PAPER *Special Section on VLSI Design and CAD Algorithms*

On-Chip Thermal Gradient Analysis and Temperature Flattening for SoC Design**

Takashi SATO[†a], *Member*, Junji ICHIMIYA[††*], Nobuto ONO[†††], Koutaro HACHIYA[††††], *Nonmembers*, *and* Masanori HASHIMOTO[†††††], *Member*

SUMMARY This paper quantitatively analyzes thermal gradient of SoC and proposes a thermal flattening procedure. First, the impact of dominant parameters, such as area occupancy of memory/logic block, power density, and floorplan on thermal gradient are studied quantitatively. Temperature difference is also evaluated from timing and reliability standpoints. Important results obtained here are 1) the maximum temperature difference increases with higher memory area occupancy and 2) the difference is very floorplan sensitive. Then, we propose a procedure to amend thermal gradient. A slight floorplan modification using the proposed procedure improves on-chip thermal gradient significantly.

key words: *thermal simulation, thermal gradient, temperature flattening, clock skew, reliability, timing*

1. Introduction

As device density increases with scaling, higher power consumption and temperature have been rapidly becoming crucial for the system on a chip (SoC) design. Thermal management techniques as well as simulation techniques are proposed for high-end processors [1]–[3], but thermal impacts on SoC design have not been studied thoroughly yet. Since ordinary SoC and high-end microprocessors are very different in various ways, especially in packaging and heat removal cost, the techniques tailored for processors may not be adequate for SoC designs. Studying thermal impact considering appropriate packaging environment for SoC is required.

Temperature variation on a chip affects gate and wire delay characteristic, thus causes fluctuation of signal timing. Specifically, long wires used in a global clock tree

---

suffer increased clock skew and performance degradation [4]. Calculating temperature distribution enables instance-based temperature-dependent timing simulation. Nevertheless, obtaining a block placement or power consumption distribution that yields best or worst thermal gradient is not a trivial task. From timing design standpoint, it is much useful to exploit constant temperature distribution through intended placement or architectural techniques [3]. Significant rework is required if timing problem is found in the verification phase considering temperature distribution.

Thermal flattening will become a key technology for more advanced process nodes. It not just achieves smaller timing variation but it also improves reliability since flattening eventually reduces the maximum temperature. Higher temperature exponentially shortens EM lifetime [5], [6]. Moreover, flattened temperature contributes to accurate performance predictability because it matches conventional worst-case simulation scenario, in which applies one worst temperature to all instances.

To realize flattened temperature distribution, it is crucially important to study floorplan design varying key parameters such as where to or not to place 'hot' logic circuits. However, as far as the authors know of, there is no paper which presents quantitative analysis targeted for SoC. Prior researches analyze local thermal distribution accurately but considering a limited number of wires [4]. Efficient calculation for thermal distribution is proposed in [7] but it assumes homogeneity of material that cannot be easily adopted for more complex problems. More general methods for faster thermal simulation are proposed in [8], [9] for final verification purposes. Papers [10], [11] modified force-directed placement [12] considering temperature as additional force to achieve flat thermal distribution in the cell placement.

Novel contributions of this paper are summarized as follows:

- analyzes thermal impacts and clarifies the necessity of considering temperature effects in SoC design.
- proposes temperature-flattening procedure that incrementally modifies floorplan to improve performance such as timing and reliability.

This paper is organized as follows: the first half of this paper examines impacts of design parameters (memory/logic area occupancy, power density, and floorplan) on thermal gradient, delay and reliability in Sects. 2 and 3. Then in Sect. 4, we propose a procedure to flatten thermal

gradient in floorplan phase. Section 5 concludes this paper.

