

Thermal Signature: A Simple Yet Accurate Thermal Index for Floorplan Optimization

Jaeha Kung[‡], Inhak Han[‡], Sachin Sapatnekar[†] and Youngsoo Shin[‡]

[‡]Department of Electrical Engineering, KAIST, Daejeon 305-701, Korea

[†]Department of ECE, University of Minnesota, Minneapolis, MN 55455, USA

ABSTRACT

A floorplanning has a potential to reduce chip temperature due to the conductive nature of heat. If floorplan optimization, which is usually based on simulated annealing, is employed to reduce temperature, its evaluation should be done extremely fast with high accuracy. A new thermal index, named thermal signature, is proposed. It approximates the temperature calculation, which is done by taking the product of Green's function and power density integrated over space. The correlation coefficient between thermal signature and temperature is shown to be quite high, more than 0.7 in many examples. A floorplanner that uses thermal signature is constructed and assessed using real design examples in 32-nm technology. It produces a floorplan whose maximum temperature is 11.4°C smaller than that of standard floorplan, on average, in reasonable amount of runtime.

Categories and Subject Descriptors: B.6.3 [Logic Design]: Design Aids—*Optimization*; B.7.1 [Integrated Circuits]: Types and Design Styles—*VLSI (very large scale integration)*

General Terms: Algorithms, Design

Keywords: Thermal analysis, thermal-aware floorplanning

1. INTRODUCTION

Thermal analysis and thermal-aware design have been a focus of research for many years due to increasing power consumption, which has been driven by the demand on more performance and higher level of integration. It is well known that there are several side effects from high temperature. These include the increase in delay, e.g. 30% increase along aluminum wire for 75°C rise of temperature [1], and reduced mean-time-to-failure of wire due to enhanced electromigration, e.g. 90% reduction for the change from 25°C to 52.5°C [2].

Thermal behavior is governed by the well-known heat conduction equation:

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

where T is temperature which is unknown, g is power density of a heat source, and κ is thermal conductivity; ρ and C_p are material dependent parameters. The meaning of (1) is: the energy stored in a volume V (left-hand side) is equal to the sum of heat entering V

through its boundary surface and the heat generated by itself (right-hand side). The equation is subject to boundary conditions. Several numerical methods have been proposed to solve the equation.

The numerical methods, however, cannot be employed in thermal-aware optimization due to their sheer runtime, e.g. order of hours [3]. Floorplanning is one of those optimizations that can benefit chip temperature. It is reported that, by arbitrarily adjusting the floorplan of Alpha 21364 processor, the maximum chip temperature can be reduced by 37°C [4]. This can be intuitively understood because hotter blocks could better be placed far from another hotter blocks considering the conductive nature of heat.

1.1 Related Work

A few methods have been proposed for fast thermal analysis, and have been integrated in thermal-aware floorplanning. Thermal RC network [5] or a resistive network [6] in case of steady state analysis is a notable example. This is a network built based on the analogy between heat transfer and electrical current: heat flow can be described as a current flowing through a thermal resistance thus yields a temperature difference analogous to voltage. The resistive network, in which resistance is modeled per block basis, is used for floorplanning [7, 8]. Temperature estimation is not accurate in this model, especially when floorplan contains blocks of large size. A chip may be divided into a number of imaginary grid cells and the resistance is modeled per grid basis for better accuracy [6, 9]. But, the resistance matrix becomes too large to solve this time. If a grid is 64×64 , it takes 2.77 seconds in the computer we used for experiment; this is prohibitive because matrix has to be solved for each iteration of floorplan optimization.

Instead of temperature, a simple measure, named heat diffusion measure, has been proposed [4]. The rationale is that a block having a higher power density, which is likely to have higher temperature, has to be adjacent to blocks of lower power density. The measure of block i is defined by $H_i = \sum_j [(d_i - d_j) * \text{shared_length}]$, where d_i is power density and shared_length is the length of edges of i and j that touch each other. The measure is then added to yield a single quantity, i.e. $\sum_i H_i$, which is then tried to maximize during floorplanning process. This measure is clearly very simple to compute, but its correlation with maximum temperature turned out to be very weak in our experiment. The correlation increases somewhat when there is a clear difference of power densities, i.e. many blocks of very high power density, many other blocks of very low power density, but no blocks in-between; even in this situation, the correlation was not strong enough, e.g. -0.4 .

