3-D Thermal Simulation with Dynamic Power Profiles

Eunjoo Choi
Dept. of Electrical Engineering, KAIST
Daejeon 305-701, Korea

Youngsoo Shin
Dept. of Electrical Engineering, KAIST
Daejeon 305-701, Korea

***Abstract*— On-chip temperature and temperature gradient have been emerging as important design criteria as technology is scaled down to nano-meter regime. There have been several approaches to analyze or simulate the thermal behavior of chips, but all the approaches assume constant average power consumption of each block, which is reasonable when the change in power is localized and transient. However, as the aggressive power management techniques are employed in block level of granularity, power consumption of blocks become fluctuating a lot, which yields a large error with the conventional thermal analysis. A 3-D thermal simulation, with time-varying power consumption of blocks, is proposed in this paper. The partial differential heat conduction equation is solved with finite difference method, and we also employ alternating direction implicit method to decrease the computational complexity. The prototype simulator was designed and tested on several examples.**

I. Introduction

On-chip temperature of VLSI circuits has been growing with increasing integration density and corresponding power density. Subthreshold leakage current, which is a main component of power consumption in nano-meter CMOS technologies such as 90- and 65-nm, increases exponentially with temperature while temperature itself increases with subthreshold leakage, which implies a positive feedback loop between subthreshold leakage and temperature, and thus may lead to thermal runaway. Growing temperature not only causes timing failure but also degrades chip reliability due to effects such as electromigration (EM). On-chip temperature gradient poses a challenge to traditional design methodology, which assumes constant temperature across a chip. It is reported that cross-chip temperature in a high performance microprocessor can vary about 30°C [1]. Therefore, accurate yet fast analysis or simulation of on-chip temperature is very important in designing modern VLSI circuits.

There have been several approaches to thermal analysis: the finite difference method with equivalent thermal RC model [2], 3-D transient thermal analysis based on the alternating direction implicit method [3], and so on. However, all these approaches assume, for each block on a chip, constant power corresponding to average power consumption over time of interest. The rationale behind this is that temperature varies very slowly (in the order of *ms*), thus any transient local change in block-level power is filtered out in chip-level after temperature-varying scale of time.

Due to employment of aggressive power management techniques such as power gating, reverse body bias, and dynamic voltage scaling [4], power profile of each block can vary significantly over time, and thus cannot be treated as local change. This can yield large errors in conventional thermal analysis based on average power consumption. Fig. 1(a) shows an example floorplan with dynamic power profile of each block. If we take an average of power over 10 *ms* period of time for each block, and use it for conventional thermal analysis, Fig. 1(b) shows a thermal map after the period of time. On the other hand, Fig. 1(c) corresponds to a thermal map if power profiles themselves are used for thermal analysis (thus, exact). The errors of the first map are clearly seen in the figure, which range from 10°C to 30°C.

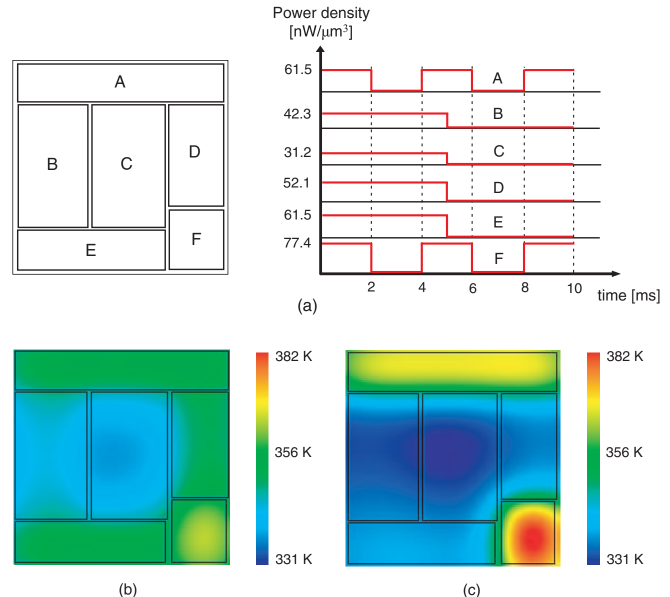

Fig. 1. (a) Example floorplan with block-level power profiles, (b) thermal map after 10 *ms* with constant average power densities, and (c) thermal map after 10 *ms* with dynamic power profiles.