2. Numerical Analysis of On-Chip Thermal Gradient for SoC

2.1 Modeling for Thermal Simulation

In our analysis, we utilize finite-difference approach similar to [13] to solve the heat diffusion equation based on electrical-thermal analogy [14]. Figure 1 shows the model structure. A die as well as packaging material is discretized by 3-D grids to form a cell which mutually connects to adjacent cells in x, y, and z directions. Each cell consists of thermal resistance (R_x, R_y, R_z) and capacitance (C_p) as depicted in Fig. 1(b). The idea of the model parameter calculation is illustrated in Fig. 2. Shaded boxes in the figure stand for metal wires in a cell. For accurate and quick estimation of a combined thermal resistance of distributed metals and the inter-metal dielectric (IMD), we categorize components in a cell into three: i) wires that pass through a cell, ii) wires that are separated by IMD (wires that are shorter than the cell edge length), and iii) the IMD. When we assume uniform wire distribution and use relatively large cell size, the type ii) wires can be packed into triangle, rectangle or trapezoid, without changing horizontal and vertical thermal resistance significantly, as shown in Fig. 2(b). This simplifies equivalent thermal resistance extraction from both statistical information of wire length distribution based on analytical models such as in [15] or actual designs. After concentrating wires, thermal resistance between ports P1 and P2 is easily calculated as combined resistance of $R_1, R_2 + R_3$, and R_4, where R_1 represents IMD, $R_2 + R_3$ represents the IMD and the metal wire, and R_4 is metal wire only. The wires which run through the cell (R_4) and the wires which do not (R_3) are distinguished in the model to capture thermal resistance accurately. As a result, total horizontal thermal resistance of a metal layer R_{P1-P2} is expressed as

$$R_{P1-P2} = \frac{R_1(R_2 + R_3)R_4}{(R_2 + R_3)R_4 + R_1R_4 + R_1(R_2 + R_3)}. \quad (1)$$

The equivalent resistances for y and z directions are calculated in the same manner. C_p consists of the sum of metal wire and insulation material thermal capacitance calculated from their volume and specific heat. Thermal resistance of the package is usually much larger than that of a silicon die therefore it dominantly determines chip temperature. In this paper, we use a package of 2.7 K/W thermal resistance as a typical package of high-end SoC.

2.2 Example Thermal Property

Table 1 shows an example of layer stack-up and corresponding thermal characteristics. Here the size of the die is 10 mm square each for x, y-direction. The dimensions are determined by referring 90-nm node process in the ITRS [16]. The LSI is divided into 16 in x, y and 9 in z direction respectively creating 2304 cells in total. Wire material is copper and dielectrics (IMD and inter-layer dielectric (ILD)) are both SiO_2. Wire and ILD are modeled as 2 layers ('wire1' and 'wire2'). Both consist of 4 metal and 4 ILD layers.

2.3 Thermal Simulation

We use DC analysis function available in SPICE-type circuit simulators [17] to calculate steady-state temperature. As a stimulus, independent current sources are attached to 'sub_s' layer cells to represent power consumption on a chip since transistors consume most of the power. Environment temperature is expressed using an independent voltage source. Accordingly, top nodes in the bump layer cells and bottom nodes in the package layer cells are connected to the positive node of the voltage source.

We assume that the allowed maximum junction temperature $T_{j_{max}}$ for this LSI is 120°C and the environment temperature is 27°C throughout the analysis. Total power consumption of the chip is 32 W which is the largest power allowed for the combination of assumed package and $T_{j_{max}}$. Maximum total power is obtained by considering the chip as a point (this is equivalent to using average temperature as the chip temperature ignoring the gradient). Further breakdown is 20% of the total power is for I/O circuits, and 80% is for memory and logic circuit area.

(a) Chip discritization.   (b) A cell construction.
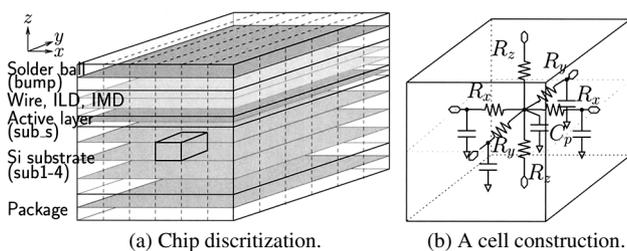
Fig. 1 Thermal simulation model used in this analysis.

(a) Example wire distribution.   (b) Simplification for modeling.
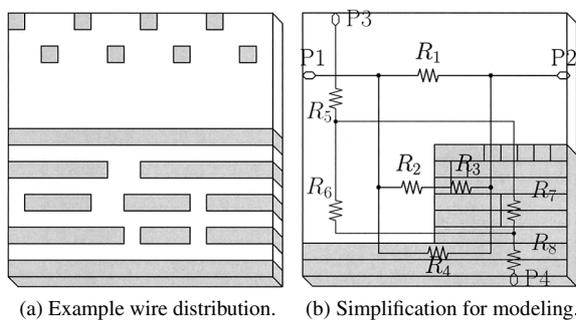
Fig. 2 Equivalent thermal resistance calculation for x, y directions.

