at node } x] \quad (26)$$

It is obvious that

$$f(\text{one of the homes}) = m_{0h} \quad (27)$$

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

$$f(x) = \sum_{i=1}^{\text{degree}(x)} p_{x,i} f(i) - m_x \quad (28)$$

For a random-walk problem with  $N$  non-home 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 (28) becomes identical to (24), and equation (27) reduces to the condition of constant voltage sources.

$$p_{x,i} = \frac{g_i}{\sum_{j=1}^{\text{degree}(x)} g_j} \quad i = 1, 2, \dots, \text{degree}(x)$$

$$m_x = \frac{I_x}{\sum_{j=1}^{\text{degree}(x)} g_j} \quad m_{0h} = V_h, \quad f(x) = V_x \quad (29)$$

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 [19]. 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 [20].

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 [6], [7].

## G. Green Function Based Methods

An alternative to the FEM and FDM methods, which mesh up the entire substrate, is a boundary element method using Green functions. This method is particularly appropriate for coarse level modeling where the total number of heat sources is small, for example, at floorplanning level. The partial differential equation to be solved for thermal analysis is linear when the material properties are region-wise uniform, and therefore, conceptually, the problem can be solved by superposition, considering one source at a time. A Green function enables such a computation: it is the response in a field region to a power source in a source region; in the presence of multiple power sources, superposition can be used to sum up the responses at a field point due to each of the sources.

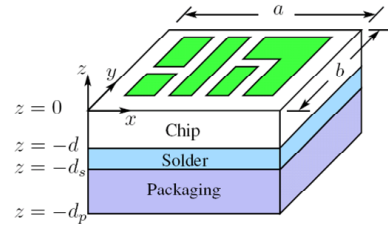


Fig. 9. Schematic of a VLSI chip with packaging.

Fig. 9 shows a schematic of a VLSI chip with the associated packaging. The shaded areas on the top surface of the chip represent the functional blocks. Let  $G(\mathbf{r}, \mathbf{r}')$ , with  $\mathbf{r} = (x, y, z)$  and  $\mathbf{r}' = (x', y', z')$ , be the distribution of temperature above  $T_a$  in the multilayered chip structure when a unit point power source of 1W is placed at position  $\mathbf{r}'$ . Then  $G(\mathbf{r}, \mathbf{r}')$  satisfies the equation

$$\nabla^2 G(\mathbf{r}, \mathbf{r}') = -\frac{\delta(\mathbf{r} - \mathbf{r}')}{k_{l(\mathbf{r})}} \quad (30)$$

and the boundary conditions

$$\left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial x} \right|_{x=0,a} = \left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial y} \right|_{y=0,b} = 0 \quad (31)$$

$$\left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial z} \right|_{z=0} = 0 \quad (32)$$

$$k_N \left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial z} \right|_{z=-d_N} = hG(\mathbf{r}, \mathbf{r}')|_{z=-d_N} \quad (33)$$

$$G(\mathbf{r}, \mathbf{r}')|_{z=-d_i+\epsilon} = G(\mathbf{r}, \mathbf{r}')|_{z=-d_i-\epsilon} \quad (34)$$

$$k_i \left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial z} \right|_{z=-d_i+\epsilon} = k_{i+1} \left. \frac{\partial G(\mathbf{r}, \mathbf{r}')}{\partial z} \right|_{z=-d_i-\epsilon} \quad (35)$$

where  $\delta(\mathbf{r}, \mathbf{r}') = \delta(x - x')\delta(y - y')\delta(z - z')$  is the three-dimensional Dirac delta function, and  $G(\mathbf{r}, \mathbf{r}')$  is the Green function. The temperature field under an arbitrary power density distribution can be obtained easily as

$$T(\mathbf{r}) = T_a + \int_0^a dx' \int_0^b dy' \int_{-d_N}^0 dz' G(\mathbf{r}, \mathbf{r}') g(\mathbf{r}') \quad (36)$$

For thermal problems encountered in chip design, both the source regions, where powers are generated, and the field regions, whose temperatures are to be computed, are located

on discrete planes. Thus, in the following analysis, we will focus on a single source plane and a single field plane, i.e., a particular  $z$  and  $z'$ . For these planes, it can be shown [8] that the Green function is given by

$$G'(x, y, x', y') \triangleq G(\mathbf{r}, \mathbf{r}')|_{z, z'} = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} C_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \cos\left(\frac{m\pi x'}{a}\right) \cos\left(\frac{n\pi y'}{b}\right) \quad (37)$$

where the coefficient  $C_{mn}$  only depends on  $z$  and  $z'$ .

The above expression is complicated, both visually and computationally, and it involves a double summation to infinity. Fortunately, several methods are available for managing the computation. The work in [8], based on the substrate analysis methods in [21], uses the discrete cosine transform (DCT) and table lookups to accelerate the Green function based thermal analysis; for the analysis of a single layer, the computational complexity is  $O(N_g^2)$ , where  $N_g$  is the number of regions. An improved method in [22] reduces the complexity from quadratic to  $O(N_g \log N_g)$ . The essential idea is to recognize that the bottleneck corresponds to a convolution operation, and this can be performed efficiently in the frequency domain: the primary cost here is in the transform from the space domain to the frequency domain.