In this paper, we propose a 3-D thermal simulator, which accepts, as an input, time-varying dynamic power profiles of blocks. The partial differential heat conduction equation is then solved with the finite difference method, which approximates a 3-D space at a number of discrete points. We also employ the alternating direction implicit method to decrease the computational complexity. The prototype simulator was designed and tested on several examples. Our thermal simulator could

978-1-4244-1684-4/08/$25.00 ©2008 IEEE

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on December 10, 2009 at 00:17 from IEEE Xplore. Restrictions apply.

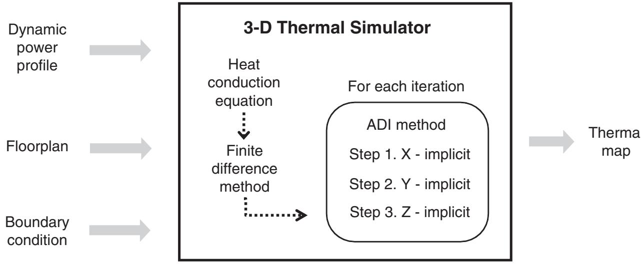

Fig. 2. Overview of 3-D thermal simulator with dynamic power profiles.

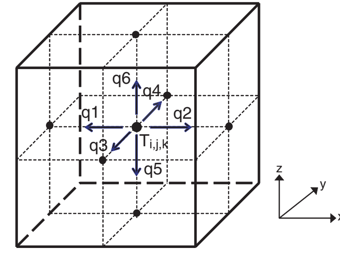

Fig. 3. 3-D view of grid point (i, j, k) with heat flow.

be integrated with analysis of performance, floorplan, and power [5], which then provides a framework for architecture design of large-scale designs such as SoC.

The remainder of this paper is organized as follows. In the next section, we overview the proposed thermal simulator and the heat conduction equation, which is the main equation for the simulator. In Section III, the finite difference method and the alternating direction implicit method, which form the basis of the simulator, will be described. Section IV presents the experimental results, and we draw conclusions in Section V.

II. Thermal Simulation with Dynamic Power Profiles

Fig. 2 shows the overview of our 3-D thermal simulator. The dynamic power profiles of blocks in a design with a floorplan are the inputs to the simulator. We take ambient temperature and convective heat transfer coefficient, which constitute a boundary condition, as another input to the simulator.

From these inputs, the simulator tries to solve the heat conduction equation, which is the governing equation for calculation of heat conduction and temperature [6], [7]:

$$\rho C_p \frac{\partial T(x,y,z,t)}{\partial t} = \nabla \cdot [\kappa(x,y,z,t)\nabla T(x,y,z,t)] + g(x,y,z,t), \quad (1)$$

where T is the temperature $[K]$ which we try to obtain, and g is the power density of a heat source $[W/m^3]$, which is derived from chip floorplan and dynamic power profiles (see Fig. 1). κ denotes the thermal conductivity $[W/m \cdot K]$, ρ is the material density $[kg/m^3]$, and C_p is the specific heat $[J/kg \cdot K]$. Note that we assume κ is independent of temperature, which is a reasonable assumption in practice, since the change of thermal conductivity over temperature is usually very small. Physically, (1) implies that the energy stored in a volume V is equal to the sum of heat entering V through its boundary surface and the heat generated in itself.