Table 1 Example layer division and thermal properties.

Layer	Thickness (μm)	R_x (K/W)	R_y (K/W)	R_z (K/W)	C_p (J/K)
package	200	2.7e4	2.7e4	2.72e3	9.81e-5
sub1~4	125	63.49	63.5	2.54	8.00e-5
sub_s	2	9.0e3	1.8e4	0.037	1.56e-6
wire1~2	3.1	7.8e3	7.8e3	0.93	5.39e-6
bump	200	5.0e4	5.0e4	1.0e3	1.69e-5
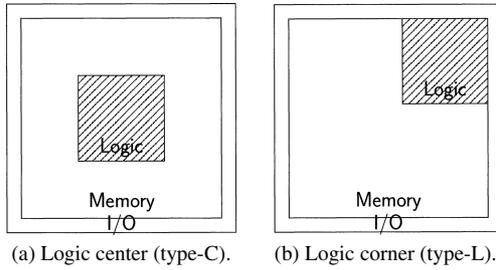
Fig. 3   Two different floorplan strategies for logic circuit placement.

Fig. 4   Temperature dependency of a gate delay (130-nm industrial process, clock buffer).

Fig. 5   Memory area occupancy dependency of on-chip temperature.

Fig. 6 ΔT_{max} as functions of package and chip power consumption.

2.4   Varying Parameters

Three parameter dependencies are investigated in the following section.

Circuit functionality dependency: In recent SoC, memory area occupancy has increasing trend—ITRS expects it becomes 75% in year 2003 and 93% in 2012 [16]. Since there is usually substantial difference in the power density between memory and logic, their ratio should affect temperature distribution. Here, we define memory area occupancy α_m by excluding I/O circuit area, i.e. α_m =(memory area)/(logic area + memory area).

Floorplan and power density dependency: Chip floorplan is another parameter. Due to big difference in power density, placement of memory and logic may have a big influence on the thermal profile. Figure 3 shows two different floorplans: type-C places logic at the center of the die, and type-L places logic at the corner.

Block partitioning and placement dependency: Starting from type-C or type-L as initial placement, we vary block partition and placement at the same time to find the optimal floorplan of the blocks.

3.   Thermal Simulation Results

In this section, we present simulation results to analyze qualitative importance of the key parameters on thermal profile in SoC design.

3.1   Temperature Characteristics of Gate and Wire Delay

Gate and wire delay change were calculated through SPICE simulations. As shown in Fig. 4, gate delay change is about 4% with 40°C temperature difference in a 130-nm industrial process. The wire resistance increased by about 12% for 40°C around the nominal temperature. Delay change for the wire resulted in about 5% for 40°C using the same process.

3.2   Circuit Functionality and Floorplan Dependency

Figure 5 shows maximum and minimum temperature, and

ΔT_{max}, which we define as maximum minus minimum temperature, as a function of the memory area occupancy α_m. Memory power density is assumed to be 0.25 W/mm^2, I/O circuit consumes 20% of total power, and the rest of the power is consumed by logic circuit. Independent of α_m, type-L floorplan always shows larger ΔT_{max}. This is because of the adiabatic boundary condition at the chip periphery. As α_m increases, ΔT_{max} becomes larger since power concentrates on smaller logic area. We see that high power consumption SoC with small logic area tends to have steeper thermal gradient. When memory occupies more than 60% of the chip area, which is not a rare case in recent SoC designs, ΔT_{max} becomes larger than 40°C in this example.