1.2 Contributions

Floorplan optimization is usually based on simulated annealing, and so a large number of floorplans are generated and evaluated. The evaluation, therefore, should be done in a short amount of time. For instance, floorplan sizing, which derives the location of each block and computes the area and total wirelength of a given floor-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2011, June 5-10, 2011, San Diego, California, USA.

Copyright 2011 ACM 978-1-4503-0636-2/11/06 ...\$10.00.

plan, takes tens or hundreds of μs as we will see in Section 2.3. In this regard, any thermal index should be very simple to compute while its correlation with real temperature is still maintained with high accuracy, which is our focus. Our main contributions are summarized as follows.

- Thermal signature, which was inspired by the use of Green's function to solve steady state heat conduction equation (Section 2).
- An implementation of floorplanner that uses thermal signature and its assessment using real design examples in 32-nm technology (Section 3).
- A study of the impact of grid size on accuracy and runtime of thermal signature (Section 2.3).

2. THERMAL SIGNATURE

2.1 Motivation

In general, steady state temperature is of importance because, once a chip reaches that state, the temperature does not respond to the instantaneous change of power consumption [6]. This is due to relatively large time constant of heat conduction (a few ms) compared to clock cycle. In steady state, in which $\partial T/\partial t$ is zero, (1) is reduced to

$$\nabla^2 T(\mathbf{r}) = -g(\mathbf{r})/\kappa(\mathbf{r}), \quad (2)$$

where $\mathbf{r} = (x, y, z)$ or $\mathbf{r} = (x, y)$ depending on the domain of computation; this is the form of well-known Poisson's equation. Assume that there is a function G that satisfies:

$$\nabla^2 G(\mathbf{r}, \mathbf{r}_0) = \delta(\mathbf{r} - \mathbf{r}_0), \quad (3)$$

where δ is the Dirac delta function and \mathbf{r}_0 is a point in R^3 (or R^2). Such a function G is called a Green's function (generally speaking, Green's function is defined for any linear differential operator including ∇^2 and in any Euclidean space R^n).

Let the right-hand side of (2) be denoted by $f(\mathbf{r})$ for simplicity of presentation. Multiplying both sides of (3) by $f(\mathbf{r}_0)$ and integrating with respect to \mathbf{r}_0 yields

$$\int_{-\infty}^{\infty} \nabla^2 G(\mathbf{r}, \mathbf{r}_0) f(\mathbf{r}_0) d\mathbf{r}_0 = \int_{-\infty}^{\infty} \delta(\mathbf{r} - \mathbf{r}_0) f(\mathbf{r}_0) d\mathbf{r}_0. \quad (4)$$

Due to the sifting property of delta function, the right-hand side of (4) is evaluated to $f(\mathbf{r})$, which is equal to $\nabla^2 T(\mathbf{r})$ from (2). Thus, we get

$$\int_{-\infty}^{\infty} \nabla^2 G(\mathbf{r}, \mathbf{r}_0) f(\mathbf{r}_0) d\mathbf{r}_0 = \nabla^2 T(\mathbf{r}). \quad (5)$$

In the left-hand side of (5), ∇^2 can be taken out of the integration because it is applied to \mathbf{r} but not to \mathbf{r}_0 . Hence, we finally get

$$T(\mathbf{r}) = \int_{-\infty}^{\infty} G(\mathbf{r}, \mathbf{r}_0) f(\mathbf{r}_0) d\mathbf{r}_0 = - \int_{-\infty}^{\infty} G(\mathbf{r}, \mathbf{r}_0) \frac{g(\mathbf{r}_0)}{\kappa(\mathbf{r}_0)} d\mathbf{r}_0. \quad (6)$$

The expression (6) implies that $T(\mathbf{r})$ can be obtained (without solving differential equation (2)) if a Green's function G is given; this is a well-known technique to solve Poisson's equation. The product of cosine functions [10] or the division of hyperbolic functions [11] have been used for G .