The heat conduction equation is subject to the general boundary condition:

$$\kappa(x,y,z,t)\frac{\partial T(x,y,z,t)}{\partial n_i} + h_i T(x,y,z,t) = f_i(x,y,z,t), \quad (2)$$

where h_i denotes the heat transfer coefficient $[W/m^2 \cdot K]$, n_i is the outward direction normal to the boundary surface i, and f_i is a function that represents the boundary condition. For convective boundary condition, f_i is equal to $h_i T_\infty$, where T_∞ denotes the ambient temperature. It has to be noted that the air is not a direct boundary of the chip due to the package and heat sink surrounding the chip. This intermediate layer is modeled as one dimensional equivalent thermal resistance and, in turn, can be modeled as the effective heat transfer coefficient.

The partial differential equation (1) can be solved by numerical methods such as the finite difference method (FDM) or finite element method (FEM), both of which discretize the continuous space domain into a finite number of grid points [8] as shown in Fig. 3. In our thermal simulator, we use FDM approach, since it is simpler to formulate and can readily be extended to 2- or 3-dimensional problems. Due to an enormous number of grid points used in FDM, especially when accuracy of simulation is important, the complexity of FDM alone could be very high. We employ alternating direction implicit (ADI) method to alleviate the complexity.

III. Simulation Procedure

A. Finite Difference Method in Heat Transfer

Unlike FEM, which is an approximation to the solution of a differential equation, FDM is an approximation of the differential equation itself. A formal basis for developing finite approximation to derivatives is through the use of Taylor series expansion. For example, the FDM approximation of the second order partial derivative of T at time step n, denoted by T^n, with respect to x can be expressed by:

$$\left.\frac{\partial^2 T^n}{\partial x^2}\right|_{i,j,k} \approx \frac{T^n_{i+1,j,k} - 2T^n_{i,j,k} + T^n_{i-1,j,k}}{(\Delta x)^2} \triangleq \frac{\delta_x T^n}{(\Delta x)^2}, \quad (3)$$

where Δx corresponds to the discretization interval in the x-axis. Note that the truncation error associated with this approximation is $O[(\Delta x)^2]$.

By applying (3) to the right-hand side of (1) and applying the first order partial derivative of T with respect to t, which is similar to (3) in its form, the discretized heat conduction equation can be obtained. In the FDM, we have freedom to choose either T^n or T^{n+1} for the second order derivative with respect to x, y, and z (right-hand side of (1)). If we take T^n, it is called a simple explicit method; by taking T^{n+1}, we have a simple implicit method. Since both methods have drawbacks, we instead take the combined method, which is called Crank-Nicolson method [8], in our FDM approach:

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on December 10, 2009 at 00:17 from IEEE Xplore. Restrictions apply.

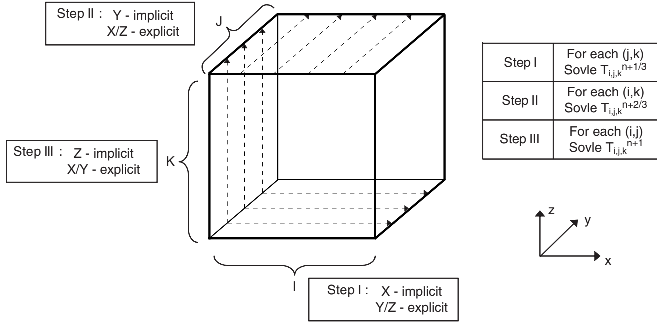

Fig. 4. 3-D ADI method.

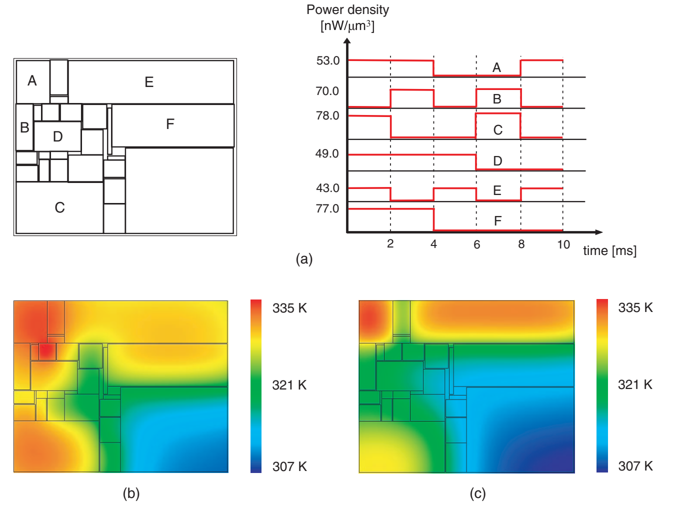

Fig. 5. (a) Floorplan of an example chip [11] with dynamic power profiles, (b) thermal map with constant average power densities, and (c) thermal map with dynamic power profiles.