#### H. Summary

Each thermal modeling method has its specific advantages. Generally speaking, the FDTD method is ideal for time domain dynamic simulation and is capable of capture time-of-flight effect, and model order reduction works well for simulation over long time periods for a stiff system, since the cost of each time step is greatly reduced. The random walk method can handle both transient and steady-state analysis, but its optimal runtime tradeoffs may be achieved with some accuracy limitations, so that it is ideal for coarse analysis or for incremental analysis.

Both finite difference and finite element methods discretize the space and can easily account for nonuniformities in thermal conductivities. As compared to finite differences, the finite element method can effectively reduce the number of elements for the same accuracy. In contrast, the basic Green function method attempts to find a closed form solution and is restricted to much simpler assumptions on the uniformities of the thermal conductivities throughout the space, though a small number of discrete discontinuities are easy to handle within its framework. Generally speaking, this method is preferred in cases where an approximate solution is adequate (for example, in coarse placement, or in floorplanning), while the other two methods work well when greater accuracy is desired.

### III. OVERVIEW OF THERMAL PLACEMENT ALGORITHMS

At the placement stage, as far as the thermal constraints are concerned, the general goals are to achieve globally uniform thermal distributions. One problem formulation is to minimize the maximal on-chip temperature gradient and obtain an even temperature distribution, but different methods may use different specific objectives that are correlated with this

goal. The degree of freedom that is available this purpose during placement is the control over the 2-dimension linear ordering of the heat sources, namely, the standard cells. In order to minimize the maximal on-chip temperature gradient, the following problem may be formulated:

**Find a permutation  $\pi$  of  $P_i : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  such that  $\max(|T_i - T_{i,neighbor}|)$  is minimum.**

At the placement stage, steady-state based analyses can be used to determine the locations of cells within the layout, by solving a set of equations of the type

$$\begin{bmatrix} G_{11} & G_{12} & \dots & G_{1m} \\ G_{21} & G_{22} & \dots & G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1} & G_{m2} & \dots & G_{mm} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \quad (38)$$

i.e.,  $\mathbf{GT} = \mathbf{P}$

where  $G$  is the thermal conductance matrix,  $\mathbf{T}$  is the vector of temperatures, and  $\mathbf{P}$  is the vector of power dissipations. For the finite difference method,  $G$  refers to the matrix associated with the thermal conductances, while for the finite element method, this corresponds to the stiffness matrix. The value of each  $P_i$  is not constant, but may change depending on which cell is located in a particular grid of the layout. Additionally, the power consumption of a cell varies with the interconnect capacitance that it drives, i.e., the length of the nets that it drives. During placement, these values are liable to change. The total power is actually dissipated by both the switching transistor and the interconnecting wires. Except for long global wires, the driver resistance is typically much larger than the metal resistance, and therefore most of the power is dissipated in the cells, and it is reasonable to ignore the part consumed by the metal wires. Even though self-heating of wires plays a very important role in the electromigration lifetime of the metal wires [23], during the placement stage, it may be ignored. It is potentially possible to take this into account during a later stage of placement using the congestion information.

In addition to thermal considerations, other critical objectives in placement are the conventional goals of minimizing the total wire length and meeting the timing constraints. Traditional non-thermal placement methods that consider these criteria can be divided into several classes:

- **Randomized methods:** Simulated annealing [24] is the most well-known example of this class of methods. Even though it is true that this method can reach arbitrary close to the global minimum if the cooling schedule is slow enough, it may require long run times for large circuits.
- **Analytical approaches:** This class includes methods such as force-driven placement and quadratic programming [25]–[28]. This method is used in many commercial placement tools. Combined with an iterative linear solver, this approach is fast and generates good results [29].
- **Partition-based methods:** The Kernighan-Lin method and its Fiduccia-Mattheyses implementation are well known min-cut methods that are utilized in partition-driven placement. Traditionally, partition methods are

0	3	7
<b>8</b>	<b>7</b>	0
<b>4</b>	<b>8</b>	2
3	1	3

8	1	8
3	0	3
3	<b>0</b>	<b>4</b>
7	<b>2</b>	<b>7</b>

Fig. 10. Illustration of power consumption distribution and hot spot.

known to have a few problems: for example, that the wire-length is optimized indirectly through the optimization of the min-cut, and that the locally greedy approach may sometimes lose the global view. However, recent progress in multilevel hyper-graph partition and partition-based placement has produced very impressive results [30]–[33].