2.2 Definition of Thermal Signature

In order to pursue a fast computation of (6) for a given floorplan, the floorplan area is divided to form a grid. Figure 1 shows an

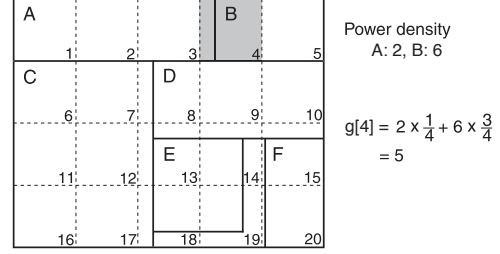


Figure 1: Computation of power density at a grid cell.

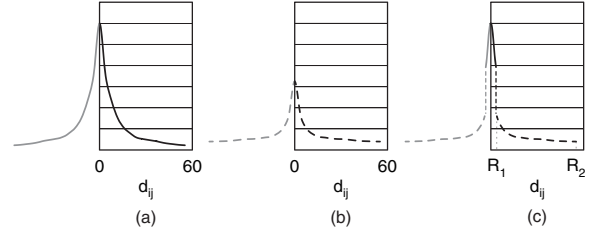


Figure 2: (a) $\nabla^2(1/d_{ij})$, (b) $\nabla^2(C/\sqrt{d_{ij}})$, and (c) $\nabla^2\tilde{G}$.

example, which contains 6 blocks and is divided into a grid of 20 cells. Expression (6) is now approximated by

$$TS[i] = \sum_{j \neq i} \tilde{G}[i, j] g[j], \quad (7)$$

which we call *thermal signature*. The computation of (7) is performed at each grid cell i ; \tilde{G} is an approximated Green's function, $g[j]$ is a power density at j , and κ is dropped from the computation since its change over space is small in practice and only relative magnitude of temperature matters for our purpose.

The computation of power density at a grid cell, say $g[4]$, is performed as illustrated in Figure 1. Assume that $1/4$ of the cell is occupied by block A and the remaining $3/4$ by B; $g[4]$ is then the average power density of the two blocks, i.e. the sum of power densities, each one multiplied by the proportion of area in the cell.

The key in the thermal signature (7) is the selection of \tilde{G} , such that its computation is done very fast while the accuracy is not sacrificed too much. Consider the following choice:

$$\tilde{G}[i, j] = \begin{cases} \frac{1}{d_{ij}} & \text{if } d_{ij} \leq R_1 \\ \frac{C}{\sqrt{d_{ij}}} & \text{if } R_1 < d_{ij} \leq R_2 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where d_{ij} is the distance (center to center) between two grid cells i and j ; C , R_1 , and R_2 are the constants to be determined. The reason why two functions ($1/d_{ij}$ and $C/\sqrt{d_{ij}}$) are used for \tilde{G} is because the Laplacian of either function alone does not approximate the delta function in faithful way (recall the definition of Green's function (3)). This is illustrated in Figure 2. The Laplacian of $1/d_{ij}$, which is equal to $1/d_{ij}^3$, is shown in Figure 2(a); $\nabla^2(1/d_{ij})$ is defined only in a half plane since d_{ij} is positive, the reflected curve in the other plane ($d_{ij} < 0$) is shown to illustrate how $\nabla^2(1/d_{ij})$ approximates the delta function. The Laplacian of $C/\sqrt{d_{ij}}$, which is equal to $\frac{C}{4}(1/\sqrt{d_{ij}})^5$, is shown in Figure 2(b). The points corresponding to $d_{ij} = 0$ in Figure 2(a) and (b) were calculated for the same arbitrary small value. It is clear that Figure 2(a) approximates the delta function very well when d_{ij} is small, while Figure 2(b) is

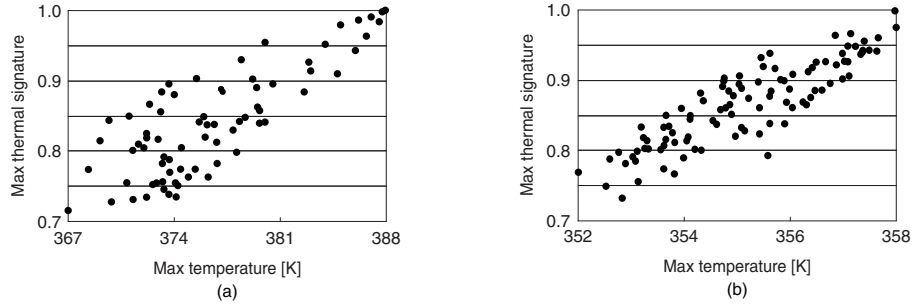


Figure 3: Correlation between maximum temperature and maximum thermal signature, which is normalized, of 100 floorplans: (a) nova and (b) ac97_ctrl.