3.3   Package Dependency

Figure 6 shows ΔT_{max} as functions of chip power and pack-
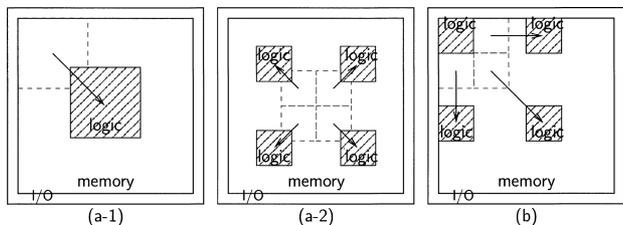
Fig. 7 Logic block movements and partitioning. Dotted blocks represent initial placement and arrows show directions of the block-shift.

age thermal resistance. ΔT_{max} is almost proportional to the chip power because thermal resistance of the package is larger than that of the die. Package cost and its thermal resistance is inversely related in general. Practically, the lowest cost package that can achieve the maximum temperature limit is selected in ordinary SoC design. The solid line shows the optimal package selection to make average chip temperature exactly as $T_{j_{max}}$. This figure implies that the use of better (and possibly more expensive) package with smaller thermal resistance reduces ΔT_{max}. However, the difference is insignificant when compared with the cost increase. Therefore, using excessively low thermal resistance package is not worthwhile in many cases. Even the best package in the figure instead of using appropriate one, the reduction of ΔT_{max} is limited to only less than 10°C.

3.4 Block Partitioning and Placement Dependency

We investigate how the temperature profile is affected by the placement and partitioning. Figure 7 shows logic block movements and partitioning considered in the analysis. Chip area organization and total chip power are both constant.

Pattern (a-1): Locate a logic block at the corner of the chip (same as type-L in Fig. 3) first, then move it diagonally to the center (becomes type-C).

Pattern (a-2): Starting from type-C, divide logic circuit into four smaller blocks and shift each block to four different corners diagonally and synchronously.

Pattern (b): Starting from type-L, divide logic circuit into four smaller blocks and shift three blocks to x, y directions, and diagonally. One block at the corner stays unmoved due to an assumed constraint.

The ΔT_{max} is illustrated in Fig. 8(a). Horizontal axis represents shift distance from original location measured by a cell-grid regardless of the direction. The final placement of (a-1) and the initial placement of (a-2) are identical; the final placement of (a-2) and (b) are also the same. The graph shows that the closer the block is to the corner, the higher the ΔT_{max} becomes. ΔT_{max} is 75°C for type-L and 34°C for type-C. Dividing a block into smaller ones reduces ΔT_{max} further until 11°C and there exists convex optimal position.

3.5 Impact of Temperature Gradient on Performance

Clock skew due to thermal distribution is analyzed using

(a) ΔT_{max}.

(b) Maximum clock skew.
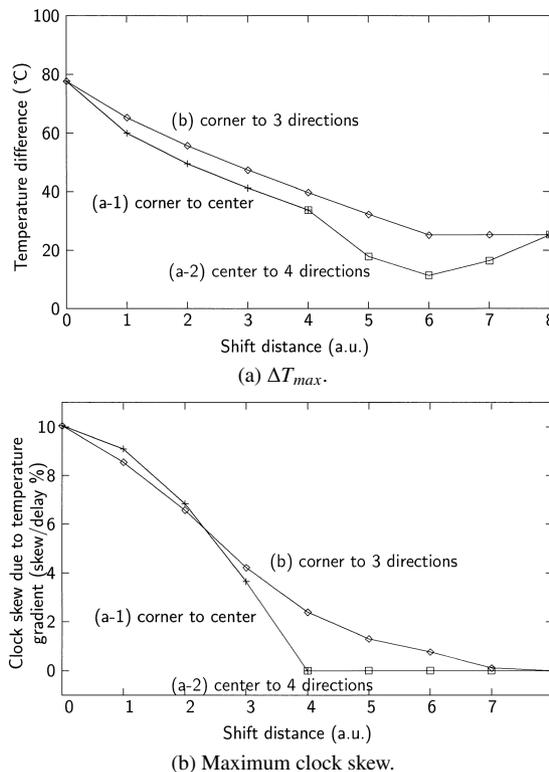
Fig. 8 Floorplan dependency of the maximum temperature difference and clock skew.
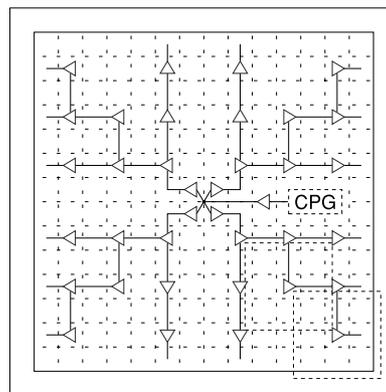
Fig. 9 Clock tree topology for skew comparison. Clock timing is compared at each buffer output.

