Temperature Aware Microprocessor Floorplanning Considering Application Dependent Power Load [*]

Chun-Ta Chu, Xinyi Zhang, Lei He, and Tom Tong Jing
Department of Electrical Engineering, University of California at Los Angeles, CA, 90095
{chunta, zxy, lhe, tomjing}@ee.ucla.edu

ABSTRACT

This paper studies microprocessor floorplanning considering thermal and throughput optimization. We first develop a stochastic heat diffusion model taking into account the application dependent power load for thermal analysis. Then, we design the floorplanning algorithm based on this model. Experimental results show that, compared with the deterministic heat diffusion model, our model obtains up to 3.2^oC reduction of the on-chip peak temperature, 1.25% reduction of the area, and 1.125x better CPI (cycles per instruction) performance, respectively. Compared with temperature aware floorplanning in the HOTSPOT tool set that ignores interconnect pipelining, our algorithm is up to 27x faster, reduces the peak temperature by up to 3^oC, and also reduces CPI significantly with a negligible area overhead.

1. INTRODUCTION

Traditional microprocessor floorplanning only considers area and wire length. Several recent studies [1] [2] [3] have optimized microprocessor floorplanning on area and CPI considering interconnect pipelining. As devices keep shrinking, the decreasing rate of power consumption in a chip cannot catch up with the shrinking chip size. Therefore, the impact of thermal effects should be considered in the floorplanning phase, which has not been done in the above studies.

The existing thermal modeling and thermal aware optimization include: [4] used HOTSPOT [5] to model the whole package as a thermal RC network and calculated the peak steady-state temperature. [6] further considered transient power and the dependency between power and CPI. One major drawback of the aforementioned work is that the temperature is calculated to evaluate each new floorplan, which is time-consuming. [7] proposed a simple deterministic heat diffusion model to avoid directly calculating temperature. However, this model is too simplified to guarantee a good solution.

In this paper, we develop an accurate, yet efficient thermal-aware floorplanning considering the correlation between power consumptions for different micro-architecture modules and for different microprocessor applications. Instead of calculating temperature directly during floorplanning, we develop a stochastic heat diffusion model with consideration of block geometry and the above power correlation. We apply this model to the iterative-improvement-based floor-

planning for thermal optimization, and also simultaneously optimize throughput using the trajectory piecewise linear model (TPWL) developed in [2].

The rest of this paper is organized as follows. Section 2 introduces the background of floorplanning, microprocessor performance and the relation between power and temperature. Section 3 reviews the deterministic heat diffusion model, points out its shortcomings and then presents our stochastic heat diffusion model. Section 4 summarizes the experimental results. We conclude in Section 5 and the details of this paper is included in a technical report at http://eda.ee.ucla.edu [8].

2. PROBLEM FORMULATION

2.1 Floorplanning

The objective function in our floorplan algorithm consists of area, CPI, and thermal effect as follows.

$$W_{area} \cdot \frac{Area}{Area_{norm}} + W_{CPI} \cdot \frac{CPI}{CPI_{norm}} +$$
$$W_{thermal} \cdot \frac{thermal}{thermal_{norm}} \quad (1)$$

where W_{area}, W_{CPI}, and $W_{thermal}$ are the weights for corresponding constraints. $Area_{norm}$, CPI_{norm}, and $thermal_{norm}$ are terms for normalization. Most current floorplanning solvers are based on simulated annealing (SA) algorithm [9] [10], which is also used in this paper.

2.2 Microprocessor Performance

Due to the increasing clock rate, interconnect delay may become longer than one clock cycle. In this case, interconnect pipelining is a must and it affects CPI. However, CPI with interconnect pipelining obtained by micro-architecture level simulation is time-consuming. Here, we apply efficient TPWL model from [2].

2.3 Power and Temperature

In this work, we assume the power is invariant over different floorplan for modules and different temperature, but our work can be easily extended to consider leakage, temperature, floorplanning interdependency [11].

3. STOCHASTIC THERMAL-AWARE FLOORPLANNING

3.1 Deterministic Heat Diffusion Model

3.1.1 Deterministic model

According to [7], the heat diffusion between two adjacent modules M_i and M_j can be represented as

$$h(M_i, M_j) = (\overline{P_{Di}} - \overline{P_{Dj}}) \cdot shared\_length_{ij} \qquad (2)$$

where $\overline{P_{Di}}$ and $\overline{P_{Dj}}$ are the average power density over time, $shared\_length_{ij}$ is the shared length between M_i and M_j.

The total heat diffusion for module i is

$$H_i = \sum_{j\ adjacent\ to\ i} h(M_i, M_j) \qquad (3)$$

Although the heat diffusion model is a good representation to estimate the lateral heat flow, this model is over simplified since it ignores other factors described below.

3.1.2 Shortcomings

First of all, given a micro-architecture and a series of applications, power vectors over time for two modules may be either positively or negatively correlated. Using average power may underestimate the peak temperature for positively correlated modules as shown in Fig. 1.
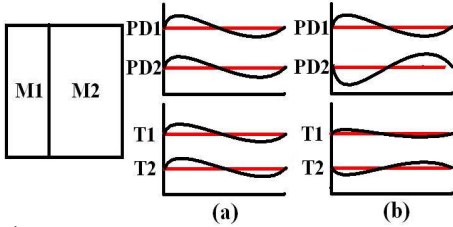

Figure 1: Temporal correlation between $M1$ and $M2$, (a) positively correlated, and (b) negatively correlated. $M1$ has a higher transient temperature in (a) than in (b), although the average power is same.