$$\frac{T^{n+1}-T^n}{\Delta t} = \alpha\left[\frac{\delta_x T^{n+1}+\delta_x T^n}{2(\Delta x)^2}+\frac{\delta_y T^{n+1}+\delta_y T^n}{2(\Delta y)^2}+\frac{\delta_z T^{n+1}+\delta_z T^n}{2(\Delta z)^2}\right]+\frac{g^{n+1}+g^n}{2\rho C_p}, \quad (4)$$

where α corresponds to $\frac{\kappa}{\rho C_p}$.

For non-homogeneous case, or when a grid point is surrounded by more than two materials, the heat conduction equation is derived by the law of energy conservation:

$$G = \frac{dE}{dt}+q_1+q_2+q_3+q_4+q_5+q_6, \quad (5)$$

where dE/dt is the energy stored in volume $\Delta V = \Delta x\Delta y\Delta z$, G denotes the heat generated within ΔV, and q_i is the outgoing heat from the node as shown in Fig. 3.

B. Alternating Direction Implicit Method

The Crank-Nicolson method guarantees stability in its computation, but its computational complexity grows fast with dimension. For instance, in 3-D problems, $N^3 \times N^3$ matrix, where N is the number of grid points, has to be solved for each direction and for each time step. To alleviate the complexity, the alternating direction implicit (ADI) method has been developed [9], [10]. The ADI method transforms the problem into solving $N \times N$ matrix N^2 times for each time step and for each direction of x, y, and z.

The time step from n to $n+1$ is split into three steps: from n to $n+1/3$, from $n+1/3$ to $n+2/3$, and from $n+2/3$ to $n+1$ as shown in Fig. 4. In step I, x direction is implicit while the other directions are explicit. Similarly, y and z directions are implicit in steps II and III, respectively.

$$\begin{aligned}
\text{Step I}: \quad T^{n+1/3}-T^n &= G+r_x(\delta_x T^{n+1/3}+\delta_x T^n) \\
&\quad +2r_y(\delta_y T^n)+2r_z(\delta_z T^n) \\
\text{Step II}: \quad T^{n+2/3}-T^n &= G+r_x(\delta_x T^{n+1/3}+\delta_x T^n) \\
&\quad +r_y(\delta_y T^{n+2/3}+\delta_y T^n) \\
&\quad +2r_z(\delta_z T^n) \qquad (6)\\
\text{Step III}: \quad T^{n+1}-T^n &= G+r_x(\delta_x T^{n+1/3}+\delta_x T^n) \\
&\quad +r_y(\delta_y T^{n+2/3}+\delta_y T^n) \\
&\quad +r_z(\delta_z T^{n+1}+\delta_z T^n)
\end{aligned}$$

where $G = g\cdot\Delta t/(\rho C_p)$, $\beta = 2\cdot\Delta t/(\rho C_p)$, $r_x = \beta/2(\Delta x)^2$, $r_y = \beta/2(\Delta y)^2$, and $r_z = \beta/2(\Delta z)^2$. In each step, each node (i,j,k) is related to three unknown variables: $T_{i-1,j,k}^{n+1/3}, T_{i,j,k}^{n+1/3}$, and $T_{i+1,j,k}^{n+1/3}$ in step I, for example. Therefore, it is a tridiagonal matrix and can be solved by Thomas algorithm [8] with time complexity $O(N)$, where N is the number of grid points.