### A. Matrix Synthesis

In [9], the thermal placement problem is modeled as a matrix synthesis problem. This approach assumes a specific thermal conductivity matrix and the temperature at a certain point on chip is determined by power dissipations of neighboring nodes within a certain distance. The example in Figure 10 shows a portion of a chip, and two examples of the distribution of the power dissipation in the subregions of the chip. Assuming that the maximum temperature of a node is determined by the total power consumption within a  $2 \times 2$  region, the darkened numbers in the figure show the  $2 \times 2$  region that corresponds to the hot spot in each case. The placement problem here corresponds to arranging the blocks to minimize that the maximum total power within any  $2 \times 2$  window.

Generalizing this to a  $t \times t$  window, the thermal placement problem is simplified to the following matrix synthesis problem (MSP):

**Given integers  $t, m, n$  and a list of  $mn$  nonnegative real numbers,  $x_0, x_1, \dots, x_{mn-1}$ , synthesize an  $m \times n$  matrix  $M$  out of  $x_0, \dots, x_{mn-1}$  such that the sum of the power numbers in any  $t \times t$  sub-matrix is minimized.**

Chu *et al.* [9] show that MSP is equivalent to the 3-PARTITION problem which is NP-complete. A simple approximation algorithm for MSP works as follows. Assuming power numbers  $x_0, \dots, x_{mn-1}$  are sorted in descending order, a matrix is synthesized by filling in power numbers in order. As shown in Fig. 11, the cells with largest power numbers  $x_0, \dots, x_8$  are spread across the chip evenly, thereby avoiding these cells from clustering together and creating hot spots. It is shown that if  $x_8 = \alpha x_0$ , then the sum of any  $2 \times 2$  sub-matrix in Fig. 11 is within  $\max(1.5, 2 - \alpha)$  times the optimal solution. The approximation ratio is valid for different values of  $m, n$  and  $t$ . Although the worst-case approximation ratio of this algorithm is 2, the approximation ratio is usually close to 1 in practice.

When the above algorithm is used with a large  $t$ , the final placement does not have much guarantee on the approximation factor for  $t' < t$ . A recursive approximation algorithm provides a solution to this problem.

<b>X<sub>0</sub></b>	$x_{17}$	<b>X<sub>1</sub></b>	$x_{16}$	<b>X<sub>2</sub></b>	$x_{15}$
$x_{26}$	$x_{35}$	$x_{25}$	$x_{34}$	$x_{24}$	$x_{33}$
<b>X<sub>3</sub></b>	$x_{14}$	<b>X<sub>4</sub></b>	$x_{13}$	$x_5$	$x_{12}$
$x_{23}$	$x_{32}$	$x_{22}$	$x_{31}$	$x_{21}$	$x_{30}$
<b>X<sub>6</sub></b>	$x_{11}$	<b>X<sub>7</sub></b>	$x_{10}$	<b>X<sub>8</sub></b>	$x_9$
$x_{20}$	$x_{29}$	$x_{19}$	$x_{28}$	$x_{18}$	$x_{27}$

Fig. 11. An approximation algorithm for MSP.

$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$
$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$
$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$
$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$
$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$
$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$
$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$	$L_0$	$L_1$
$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$	$L_2$	$L_3$

(a)

$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$
$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$
$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$
$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$
$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$
$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$	$l_0$	$l_1$
$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$	$l_2$	$l_3$

(b)

Fig. 12. Illustration of the recursive approximation algorithm. (a) The labels for the first level of recursion. (b) The labels for the second level recursion.

### ALGORITHM A3

1. Divide the input numbers into 4 groups  $G_0, G_1, G_2$  and  $G_3$  and label the matrix by  $L_0, L_1, L_2$  and  $L_3$  with  $t = 2$ .
2. Recursively place the numbers in  $G_0$  into the sub-matrix formed by entries marked with  $L_0$  until the size of each group is  $\frac{n^2}{t^2}$ . In that case, we do the placement arbitrarily instead of doing it recursively.
3. Apply the same procedure to  $G_1, G_2$  and  $G_3$ .

The illustration of the recursive approximation algorithm is shown in Fig. 12.

### B. Simulated Annealing

In [10], a compact substrate thermal model was developed, and two algorithms for standard cell and macro cell style design were presented. For standard cell style designs, the targeted power distribution is first computed from the desired temperature profile and this is then used to impose constraints



during placement. For macro cell style designs, special consideration is added to avoid hot spots created by thermal coupling between nearby hot macro cells.

By the superposition principle, the boundary conditions and thermal distribution constraints can be converted into power distribution constraints. The temperature profile  $T_{total}$  consists of the following components:

$$\begin{aligned} T_{total} &= \sum T_{movable\_cells} + \sum T_{fixed\_cells} + T_{ambient} \\ &= \sum T_{movable\_cells} + T_{BC} \end{aligned} \quad (39)$$

$T_{BC}$  can be obtained by an analysis with power consumptions from fixed cells and boundary conditions, and

$$\sum T_{movable\_cells} = RP$$

where  $R$  is the thermal resistance matrix and  $\mathbf{P}$  is the vector of power dissipations of movable cells. Let the optimal temperature profile be a constant temperature  $T_s$  across chip, thermal and power constraints can be combined as:

$$\begin{bmatrix} \mathbf{R}^{m \times m} & -\mathbf{1}^{m \times 1} \\ \mathbf{1}^{1 \times m} & 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{P}^{m \times 1} \\ T_s \end{bmatrix} = \begin{bmatrix} -\mathbf{T}_{BC} \\ P_{total} \end{bmatrix} \quad (40)$$

where  $P_{total}$  is the total power consumption of movable cells. Given a total power budget, the power distribution constraints (power budget at each node) and the optimal temperature  $T_s$  can be obtained by solving (40).