Table 1: Correlation between maximum temperature and maximum thermal signature

Circuit	# Gates	# Blocks	Average block size (μm^2)	Correlation coefficient
mem_ctrl	4052	15	365	0.846
usb_ctrl	8121	23	561	0.739
ac97_ctrl	6177	29	198	0.839
aemb	17487	29	678	0.727
aquarius	19230	35	576	0.724
nova	29032	53	621	0.765
tv80	7161	80	3373	0.724
wb_dma	3389	80	2178	0.711
aes	7959	120	3011	0.738
pci_bridge	16816	450	1890	0.735

a better choice for larger value of d_{ij} . This leads us to combine the two functions as shown in Figure 2(c).

The constants C , R_1 , and R_2 are determined in empirical fashion. In our experimental setting based on 32-nm industrial technology, $R_1 = 3 \mu\text{m}$ was chosen. Letting $1/d_{ij} = C/\sqrt{d_{ij}}$ when $d_{ij} = R_1$, so that the two functions become continuous, yields $C = 3/5$. R_2 defines the distance beyond which the power consumption of a grid cell on one end of distance affects the temperature of a grid cell on the other end very little; $R_2 = 100 \mu\text{m}$ was used in the experiment.

Tuning: If the objective of floorplanning is to derive a floorplan in which the maximum temperature becomes as low as possible (in addition to the wirelength and area, the usual objective, being reduced as much as possible), we may skip the computation of (7) for the grid cells that are unlikely to have the maximum temperature. For this purpose, we list the blocks (such as A and B in Figure 1) in decreasing order of power density and in decreasing order of power consumption, power density multiplied by area. The blocks that have both power density and power consumption being larger than 60% (in our experiment) of that of the block in the top of corresponding list are selected. The grid cells that are covered or partially covered by the selected blocks are considered for the computation of thermal signature. This greatly helps reduce the computation time, since only $\sim 30\%$ of grid cells are involved as a result. The power density affects temperature while the area is related to heat capacity; the blocks that have higher value in both quantities are likely to yield high temperature, which is the rationale behind the heuristic.

Note that the computation of (8) is independent of floorplan and independent of circuit; it is only determined by distance between

two grid cells (d_{ij}). It is thus calculated a priori for every possible d_{ij} within R_2 and stored as a table, so that it can be looked up during the computation of (7). The advantage of using \tilde{G} in this regard is that precomputation of the table takes less time, tens of μs . This can be compared to the time when G is directly used, which is about tens of seconds [10].

2.3 Experiments

To assess the effectiveness of thermal signature, we took 10 designs from OpenCores [12] including processor cores (aemb, aquarius, and tv80), multimedia cores (nova and aes), and control circuits (mem_ctrl, ac97_ctrl, usb_ctrl, wb_dma, and pci_bridge); they are listed in Table 1. Each design, given as Verilog code, was taken to a commercial logic synthesis tool [13] to generate a netlist using 32-nm technology library; the second column reports the total gate count. The number of blocks that we assumed for floorplanning is listed in the third column. Power consumption of each block was estimated [13] assuming 0.5 as a signal probability at each input, which was then used to get power density.

2.3.1 Accuracy

For each design, we generated 100 floorplans. Thermal analysis [10] was performed for each floorplan to obtain the maximum temperature. The maximum thermal signature was also calculated using (7). The two figures are then compared to assess the accuracy of the latter. Figure 3 shows, for designs nova and ac97_ctrl, the correlation between the two, in which thermal signature is shown as a normalized value. Notice that thermal signature (7) does not report absolute temperature but reports a relative measure. It is clear that the two figures are highly correlated with correlation coefficient of 0.724 and 0.839, respectively. The last column of Table 1 reports the correlation coefficient of all the designs. It deserves notice that there is about 20°C difference of maximum difference (x-axis of Figure 3(a)) in nova while different floorplans are tried, which demonstrates the effectiveness of thermal-aware floorplanning.