SPICE simulation. Assumed global clock forms symmetrically balancing tree over the entire chip as depicted in Fig. 9. Naturally, if there exists no thermal distribution, clock skew defined at the buffer output of the same buffer depth is zero. On the other hand, we should see non-zero clock skew when a chip suffers nonuniform thermal distribution as seen in Fig. 8(a). The clock tree has 6 buffer stages at the maximum from tree root to the starting points of local clock distribution. The longest wire length in this example is 1.25-mm and each wire is modeled as 4 stage π-model. Correspondent to the temperature distribution in Fig. 8(a), Fig. 8(b) shows the maximum skew normalized by the nominal clock delay
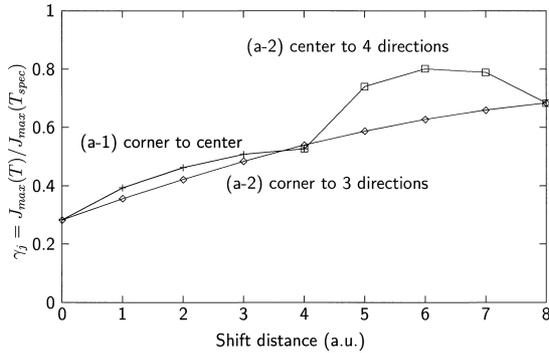
Fig. 10 Degradation ratio of J_{max} relative to specification defined under flat-temperature assumption. The smaller γ_j is, the smaller the actual j_{max} target becomes.

measured at 36 buffer output points. The maximum skew reaches about 10% of clock latency when ΔT_{max} is at its maximum. The clock skew becomes smaller for floorplans with smaller ΔT_{max}.

Similarly, the reliability degradation due to thermal gradient is analyzed. In Fig. 10, we show $\gamma_j = J_{max}(T)/J_{max}(T_{spec})$ using Black's equation [5]: MTTF = $A j^{-n} \exp(E/kT)$ where A and E are process and geometry dependent constants and k is Boltzmann's constant. Here, $J_{max}(T_{spec})$ is the maximum current density defined at the specification temperature ($T_{spec} = 120°C$ in this example), and $J_{max}(T)$ is a 'new' current limit calculated using the actual maximum temperature so that the same MTTF (mean time to failure) is maintained. The higher the chip temperature becomes, the smaller the actual J_{max} target must be, which is represented as $\gamma_j < 1$. When the maximum temperature equals to T_{spec}, γ_j becomes 1. We see that in high-performance SoC, local temperature exceeds specification, which enforces exponentially strict current density limit which is represented as $\gamma_j < 1$ in the graph. Figure 8(c) clearly shows that different floorplan can relax current density limit from 0.3 to 0.8, which means substantial reliability improvement.

The models used in the thermal simulation are simplified so that we can clearly understand the effects of block power and placement on thermal distribution. Considering practical applications, the model has still room for improvements. For example, more detailed modeling of the wire layers is possible if detailed statistical information of the wire distribution is available. The thermal property can be different cell by cell depending on wire congestion and length distribution which may result in more accurate thermal map than using constant values as in Table 1. Using more accurate power information is another possibility. When we move circuit block, block power may become also different. We would like to mention that these modifications are possible and straightforward.

Above analysis shows that effects of on-chip thermal gradient are unavoidable for high-end SoC especially in the designs containing a lot of memory and concentrated logic power. From Fig. 8, we understand that there are two ways to reduce temperature-dependent clock skew. One is to minimize ΔT_{max}, and the other is to equalize temperature for clock distributing branches. The efficacy of matching branch temperatures can be understood by pattern (a-2). Because the clock tree is constructed as both x- and y-axis symmetric, temperature distribution of each branch becomes the same, resulting in zero skew although ΔT_{max} is non-zero. However, this situation is practically hard to realize. On the other hand, reducing ΔT_{max} is block placement dependent and is worth considering. Thermal flattening to minimize ΔT_{max} will be discussed in the next section.