With the power distribution constraints, a simulated annealing based thermal placement tool for standard cell designs is developed in [10]. During the annealing process, power distribution constraints are treated as hard constraints, i.e., the moves that violate nodal power constraints are rejected. Although this slows down the annealing process and increase the runtime, experimental results show that the runtime is only  $1.5\times$  than without power distribution constraints. Moreover, the slower annealing process also increases the solution quality (total wirelength) in many tested circuits.

For macro cells, the power dissipation remains relatively constant regardless of the placement. Therefore, it is not easy to achieve a flatter temperature profile by simply moving cells around, and the thermal coupling between cells must be captured to avoid hot spots. Recalculating temperature profile for every move and analyzing hot spots caused by thermal coupling can cause lengthy simulation. The approach in [10] incrementally updates the temperature profile as follows. Assuming cell  $a$  is moved from grid point  $i$  to point  $j$ . The change of the power dissipation vector can be written as

$$\mathbf{P}' = [0, \dots, 0, -P_a, 0, \dots, 0, P_a, 0, \dots, 0]^T. \quad (41)$$

The change of the temperature is thus  $\mathbf{T}' = R \times \mathbf{P}'$ . The new temperature profile can then be obtained by adding  $\mathbf{T}'$  to the original temperature profile. A *thermal penalty* is added to the objective function to discourage moves that generate uneven temperature profiles. A possible thermal penalty function is

$$Penalty = \alpha m \left[ \sum_{i=1}^m \max((T_i - T_s)^3, 0) + \beta (T_{max} - T_s)^3 \right] \quad (42)$$

### C. Partitioning-based Placement

Partitioning-based approaches to placement are based on the idea of recursively dividing the layout into regions and assigning cells to each region. The key issue in partitioning-based placement, tackled in [11], is to simplify the thermal model at each level of partitioning to achieve the goal of placing the cells so that  $T$  is evenly distributed across the chip. Equation (38) cannot be directly used, since at each partition level, the only location information that is available is identity of the partitioning blocks that the cell belongs to. Assuming that all the cells belonging to a block are located at its center, the corresponding analysis will correspond to an incorrect thermal analysis of the partition. For example, it is easy to see that this will result in an exaggerated thermal gradient within the partition, since the temperature at the center of the partition will be much higher than that at its periphery.

For simplicity, consider the thermal model for a top-down bipartitioning process; this process can be easily extended to a top-down  $k$ -way partitioning process. At any one particular partition stage, a single block is being partitioned into two sub-blocks so that the cuts between the boundary of sub-blocks are minimized. For now, the actual temperature profile in the sub-blocks is not of direct concern, since cells inside the blocks will be further partitioned later, and this can be considered at that time. However, it is important to minimize the temperature discrepancy between the blocks. If some high-power cells are accumulated into one of the blocks, then at a later stage it will not be possible to move these cells out of the block, due to the divide-and-conquer nature of top-down partitioning. It is reasonable to assume that the temperature inside each of the sub-blocks are uniform, since if such an objective were to be enforced at every step of the partitioning, then a uniform temperature distribution would indeed result. Under this assumption, a simplified thermal model is obtained, and assumptions about the precise cell locations inside the sub-block need not be made.

The problem can also be viewed from the point of view of multigrid methods [34]. It has been known that the Poisson equation that describes the thermal behavior can be solved effectively using multigrid methods. The spatial variation of temperature can be thought of as having ‘‘high frequency’’ and ‘‘low frequency’’ components. A very uniform temperature distribution over space can be thought of as having predominantly ‘‘low frequency’’ components and very small ‘‘high frequency’’ components, and a very widely varying distribution can be considered to show the opposite property. The multigrid method solves (8) on an  $n \times n$  grid by using the following ideas: it builds a series of gradually refined meshes:  $m_1 \times m_1, m_2 \times m_2, \dots, m_k \times m_k$ , where  $m_k = n$ , and each mesh is a coarsened mesh for all the meshes after it. It then solves for the lower ‘‘frequency’’ terms of the spatial distribution of  $T$  on the coarse meshes and interpolates the result onto the refined meshes. This method is based on the observation that lower frequency components of  $T$  can be effectively solved on a coarse mesh.