Figure 4(a) is a thermal map of ac97_ctrl obtained by thermal analysis. Another thermal map was constructed using thermal signature, which is shown in Figure 4(b). Thermal signature was scaled so that its maximum value matches the maximum temperature of Figure 4(a), again because thermal signature does not report the absolute temperature. All the blocks were involved in this thermal signature computation. The two maps seemingly match very well, which implies that the maximum temperature and maximum thermal signature do not simply correlate by chance but they are likely to be observed in the same location of a floorplan.

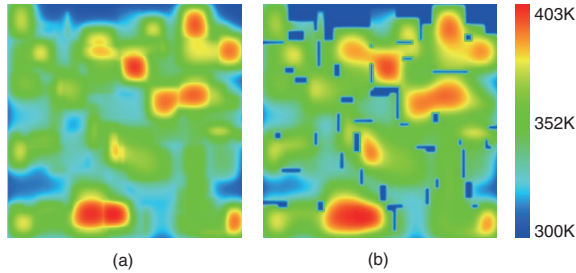


Figure 4: Thermal map of ac97_ctrl: (a) from thermal analysis [10] and (b) from thermal signature. Thermal signature is scaled so that its maximum value matches maximum temperature.

Table 2: Runtime to compute thermal signature; comparison is made to the runtime for floorplan sizing

Circuit	Total # cells	# Cells for TS computation	Runtime (μs)	Runtime for sizing (μs)
mem_ctrl	121	36	13	18
usb_ctrl	289	168	51	310
ac97_ctrl	624	338	320	25
aemb	400	175	59	502
aquarius	400	61	32	424
nova	992	90	98	145
tv80	1296	350	239	350
wb_dma	840	248	123	153
aes	1722	398	464	355
pci_bridge	4032	950	3900	901

2.3.2 Size of Grid Cell

A prominent issue in thermal signature computation is its runtime, because we want it to be done in each iteration of floorplan optimization, typically based on simulated annealing. Intuitively, it can be presumed that the size of grid cell, which determines the number of cells, affects runtime as well as the accuracy of thermal signature. Figure 5(a) shows the accuracy measured as correlation coefficient for three representative designs while the cell width (or height; grid cell is a square) is varied. The accuracy tends to degrade as the cell width is increased, as it should, but the width in which rapid degradation starts to occur is different. This can be understood by looking at the average block size shown in the fourth column of Table 1 (in particular, 198 for ac97_ctrl, 621 for nova, and 2178 for wb_dma). In our experiment, we determined the cell size in proportion to the average block size, which turned out to provide consistent accuracy. The cell size determined as such is marked in Figure 5(a). The correlation coefficient reported in Table 1 is also based on the cell size determined following the heuristic method.

The runtime to compute thermal signature is determined by the number of grid cells that are involved in the computation (see (7) and (8)). The number of cells are dependent on three factors: the cell size, which determines the total number of cells; the blocks that are expected to affect the maximum temperature (see *Tuning* of Section 2.2; recall that the cells that do not overlap with these blocks are dropped from thermal signature computation); and the cells that fall within R_2 (see (8)).

Columns 2 and 3 of Table 2 report the total number of grid cells and the number of cells that are involved in the computation. Figure 5(b) illustrates the change of runtime with varying cell width.

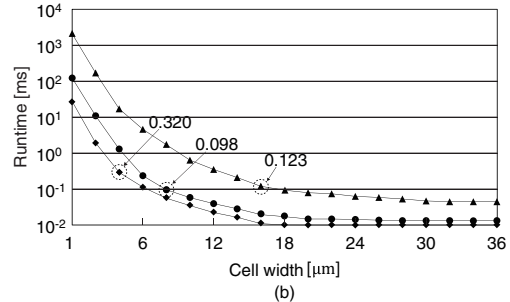
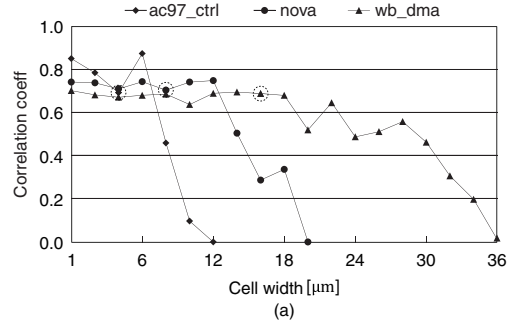


Figure 5: (a) Accuracy measured as correlation coefficient and (b) runtime of computing maximum thermal signature with varying grid cell width (or height).