4. Temperature Flattening Procedure in Floorplan Design

Above simulation results suggest that earlier design stages are more suitable for improving thermal gradient because small change made in floorplan makes large difference in temperature distribution. We therefore propose a temperature flattening procedure suitable for floorplan design. In the procedure, we first generate a floorplan considering timing as a dominant cost, then amend it incrementally using thermal analysis described in previous sections. The other possible approach is to optimize everything together by adding temperature as one of the cost function as well as the other costs such as timing, net length, or power (e.g. multiple cost evaluation in simulated annealing [13]). We take incremental optimization approach after initial floorplan based on timing. The reason is as follows. We found that the temperature effect in delay is approximately 10% at the largest for recent SoC. It may be a rare case when clock and data are routed using paths that are significantly different in temperature but converge in the same FF because the thermal gradient on an SoC cannot become sharp for typical combination of the chip power and package thermal resistance in our experiments. Thermal impact on delay becomes progressively important but it is, for a while, a secondary problem. We assume that block-based placement and block shape is rectangular. We have not yet analyzed convex shaped block but the procedure should also be applicable without significant modification.

In the following, we limit grid-cells in discussion are at the substrate surface, layer 'sub_s.' Let T_i be temperature of i-th grid-cell. Here $i \in S_k$ and S_k is a set of cell number of continuous grid-cells that belongs to a logic block k. The procedure to decrease ΔT_{max} is summarized as follows.

1. Generate an initial floorplan based on timing optimization without considering temperature.
2. Calculate temperature distribution. Stop if ΔT_{max} is less than predefined threshold or iteration count exceeds predefined limit. Otherwise, go to step-3.
3. For each block, calculate sum of thermal gradient vector v_d projected on x-y plane. Move a block j that has the largest $\|v_d\|$ to the direction that v_d points to improve logic block placement, then go to step-2. Here, $v_d = \sum_i \text{grad } T_i$ and $\|a\|$ is Euclidean norm of a vector a.

The loop in this procedure does not explicitly explain how to keep the timing constraint which should be derived at the step-1. Again, this is because the temperature effect is a secondary factor. A possible scenario to preserve the timing constraint is, for example, after step-1, extract inter-block timing slack and check after step-3 movement. If this movement violates this constraint, try to insert buffer to reduce interconnect delay. If the timing after buffering still does not satisfy the constraint, return to step-3 and select a block that has next largest $\|v_d\|$ components as a block move candidate.

Results in Sect. 3 suggest that 1) partitioning a logic block into smaller pieces and 2) distributing them apart are effective to flatten chip temperature. Because effective thermal resistivity of a chip is determined mainly by a package used, projection of v_d on x-y plane can usually be well approximated by ignoring a z coordinate of the gradient vector $\partial T_i / \partial z$. In the following experiments, the direction of the movement is 8-ways neighbor and its distance is limited to one cell-grid.

Figure 11(a) shows logic block placement after initial floorplan. Ten blocks each of 2×2 grid size are numbered for reference. Initial isothermal line and thermal gradient vector projection are also overwrapping. Calculated ΔT_{max} for the initial floorplan is 26.1°C. Due to power concentration on the right-half of the chip, we notice thermal gradient as many left-pointing arrows. The gradient can also be observed distinctively in the temperature map in Fig. 12(a). This gradient pushes blocks mainly to the left. After 23 iterations, ΔT_{max} has been reduced to 7.4°C (Fig. 12(b)). Here, we see that relative position of the blocks does not change much during the flattening process, which is a good property for timing convergence. Due to edge effect and sparse discretization in thermal simulation model, the optimization result in Fig. 11(b) does not appear to be completely uniform distribution. However, we also understand from Fig. 12(b) that each block has been moved to a position where there is almost no temperature interaction between blocks of high-power consumption. Final solutions are acceptable for all test cases we have used although proposed procedure is a greedy process. Table 2 shows all block movements for this example to help understand how the proposed procedure pushed each block to flatten temperature. Block number corresponds the ones in Fig. 7, and movement directions are represented as n, s, w, e being upward, downward, left, and right, respectively.

Table 3 summarizes results on other examples. Circuits 1 through 5 cover different block number, size, shape, and initial floorplan but the total power and power ratio between memory and logic are the same for all circuits. The block shape includes large square, thin and tall rectangular, L-shape, and combination of these shapes. Experiments show that within relatively small number of iterations, proposed

(a) Initial.