For the boundary conditions, the concept of virtual node should be introduced. By applying FDM to (2), the equation for boundary nodes can be derived. For example, the boundary condition at $x = 0$ can be expressed by:

$$-\kappa\frac{T_{1,j,k}^n - T_{-1,j,k}^n}{2\Delta x}+h_{x^-}T_{0,j,k}^n = h_{x^-}^e T_\infty. \quad (7)$$

If (7) is rearranged for $T_{-1,j,k}^n$ and if it is considered as a real grid point at $x=-1$, the temperature at boundary $x = 0$ can be obtained. The other virtual points $T_{I+1,j,k}^n, T_{i,-1,k}^n, T_{i,J+1,k}^n, T_{i,j,-1}^n, T_{i,j,K+1}^n$ can be derived similarly.

IV. Experiments

Our proposed 3-D thermal simulator with dynamic power profiles was implemented with C language under SunOS 5.8. We performed experiments on two test chips, one taken from [11] and the other from [12]. The former example is a unit-level layout of a prototype ULSI high-performance chip. The number of grid points is $250 \times 200 \times 20$ with $\Delta x = \Delta y = \Delta z = 20\ \mu m$, $\Delta t = 100\ \mu s$, and $T_\infty = 300\ K$. The effective heat transfer coefficients were taken from [3], and the simulation was performed for a period of 10 *ms*, which corresponds to 100 iterations. The floorplan with block-level power profiles is shown in Fig. 5(a). The blocks, which are not marked in the floorplan, were assumed to be active for the first 5 *ms* consuming a constant average power, and then to be idle for the last 5 *ms* consuming no power. Fig. 5(b) and (c) show the thermal maps when we assumed a constant average power density over 10 *ms* for each block (thus, conventional

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on December 10, 2009 at 00:17 from IEEE Xplore. Restrictions apply.

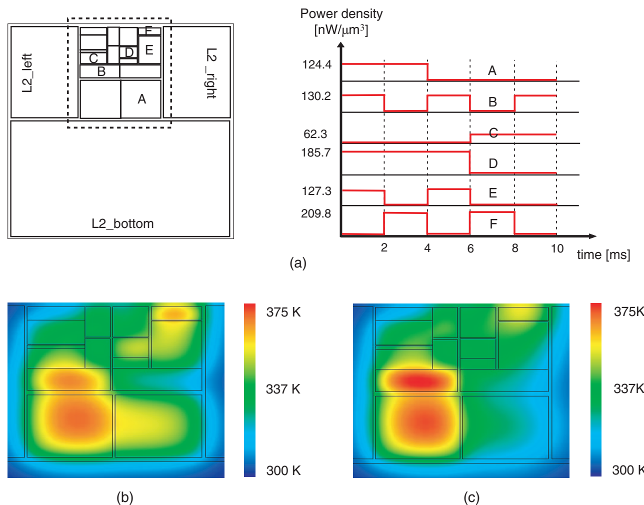

Fig. 6. (a) Floorplan of the Alpha processor [12] with dynamic power profiles, (b) thermal map with constant average power densities, and (c) thermal map with dynamic power profiles.

approach) and when we used the dynamic power profiles themselves, respectively.

All the blocks of the test chip, under the conventional thermal analysis, continuously generate heat over the period of simulation; they actually stop generating heat when they are idle, thus get cooler by conduction toward the cooler sides due to the nature of heat flow. This is the main reason why the thermal map in Fig. 5(b) has errors compared to that of Fig. 5(c). Especially, the blocks B, C, D, and F have higher temperature at the end of the simulation in Fig. 5(b) than they really have to have in Fig. 5(c). This is because they are idle for the last 2 *ms* (see Fig. 5(a)), thus should have a chance to get cooler, but it is not reflected in the conventional thermal simulation of Fig. 5(b), due to assumption of constant average power consumption. Similarly, the blocks A and E have lower temperature in Fig. 5(b) than they should have (Fig. 5(c)). This is because they are active for the last 2 *ms* with relatively high power densities, which is not reflected in the conventional simulation.