The runtime is shown in the fourth column of Table 2. The runtime for floorplan sizing (given a floorplan representation, derive the location of each block, and compute area and wirelength) is shown in the last column for comparison. The runtime to compute thermal signature is smaller than that of floorplan sizing in seven examples, but there is large increase of runtime in ac97_ctrl, aes, and pci_bridge due to their large number of cells that are involved in thermal signature computation. For those examples, we can expect that the runtime of thermal aware floorplanning will grow up accordingly, which we discuss in Section 3.

3. THERMAL-AWARE FLOORPLANNING

3.1 Implementation

A prototype of floorplanner that uses thermal signature was implemented based on a conventional floorplanner [14]. A sequence pair [15] is used for floorplan representation. The iteration relies on simulated annealing with piecewise linear cooling schedule, i.e. rapid cooling in high temperature and slow cooling in low temperature. Three moves are randomly selected (with a probability of 0.5, 0.3, and 0.2, respectively) to generate a new sequence pair. The first move randomly picks two blocks and swaps them in a sequence pair (in one sequence or in both). In the second move, a randomly picked block is moved to a new position where its wirelength is minimized (to be specific, a sequence pair is altered such that it becomes next to the block already in that position). The randomly picked block is rotated 90° in the last move.

The difference of objective function (energy in simulated annealing) between two consecutive floorplans, which incorporates thermal signature as well as area and wirelength, is defined by

$$\Delta E = \Delta A + \alpha \Delta W + \beta \Delta(\max TS), \quad (9)$$

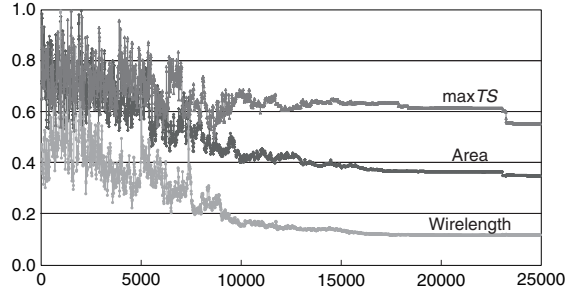


Figure 6: Change of area (A), wirelength (W), and maximum thermal signature ($\max TS$) of nova (all normalized) as annealing progresses.

where α and β are weighting factors; A is the area of bounding box which is normalized to the total area of blocks; W and $\max TS$ are total wirelength and maximum thermal signature, each one normalized to the corresponding quantity of a floorplan that was last accepted. The floorplan is accepted if $\Delta E < 0$; otherwise it is accepted with probability, as in standard simulated annealing process.

Figure 6 illustrates how the three parameters of objective function change as simulated annealing progresses; the x-axis is an index of floorplan that is accepted. The extent of fluctuation fades away as annealing continues, and each parameter eventually converges to a stable value.

3.2 Assessment

We compared the proposed floorplanner ($\alpha = \beta = 1$) with the one that considers only area and wirelength (standard floorplanner). The result is shown in Table 3. The area and wirelength necessarily increase by the average of 11% and 7% respectively, since they are sacrificed for reduced maximum thermal signature. For each floorplan of design that two floorplanners produce, we perform thermal analysis [10]. Thus, the two columns under heading ‘Max T ’ is temperature rather than thermal signature. The maximum temperature is reduced by 11.4°C on average (as high as 27.7°C in tv80).

Runtime of the proposed floorplanner shown in the last column of Table 3 is comparable¹ to that of standard floorplanner in seven designs, but as we discussed along with Table 2, runtime increases substantially in the remainder of designs, which are marked in bold-face. The large runtime can be predicted a priori since it is caused by the number of grid cells involved for the thermal signature computation (see column 3 of Table 2) and the number of cells within the range of interest R_2 (see (8)). Therefore, for those designs which are likely to have runtime issue, we may adjust simulated annealing process for speedup at the cost of degradation in floorplan quality.

Two options were tried and the result is shown in Figure 7. The black bars correspond to the original setting of proposed floorplanner, i.e. the last four columns of Table 3; the number of iterations at each annealing temperature was reduced by 4× in the white bars; the width of grid cell was increased by 2× in the gray bars. Both options provide appreciable amount of speedup as shown in Figure 7(d); runtime is now kept about 2× or below of standard floorplanner. The area and wirelength become worse in the first option, where there is less number of iterations per annealing temperature; this is because of reduced number of floorplans that are tried for

¹We may regard 2× increase of runtime, for example, to be acceptable considering that thermal signature, which has high correlation with real temperature, is extracted.