(b) Final.
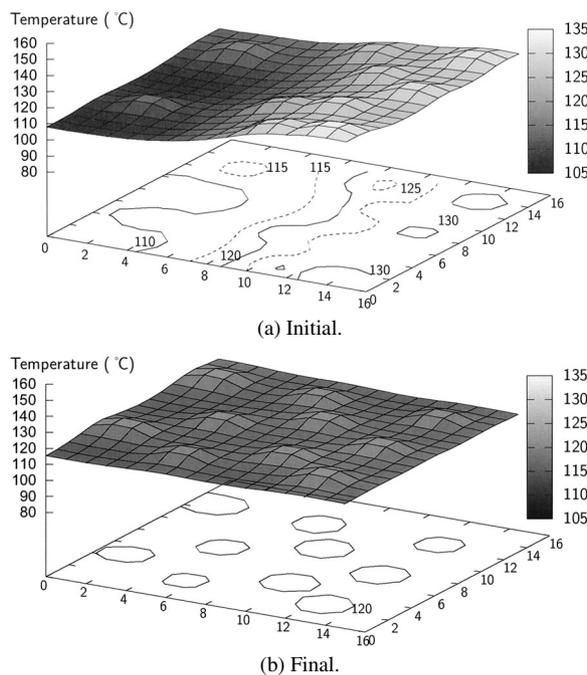
Fig. 12 Die surface temperature distribution of Fig. 11.

Table 2 A track of block movement for Fig. 11.

#	blk#	dir.	#	blk#	dir.	#	blk#	dir.
1	3	w	9	8	sw	17	5	w
2	0	w	10	9	sw	18	3	sw
3	0	w	11	3	nw	19	4	nw
4	8	w	12	3	w	20	5	s
5	3	w	13	0	sw	21	7	sw
6	6	w	14	6	sw	22	7	e
7	0	w	15	2	w	23	7	nw
8	0	w	16	0	nw			

(a) Initial.

(b) Final.

Fig. 11 Logic block placement before and after thermal flattening. Non-hatched areas are memory and I/O circuits. (a) Also shows thermal gradient vector and isothermal map, (b) shows block movements from (a). Only slight changes in floorplan improve temperature difference significantly from $\Delta T_{max} = 26.1°C$ to 7.4°C after 23 iterations.

Table 3 Summary of thermal flattening experiments.

circuit number	# of blocks	initial ΔT_{max}	final ΔT_{max}	# of iteration
1	4	51.0	23.9	7
2	4	49.3	24.6	6
3	4	50.9	24.2	6
4	10	24.8	7.8	20
5	10	18.8	11.7	8

procedure reaches final temperature and decrease temperature difference to about 1/2 to 1/3 compared with its initial value. The procedure not only improves quality of timing design by reduced timing difference due to thermal gradient, but significantly improves reliability because the process reduce maximum temperature on a chip.

5. Conclusion

We have proposed a thermal modeling method that is effective for chip-level analysis and optimization in SoC design. Effects of dominant factors for determining the thermal gradient such as memory and logic area, power consumption, floorplan, etc. are quantitatively analyzed. Further, the global clock skew is simulated for various floorplans using industrial device models, which confirmed the correlation between maximum temperature difference and skew. We also showed that modifying floorplan effectively reduces EM risk. Finally, a practical temperature flattening procedure is presented. It was found that even a small shift of logic blocks is able to improve circuit performance substantially.

References

[1] J. Clabes, J. Friedrich, M. Sweet, J. DiLullo, S. Chu, D. Plass, J. Dawson, P. Muench, L. Powell, M. Floyd, M. Lee, M. Goulet, J. Wagoner, N. Schwartz, S. Runyon, G. Gorman, P. Restle, R. Kalla, J. McGill, and S. Dodson, "Design and implementation of the POWER5^{TM} microprocessor," Proc. ISSCC, pp.56–57, 2004.

[2] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage estimation considering power supply and temperature variations," Proc. ISLPED, pp.78–83, Aug. 2003.

[3] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware computer systems: Opportunities and challenges," IEEE Micro, vol.23, no.6, pp.52–61, Nov.-Dec. 2003.

[4] A.H. Ajami, M. Pedram, and K. Banerjee, "Effects of non-uniform substrate temperature on the clock signal integrity in high performance designs," Proc. CICC, pp.233–236, 2001.

[5] J. Black, "Electromigration—A brief survey and some recent results," IEEE Trans. Electron Devices, vol.ED-16, no.4, pp.338–347, April 1969.