The second example is the Alpha processor from SPEC2000 benchmark, whose floorplan and power profiles of blocks are shown in Fig. 6(a). For this example, the number of grid points is $400 \times 400 \times 10$ with $\Delta x = \Delta y = \Delta z = 40$ μm, $\Delta t = 100$ μs, and $T_\infty = 300$ K. We used the same effective heat transfer coefficients as the first example. The power densities of the blocks in the Alpha processor were extracted from [12]. Fig. 6(a) shows the floorplan with block-level power profiles. The other blocks, for which power profiles are not shown, are assumed to be active over whole period of 10 *ms*. The thermal maps for the blocks without L2 cache are shown in Fig. 6(b) and (c) with conventional and our thermal simulation, respectively. The errors from the conventional approach are smaller than those in the first test case, since we assumed the constant power consumption for all the blocks which are not marked. However, we can still identify the errors in the blocks of B and F, for example.

From the experiments, it is clear that the conventional thermal analysis with constant power consumption can yield large errors, especially when the blocks have large variation of power. The simulation time with our simulator was 40 min. and 55 min., respectively, and found to be roughly proportional to the number of total grid points, implying a trade-off between the simulation time and spatial accuracy.

V. Conclusion

Although several approaches have been developed for thermal analysis and simulation, they all assume constant average power consumption of each block of a chip, which is only reasonable when the change in power is localized and transient. The aggressive power management techniques, which have become prevalent in designing VLSI circuits, yield large fluctuation in block-level power consumption, and are sources of large errors with conventional thermal analysis. A 3-D thermal simulation, with time-varying power consumption of blocks, has been proposed. The finite difference method, combined with alternating direction implicit method, was employed to solve the partial differential heat conduction equation. The prototype simulator was designed and tested on several examples.

Acknowledgement

This work was supported by Samsung Electronics.

References

[1] P. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allrnon, "High performance microprocessor design," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, May 1998.

[2] Y.-K. Cheng, P. Raha, C.-C. Teng, E. Rosenbaum, and S.-M. Kang, "ILLIAS-T: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of CMOS VLSI chips," *IEEE Trans. on Computer-Aided Design*, vol. 17, pp. 668–681, Aug. 1998.

[3] T.-Y. Wang and C. C.-P. Chen, "3-D thermal-ADI: a linear time chip level transient thermal simulator," *IEEE Trans. on Computer-Aided Design*, vol. 21, no. 12, pp. 1434–1445, Dec. 2002.

[4] S. G. Narendra and A. Chandrakasan, Eds., *Leakage in Nanometer CMOS Technologies*. Springer, 2005.

[5] R. Bergamaschi, Y. Shin, N. Dhanwada, S. Bhattacharya, W. Dougherty, I. Nair, J. Darringer, and S. Paliwal, "SEAS: a system for early analysis of SoCs," in *Proc. Int. Conf. on Hardware/Software Codesign and System Synthesis*, Oct. 2003, pp. 150–155.

[6] F. P. Incropera and D. P. DeWitt, Eds., *Fundamentals of Heat and Mass Transfer*, 4th ed. John Wiley & Sons, 1996.

[7] J. P. Holman, Ed., *Heat Transfer*, 8th ed. McGraw-Hill, 1997, ch. 1–4.

[8] M. N. Ozisik, Ed., *Finite Difference Methods in Heat Transfer*. New York: CRC, 1994.

[9] D. W. Peaceman and J. H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *Journal of the society for industrial and applied mathematics*, vol. 3, no. 1, pp. 28–41, Mar. 1955.

[10] J. J. Douglas and J. E. Gunn, "A general formulation of alternating direction methods–Part I: Parabolic and hyperbolic problems," *Numer. Math.*, vol. 6, pp. 428–453, 1964.

[11] Y. K. Cheng, C. H. Tsai, C. C. Teng, and S. M. Kang, Eds., *Electrothermal Analysis of VLSI Systems*, 1st ed. Springer, 2000.

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

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on December 10, 2009 at 00:17 from IEEE Xplore. Restrictions apply.