If the partition lines are limited to the thermal meshes during the top-down hierarchical partitioning, the partition process

can be thought of as a series of operations on a set of gradually refined meshes. At any particular level, we are only concerned about the spatial distribution of the temperature in a specific “frequency range,” and as the mesh is refined further during the top-down partitioning method, higher frequency terms, corresponding to local variations, are incorporated.

In the first step in top down partitioning, the chip is partitioned into two blocks, the left block and the right block. For simplicity, assume that number of thermal cells in each region is the same, although this assumption can easily be discarded. The thermal cells  $i = 1 \dots m/2$  will be said to be in the left block and cells  $i = m/2 + 1 \dots m$  in the right block. Now assume, as stated above, that the block on the left has an even temperature of  $T_l$  and the temperature of the block to the right is  $T_r$ . Equation (38) can now be simplified to:

$$\begin{bmatrix} \sum_{i=1}^{m/2} \sum_{j=1}^{m/2} G_{ij} & \sum_{i=1}^{m/2} \sum_{j=m/2+1}^m G_{ij} \\ \sum_{i=m/2+1}^m \sum_{j=1}^{m/2} G_{ij} & \sum_{i=m/2+1}^m \sum_{j=m/2+1}^m G_{ij} \end{bmatrix} \begin{bmatrix} T_l \\ T_r \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m/2} P_i \\ \sum_{i=m/2+1}^m P_i \end{bmatrix} \quad (43)$$

This reasoning can be extended to a general case where the chip is partitioned into  $k$  regions, each with possibly a different number of cells. Each region  $i, (i = 1, \dots, k)$  has the same temperature  $T'_i$  and contains a thermal cell set  $S_i: s_i^1, s_i^2, \dots, s_i^{n_i}, n_i$  is the number of thermal cells in the region, and  $S_i, (i = 1..k)$  is a  $k$ -way partition of the index set  $\{1, 2, \dots, m\}$ . Equation (38) can then be simplified into the following form:

$$\begin{bmatrix} G'_{11} & G'_{12} & \dots & G'_{1k} \\ G'_{21} & G'_{22} & \dots & G'_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ G'_{k1} & G'_{k2} & \dots & G'_{kk} \end{bmatrix} \begin{bmatrix} T'_1 \\ T'_2 \\ \vdots \\ T'_k \end{bmatrix} = \begin{bmatrix} P'_1 \\ P'_2 \\ \vdots \\ P'_k \end{bmatrix} \quad (44)$$

where  $G'_{ij} = \sum_{i' \in S_i} \sum_{j' \in S_j} G_{i'j'}$  and  $P'_i = \sum_{i' \in S_i} P_{i'}$ . It is easy to show that if  $G$  is positive definite, then so is  $G'$ .

The approach in [11] uses a top-down two-way partitioner based on the Fiduccia-Mattheyses algorithm [35], but can be extended naturally to incorporate a state-of-the-art multi-level partitioner. To reduce the computational cost for incremental temperature updates, the notion of an “effective thermal influence region” of a block is introduced. For a unit heat source on a block, this corresponds to the area outside of which the temperature induced by the unit heat source is less than a certain percentage of the maximum temperature induced by the unit heat source: this can be easily computed once the thermal resistance matrix is known.

The parameter  $\delta T_S$  is used to guide the algorithm:

$$\delta T_S = \max(|T_i - T_{i,neighbours}|), i \in S, \quad (45)$$

where  $T_{i,neighbours}$  are the temperature for cells adjacent to cell  $i$  and  $S$  is a set of blocks.

The process starts with the computation of the conductivity matrix. The top-down bipartitioning starts from the top level block and partition the every block into two blocks. This is done recursively until the number of standard cells contained in each block is less than a certain threshold. At each level, the following steps are performed:

- 1) The simplified thermal conductivity matrix  $G'$  is created, as shown in equation (44).  $G'$  is then inverted to obtain the thermal resistance matrix  $R'$ , since this is required for the incremental update that is to be performed later. Here in the worst case, an  $m \times m$  matrix is inverted, where  $m$  is the number of mesh nodes on the wafer surface.
- 2) Using the thermal influence region concept introduced above,  $R'$  is converted to a sparse matrix  $R'_{sp}$  as follows: each element  $R'_{ij}$  is compared against the diagonal element  $R'_{ii}$ . If  $R'_{ij}/R'_{ii} < \epsilon$ , where  $\epsilon$  is a small number,  $R'_{ij}$  is set to 0. This reduces the computational expense of evaluating moves, with a small loss in accuracy that can be bounded.

At first, multiple solutions are generated randomly while partitioning a block. For each solution,  $\delta T_S$  is computed, where  $S$  is a set of blocks that are adjacent to the blocks being partitioned. If the maximum and minimum values are  $\delta T_{max}$  and  $\delta T_{min}$ , the thermal budget for this partition is set to be  $(1 - \alpha)\delta T_{max} + \alpha\delta T_{min}$ , where  $0 \leq \alpha \leq 1$ . The choice of  $\alpha$  is a tradeoff between partition quality and thermal constraints. The solution with the lowest  $\delta T$  is chosen as the initial solution for partitioning.