[6] K. Banerjee, M. Pedram, and A.H. Ajami, "Analysis and optimization of thermal issues in high-performance VLSI," Proc. ISPD, pp.230–237, 2001.

[7] Y.K. Cheng and S.M. Kang, "Fast thermal analysis for CMOS VLSIC reliability," Proc. CICC, pp.479–482, 1996.

[8] Z. Yu, D. Yergeau, and R. Dutton, "Full chip thermal simulation," Proc. ISQED, pp.145–149, 2000.

[9] T.Y. Wang and C.C.P. Chen, "3-D thermal-ADI: A linear-time chip level transient thermal simulator," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.21, no.12, pp.1434–1445, Dec. 2002.

[10] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," Proc. ICCAD, pp.86–89, 2003.

[11] B. Obermeier and F.M. Johannes, "Temperature-aware global placement," Proc. ASP-DAC, pp.143–148, 2004.

[12] H. Eisenmann and F. Johannes, "Generic global placement and floorplanning," Proc. DAC, pp.269–274, 1998.

[13] C.H. Tsai and S.M. Kang, "Cell-level placement for improving substrate thermal distribution," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.19, no.2, pp.253–266, Feb. 2000.

[14] T.Y. Chiang, K. Banerjee, and K.C. Saraswat, "Compact modeling and SPICE-based simulation for electrothermal analysis of multilevel ULSI interconnects," Proc. ICCAD, pp.165–172, 2001.

[15] J.A. Davis, V.K. De, and J.D. Meindl, "A stochastic wire-length distribution for gigascale integration (GSI) part I: Derivation and validation," IEEE Trans. Electron Devices, vol.45, no.3, pp.580–589, March 1998.

[16] SIA, International Technology Roadmap for Semiconductors, 2003 Edition.

[17] L.W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Tech. Rep., UCB Memorandum, 1975.

Takashi Sato received the B.E., M.E. degrees from Waseda University, Tokyo, Japan and the Ph.D. degree from Kyoto University, Kyoto, Japan, respectively. From 1991 to 2003, he worked for Hitachi, Ltd., where he was engaged in the design and development of analog circuit simulator, high-speed processor—memory interface circuits. Since 2003, he has been with Renesas Technology Corp., where he is engaged in signal and power supply integrity analysis and related design methodologies. He was a visiting industrial fellow at the University of California, Berkeley, from 1998 to 1999. His research interests include analog circuit simulation techniques, on-chip and on-board interconnect modeling, signal and power supply integrity analysis, and their application to high speed interface circuits. Dr. Sato is a member of the IEEE. He received the Beatrice Winner Award at ISSCC 2000 and the Best Paper Award at ISQED 2003.

Junji Ichimiya was born in Osaka, Japan, in 1973. He received the B.E degree in electrical engineering from Osaka Electro-Communication University, in 1997. He joined imaging system LSI development center, Ricoh Company, Ltd. and then he had been engaged in the designs and development of SRAMs and processors. Currently, he is with sever systems unit, Fujitsu limited.

Nobuto Ono was born in Miyagi, Japan, on September 1, 1958. He received the B.E. degree in control engineering from Tokyo Institute of Technology, Japan in 1981. He joined the EDA department of Seiko Instruments Inc., in 1981. Currently, he is director of Jedat Innovation Inc. in Japan.

Koutaro Hachiya was born in Miyagi, Japan on May 31, 1968. He received the B.E. and M.E. degrees in computer science from Tohoku University in 1990 and 1992, respectively. From 1992 to 2002 he was with NEC Corp., Kanagawa, Japan, where he has been developing in-house circuit simulation tools. From 1998 to 1999 he was a visiting scholar at Kiel University, Germany. Since 2002 he has been with NEC Electronics Corp., Kanagawa, Japan, where he has been developing Chip-Package-Board co-design environment. His research interest include computational electronics, especially circuit simulation and interconnect modeling. He is a member of IPSJ, IEEE, ACM, and SIAM.

Masanori Hashimoto received the B.E., M.E. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively. Since 2001, he was an Instructor in Department of Communications and Computer Engineering, Kyoto University. Since 2004, he has been an Associate Professor in Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University. His research interest includes computer-aided-design for digital integrated circuits, and high-speed circuit design. He is a member of IEEE, ACM and IPSJ.