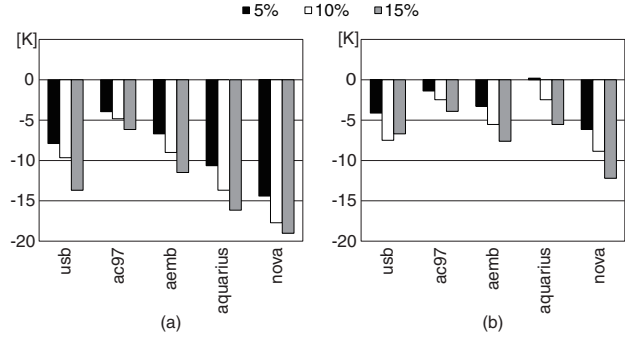


Figure 8: Area apportionment to further reduce maximum temperature: (a) extra area apportioned to blocks of high power density and (b) whitespace allocation.

optimization. Maximum temperature is further reduced in aes and pci_bridge, on the other hand; this is believed to be affected by more whitespace introduced as a result of area increase. The number of floorplans that are tried remains the same in the second option, but thermal signature is used, which has less correlation with actual temperature now. In other words, there is a possibility that a floorplan may be regarded to have low maximum temperature while it is opposite in reality, which explains the result in Figure 7(c) and in turn (a) and (b).

3.3 Application: Area Apportionment

If the maximum temperature should be further reduced after the proposed floorplanning is performed, one simple technique is to allow area increase if it can be accommodated. There are two directions for this purpose: deliberately increase the area of blocks having higher power density so that their power density can go down, or allocate whitespace. Figure 8 shows the result of experiment, one for each method, in which the temperature corresponding to column 8 of Table 3 is used as a reference. In the first method, the blocks whose power density exceeds 90% of the maximum power density were selected; the area of each selected block was increased in proportion to its original area within the budget. In the second method, 10 dummy blocks, which have no connection and have 0 power density, were created such that their total area matches the budget. Each design was then submitted to our proposed floorplanner, which was then followed by thermal analysis to get the temperature.

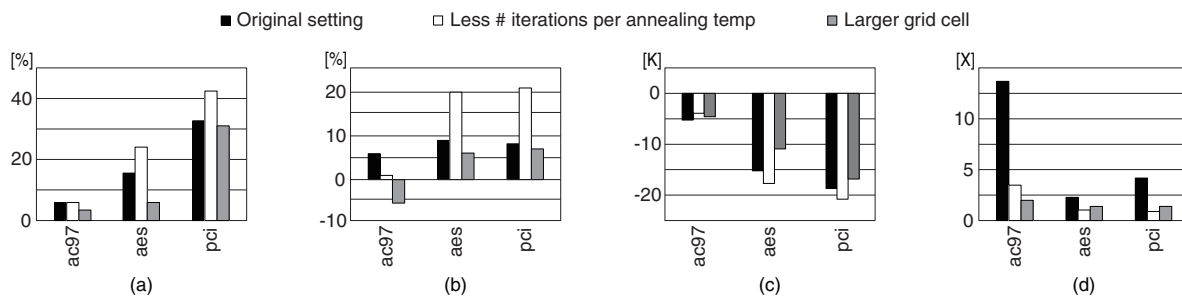
It is observed that suppressing higher power density (Figure 8(a)) is better than allowing whitespace around the blocks of higher temperature (Figure 8(b)), at least in our simple experimental setting. This is in particular true when the area budget is small, e.g. 5%. Designs such as aquarius and nova benefit a lot from the first method; there is rather clear difference of power densities in these designs, i.e. some blocks of very high power density and many others of low power density.