When the partitioner decides to move a cell, the thermal constraint will be one of the constraints that will decide whether the move is legal or not. This constraint determines whether move should be accepted or not using the following steps:

- 1) Compute the delta power vector.

$$\Delta P = [0, \dots, 0, \delta P_1, 0, \dots, 0, \delta P_2, 0, \dots, 0, \delta P_l, 0, \dots, 0]$$

where  $l$  is the number of blocks whose power dissipation is affected. This set of blocks includes not only the blocks that the cell is moving from and moving to, but also blocks that contain cells that drive the input of the cell to be moved, since the wire load for these cells will also change.

- 2) Compute the delta temperature vector  $\Delta T = R'_{sp} \Delta P$  and correspondingly update  $\delta T_S$  using Equation (45). Since this involves a sparse matrix-vector multiplication, the computation cost is practically constant.
- 3) If the  $\delta T_S$  is within the current budget, the move is accepted. Otherwise, it is rejected.

#### D. Force-directed Placement

In force-directed methods, an analogy to Hooke’s law is used by representing nets as springs and finding the placement corresponding to the system’s minimum energy state. Attractive forces are created between interconnected cells and are made proportional to the separation distance and interconnectivity. Other design criteria such as cell overlap,

timing, and congestion are used to derived the repulsive forces. After 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.

The work in [4] presents a force-directed approach to thermal placement. The application domain is in the design of 3D integrated circuits, where chips have multiple levels of active devices. Therefore, placement must be carried out in not just the  $xy$ -plane, but the entire  $xyz$ -space in three dimensions. In current technologies, in the  $z$  dimension, the number of layers is restricted to a small number. The work in [4] uses a force-directed framework with FEA-based thermal analysis, using repulsive forces to avoid hot spots. The thermal forces are calculated using the temperature gradient, which itself can be related to the stiffness matrix and its derivative. The temperature gradient determines both the direction and relative magnitude of the thermal forces, thereby moving cells away from areas with high temperature.

Fundamentally, force-directed methodologies involve minimizing an objective function corresponding to a summation of cost components from each net. For 3D layouts, this takes the form

$$c_{ij} [(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2] \quad (46)$$

where  $c_{ij}$  is the weight of the connection between the two nodes. If the  $c_{ij}$  coefficients are combined into a global  $C$  matrix, an objective function can be written for the entire system:

$$\frac{1}{2} \mathbf{x}^T C \mathbf{x} + \frac{1}{2} \mathbf{y}^T C \mathbf{y} + \frac{1}{2} \mathbf{z}^T C \mathbf{z} \quad (47)$$

where  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  are the  $x$ ,  $y$ , and  $z$  coordinates of all cells and points of interest. This objective function can be minimized by solving the following three systems of equations:

$$C \mathbf{x} = \mathbf{f}_x, C \mathbf{y} = \mathbf{f}_y, C \mathbf{z} = \mathbf{f}_z \quad (48)$$

In the absence of external repulsive forces, the total force vectors,  $\mathbf{f}_x$ ,  $\mathbf{f}_y$ , and  $\mathbf{f}_z$ , would be zero. The net stiffness matrix,  $C$ , describes the entire net connectivity. Fixed coordinate values, created by physical constraints, can be used to reduce and solve the system of equations, much like the conductive boundary conditions in FEA.

Generally, an 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.

A sample 3D thermal placement for a four-layer process, generated using this approach, is shown in Fig. 13. The heat sink is placed at the bottom of the 3D chip, and the red regions are hotter than the blue regions. It is clear that the coolest cells are those in the bottom layer, next to the heat sink, and the temperature increases as we move to higher layers. The thermal placement method consciously mitigates the temperature by making the upper layers sparser, in terms of the percentage of area populated by the cells, than the lower layers. Thermal vias are often added to achieve even thermal

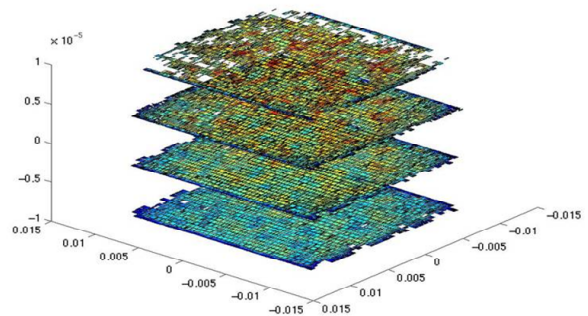


Fig. 13. A 3D thermal placement for a four-layer process using iterative force-directed approach.

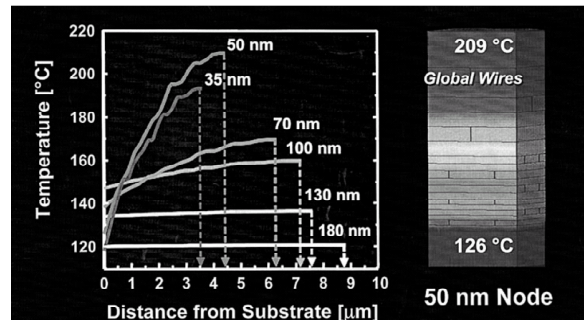


Fig. 14. Maximum temperature distribution along vertical distance from the substrate to the top metal layer [37].

distribution [36], and their effects must be properly modeled for accurate thermal analysis, including resistive effects that locally generate heat.

#### IV. FUTURE THERMAL PROBLEMS AND PHYSICAL DESIGN SOLUTIONS

##### A. Interconnect Thermal Distribution

With technology scaling and the corresponding increase in the number of metal layers, interconnect power dissipation is likely to contribute a significant portion of total chip power consumption, and the contribution of self-heating of interconnect to the total heat generated can become significant. As shown in Fig. 14, self-heating can result in elevated temperature in high metal layers, which are far away from heat sinks in flip-chip designs. Although we can also model

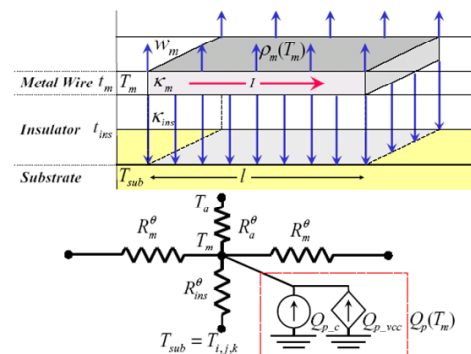


Fig. 15. Interconnect lumped model.

interconnects as lumped circuit elements, as shown in Fig. 15, explicitly modeling every interconnect is not practical. Further research is needed to efficiently and accurately analyze the impact of thermal issues due to interconnects.

A second aspect of interconnect is that it has a much larger heat conduction coefficients than the interlayer dielectric, or the insulator in silicon on insulator (SOI) devices: these effects become worse with the use of low-k dielectrics. To counter this, interlayer thermal vias may be inserted to act as heat pipes in the chip: if the temperature difference between both ends of an interconnect is sufficiently large, these serve to redistribute heat through the volume of the chip. Although early work has been carried out on thermal via insertion, it remains an interesting research problem.

### B. Floorplanning

Thermal planning may be carried out throughout the physical design process. Prior to placement, this may commence at the floorplanning stage, early in the design cycle. The tradeoff here is that there is a greater deal of flexibility in mitigating thermal problems, but also a larger amount of uncertainty with regard to the precise thermal profile, as the design is still far from complete. These floorplanning approaches may need to consider both static and dynamic thermal issues.

### C. Effects of Temperature on Performance and Reliability

The temperature profile can change when the workload of a chip changes. Since the performance of each transistor is sensitive to the operating temperature, a change in the temperature profile can cause path delay variations, and possibly timing violations, both on signal paths and clock paths. On-chip clock tuning is a possible solution to avoid temperature related timing violations: by controlling the clock arrival times of the flip-flops on temperature sensitive paths through programmable delay elements, the chip can be made to function correctly at a wide range of temperatures, without sacrificing performance. On signal paths, thermal effects must be taken into account during timing analysis and optimization. Finally, thermally-driven aging effects such as negative bias temperature instability (NBTI) must also be incorporated as a first-level design consideration.

### D. Packaging

Several new packaging techniques have been proposed to alleviate the interconnect delay dominance problem in deep submicron technologies. 3D multichip modules, wafer bonding, and 3D IC (in both silicon and packaging) are three examples of such solutions. Both passive and active solutions may be used: passive solutions include the employment of thermal vias within the package for heat conduction, while active cooling may employ advanced cooling and packaging techniques, for example, for spot cooling. Active cooling techniques, for the most part, are not economically viable today, but this may change as technology progresses. Novel research techniques are necessary to incorporate these effects for chip-package thermal codesign.

### E. Three-dimensional Circuits

Three dimensional (3D) circuit technologies offer an intriguing possibility for the future, and several practical 3D strategies have been developed both in industry [38] and in academia [39]. By vertically stacking layers of active devices, 3D technologies succeed in significantly reducing the lengths of on-chip wires, and hence the delays. However, this also results in packing a larger number of devices, or heat sources, per unit volume, resulting in aggravated thermal problems. Challenges in 3D design include placement, floorplanning and thermal via insertion and several techniques have been proposed to solve this issue [4], [36].