4. CONCLUSION

A new thermal index, named thermal signature, has been proposed. It was inspired by the fact that temperature can be obtained by the product of Green’s function and power density, which is integrated over space. If Green’s function is simplified by some other function and a floorplan is divided into a grid, the overall calculation can be approximated by a series of multiplication and addition. The developed thermal signature has been shown to exhibit

Table 3: Comparison of floorplanning without and with thermal signature

Circuit	Standard floorplanning				Floorplanning with thermal signature			
	Area (μm^2)	Wirelength (mm)	Max T (K)	Runtime (s)	Area (\times)	Wirelength (\times)	Max T (Δ)	Runtime (s)
mem_ctrl	6708	18	371.1	0.5	1.08	1.11	-8.2	0.8
usb_ctrl	15744	655	369.9	12.8	1.04	1.04	-9.3	15.0
ac97_ctrl	8855	18	355.2	1.3	1.06	1.06	-5.3	17.7
aemb	21896	1330	381.3	26.4	1.06	1.05	-7.9	28.8
aquarius	22294	988	382.9	26.5	1.08	1.01	-3.8	28.2
nova	63993	226	360.9	14.0	1.06	1.17	-8.4	22.0
tv80	292100	2543	393.8	46.6	1.15	1.10	-27.7	80.2
wb_dma	198679	917	374.7	21.5	1.08	1.05	-9.4	41.0
aes	414510	3261	364.1	75.4	1.16	1.09	-15.2	174.2
pci_bridge	995946	10308	373.0	730.4	1.33	1.08	-18.5	3031.7
Average					1.11	1.07	-11.4	

**Figure 7: Assessment of tuning of simulated annealing; (a) Δ Area, (b) Δ Wirelength, (c) Δ (Max temperature), and (d) runtime ratio with standard floorplanner as a reference. Some design names are written in shortened.**

very high correlation with actual temperature with small amount of computation time. A prototype of thermal-aware floorplanner has been designed and tested using real examples in 32-nm technology.

A floorplan move that explicitly targets temperature, such as moving a block of higher temperature to a region of lower temperature, could benefit thermal-aware floorplanning. We investigated area apportionment in a simple experimental setting; a systematic approach toward the problem deserves future investigation.

References

- [1] D. Chen, E. Li, E. Rosenbaum, and S.-M. Kang, "Interconnect thermal modeling for accurate simulation of circuit timing and reliability," *IEEE Trans. on Computer-Aided Design*, vol. 19, no. 2, pp. 197–205, Feb. 2000.
- [2] S. Rzepka, K. Banerjee, E. Meusel, and C. Hu, "Characterization of self-heating in advanced VLSI interconnect lines based on thermal finite element simulation," *IEEE Trans. on Components, Packaging, and Manufacturing Technology*, vol. 21, no. 3, pp. 406–411, Sept. 1998.
- [3] T.-Y. Wang, Y.-M. Lee, and C. Chen, "3D thermal-ADI: an efficient chip-level transient thermal simulator," in *Proc. Int. Symp. on Physical Design*, Apr. 2003, pp. 10–17.
- [4] Y. Han and I. Koren, "Simulated annealing based temperature aware floorplanning," *Journal of Low Power Electronics*, vol. 3, no. 2, pp. 141–155, Aug. 2007.
- [5] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: a compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on VLSI Systems*, vol. 14, no. 5, pp. 501–513, May 2006.
- [6] C.-H. Tsai and S.-M. Kang, "Cell-level placement for improving substrate thermal distribution," *IEEE Trans. on Computer-Aided Design*, vol. 19, no. 2, pp. 253–266, Feb. 2000.
- [7] W. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. Irwin, "Thermal-aware floorplanning using genetic algorithms," in *Proc. Int. Symp. on Quality Electronic Design*, Mar. 2005, pp. 634–639.
- [8] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *The Journal of Instruction-Level Parallelism*, vol. 7, Oct. 2005, <http://www.jilp.org/vol7/>.
- [9] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2004, pp. 306–313.
- [10] Y. Zhan and S. Sapatnekar, "A high efficiency full-chip thermal simulation algorithm," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2005, pp. 635–638.
- [11] B. Wang and P. Mazumder, "Accelerated chip-level thermal analysis using multilayer Green's function," *IEEE Trans. on Computer-Aided Design*, vol. 26, no. 2, pp. 325–344, Feb. 2007.
- [12] "Opencores," <http://www.opencores.org/>.
- [13] Synopsys, *Design Compiler User Guide*, Sept. 2008.
- [14] S. Adya and I. Markov, "Fixed-outline floorplanning: enabling hierarchical design," *IEEE Trans. on VLSI Systems*, vol. 11, no. 6, pp. 1120–1135, Dec. 2003.
- [15] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing-based module placement," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 1995, pp. 472–479.