### REFERENCES

- [1] Y. K. Cheng, P. Raha, C. C. Teng, E. Rosenbaum, and S.-M. Kang. ILLIADS-T: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of CMOS VLSI chips. *IEEE Transactions on Computer-Aided Design*, 17(8):668–681, August 1998.
- [2] Mobile Pentium III processors - Intel SpeedStep technology.
- [3] 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*, 21(12):1434–1445, December 2002.
- [4] 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*, pages 86–89, 2003.
- [5] T.-Y. Wang and C. C.-P. Chen. SPICE-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis base on model order reduction. In *Proceedings of the International Symposium on Quality Electronic Design*, 2004.
- [6] H. Qian, S. R. Nassif, and S. S. Sapatnekar. Random walks in a supply network. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 93–98, 2003.
- [7] H. Qian and S. S. Sapatnekar. Hierarchical random-walk algorithms for power grid analysis. In *Proceedings of the Asia/South Pacific Design Automation Conference*, pages 499–504, 2003.
- [8] Y. Zhan and S. S. Sapatnekar. Fast computation of the temperature distribution in VLSI chips using the discrete cosine transform and table look-up. In *Proceedings of the Asia/South Pacific Design Automation Conference*, pages 87–92, 2005.
- [9] C. Chu and D. F. Wong. A matrix synthesis approach to thermal placement. *IEEE Transactions on Computer-Aided Design*, 17(11):1166–1174, November 1998.
- [10] C. Tsai and S. M. Kang. Cell-level placement for improving substrate thermal distribution. *IEEE Transactions on Computer-Aided Design*, 19(2):253–266, February 2000.
- [11] G. Chen and S. S. Sapatnekar. Partition-driven standard cell thermal placement. In *Proceedings of the International Symposium on Physical Design*, pages 75–80, 2003.
- [12] M.N. Ozisik. *Boundary Value Problems of Heat Conduction*. Dover, New York, NY, 1968.
- [13] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactions on Computer-Aided Design*, 9(4):352–366, April 1990.
- [14] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 58–65, 1997.
- [15] Yahong Cao, Yu-Min Lee, Tsung-Hao Chen, and Charlie Chung-Ping Chen. HiPRIME: hierarchical and passivity reserved interconnect macromodeling engine for RLKC power delivery. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 379–384, 2002.
- [16] J. M. Wang and T. V. Nguyen. Extended krylov subspace method for reduced order analysis of linear circuits with multiple sources. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 247–252, 2000.
- [17] C. Ho, A. E. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, June 1975.
- [18] D. L. Logan. *A First Course in the Finite Element Method*. Brooks/Cole Pub. Co., 3 edition, 2002.

- [19] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, Washington, DC, 1984.
- [20] R. D. Yates and D. J. Goodman. *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. John Wiley and Sons, New York, NY, 1999.
- [21] R. Gharpurey and R. G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. *IEEE Journal of Solid-State Circuits*, 17(4):305–315, March 1996.
- [22] Y. Zhan and S. S. Sapatnekar. A high efficiency full-chip thermal simulation algorithm. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 634–537, 2005.
- [23] K. Banerjee, A. Mehrotra, A. Sangiovanni-Vincentelli, and C. Hu. On thermal effects in deep sub-micron VLSI interconnects. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 885–891, 1999.
- [24] W. Sun and C. Sechen. Efficient and effective placement for very large circuits. *IEEE Transactions on Computer-Aided Design*, 14(3):349–359, March 1995.
- [25] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. GORDIAN: VLSI placement by quadratic programming and slicing optimization. *IEEE Transactions on Computer-Aided Design*, 10(3):356–365, March 1991.
- [26] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. In *Proceedings of the International Symposium on Physical Design*, pages 18–25, 2004.
- [27] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 269–274, 1998.
- [28] K. Vorwerk, A. Kennings, and A. Vannelli. Engineering details of a stable force-directed placer. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 573–580, 2004.
- [29] C. J. Alpert, T. Chan, A. B. Kahng, I. Markov, and P. Mulet. Faster minimization of linear wirelength for global placement. *IEEE Transactions on Computer-Aided Design*, 17(1):3–13, January 1998.
- [30] C. Alpert, J.-H. Huang, and A. B. Kahng. Multilevel circuit partitioning. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 530–533, 1997.
- [31] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: application in VLSI design. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 526–529, 1997.
- [32] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable placements? In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 477–482, 2000.
- [33] T. Chan, J. Cong, T. Kong, and J. Shinnerl. Multilevel optimization for large-scale circuit placement. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 171–176, 2000.
- [34] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, New York, NY, 2001.
- [35] C. M. Fiduccia and R. M. Mattheyses. A linear time heuristic for improving network partitions. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 175–181, 1982.
- [36] B. Goplen and S. S. Sapatnekar. Thermal via placement in 3D ICs. In *Proceedings of the International Symposium on Physical Design*, pages 167–174, 2005.
- [37] S. Im and K. Banerjee. Full chip thermal analysis of planar (2-D) and vertically integrated (3-D) high performance ICs. In *Technical Digest of IEEE International Electron Devices Meeting*, pages 727–730, 2000.
- [38] 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. Purushothaman, and W. E. Haensch. Electrical integrity of state-of-the-art 0.13  $\mu\text{m}$  SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication. In *IEEE International Electron Devices Meeting*, pages 943–945, 2002.
- [39] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep submicron interconnect performance and systems-on-chip integration. *Proceedings of the IEEE*, 89(5):602–633, May 2001.