Second, the module next to the border of a die has extra heat flow to the heat spreader or the ambient.

Third, the heat diffusion from some modules to the dead space (shadow in Fig. 2) is much larger than that from one module to another module as shown in Fig. 2.
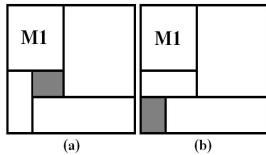

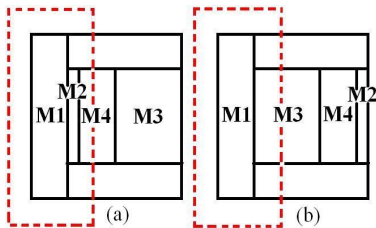**Figure 2:** Dead space effect: $M1$ has a lower temperature in (a) than in (b)

Figure 3: $M1$ has a lower temperature in (b) since $M3$ and $M2$ have same power density but $M3$ is larger than $M2$

Fourth, given four modules $M1$, $M2$, $M3$, and $M4$ with power density $P_{D1} > P_{D4} > P_{D2} = P_{D3}$ in Fig. 3, $M1$ may have a lower temperature with adjacency of $M3$ since $M1$ can diffuse more heat to $M3$ than $M2$, which suggests the heat diffusion should also consider the depth of the adjacent module, as well. Considering this, we can predefine a *penetration window* to enclose the target module. For example, in Fig. 3 (a), $M2$ is inside the red window (dash line), and we have to consider the area of $M4$ inside the red window as well. Similarly, since $M3$ crosses the window, we only consider the area of $M3$ inside the window. Details will be described in the next section.

Finally, considering the hottest modules, just summing their heat diffusion may not guarantee a good solution. We should use the weighted sum of the heat flow for those hottest modules to reduce peak temperature more effectively.

3.2 Stochastic Modeling

3.2.1 Stochastic heat diffusion model

Based on the observations in Sub-section *3.1.2*, we propose an accurate and efficient stochastic heat diffusion model below. Given a micro-architecture floorplan with m modules, n dead spaces, and power vector $P_i = [p_{i1}, ..., p_{iT}]$ over T time steps for module M_i, $1 \le i \le m$.

The mean power density $\overline{P_{Di}}$ for module M_i is

$$\overline{P_{Di}} = E(P_{Di}) = \frac{1}{A_i} \cdot \frac{1}{T} \cdot \sum_{j=1}^{T} p_{ij} \qquad (4)$$

where A_i is the area for module M_i, P_{Di} is the transient power density vector, which equals $\frac{P_i}{A_i}$. In this paper, $E(X)$ is the expectation value of vector X.

The power density covariance between any two modules M_i and M_j is

$$cov(P_{Di}, P_{Dj}) = E(P_{Di} \cdot P_{Dj}) - \overline{P_{Di}} \cdot \overline{P_{Dj}} \qquad (5)$$

Given x adjacent modules and y adjacent dead spaces, and a penetration window size $W \times L$, the heat diffusion vector to the adjacent modules H_{i_adj} and to the adjacent dead spaces H_{i_dead} for module M_i are defined as follows, respectively.

$$H_{i\_adj} = \sum_{j\ =\ 1}^{x} (P_{Di} - P_{Dj}) \cdot L_{ij} \qquad (6)$$

$$H_{i\_dead} = \sum_{j\ =\ 1}^{y} P_{Di} \cdot C_{ij} \qquad (7)$$

where L_{ij} is the shared length between M_i and M_j, C_{ij} is the shared length between M_i and dead space N_j

The heat diffusion vector to the border is

$$f(B_i) = P_{Di} \cdot B_i \cdot \frac{Con\_lateral}{Con\_adjacent} \qquad (8)$$

where B_i is the shared length between M_i and the border of the die, $Con\_lateral$ and $Con\_adjacent$ are the unit lateral conductance between the heat spreader and M_i and between two adjacent modules, respectively, both of which can be calculated according to [5].

The standard deviation of the total heat diffusion for module M_i is

$$\sigma_i = sqrt(E((H_{i\_adj} + H_{i\_dead} + f(B_i))^2) - (E(H_{i\_adj} + H_{i\_dead} + f(B_i)))^2) \qquad (9)$$

The stochastic heat diffusion model for M_i is

$$\tilde{H}_i = E(H_{i\_adj}) + E(H_{i\_dead}) + E(f(B_i)) + 3 \cdot \sigma_i \quad (10)$$

where the first two terms are the mean heat diffusion to the adjacent modules and dead space, respectively, the third term is the mean heat diffusion to the lateral heat spreader, and $3\sigma_i$ is the term for the correlation impact approximated by Equation (9). The larger the standard deviation between modules is, the smaller the correlation is.

If N_j or M_j are totally inside the penetration window, we have to consider other modules which are partially inside the window. Then P_{Dj} is modified to

$$P_{Dj} = \frac{\sum_{k\,=\,1}^{K} \tilde{P_{Dk}} \cdot D_k \cdot (K - k + 1)}{\sum_{k\,=\,1}^{k} D_k \cdot (K - k + 1)} \quad (11)$$

where K is level number between the target block and the window, the level contacting M_i is level 1 and the level contacting the window is level K. $\tilde{P_{Dk}}$ is the average power density in level k. D_k is the depth of each level k. In Fig. 4, the red window (dash line) defines the modules involved, and the blue (slash) one defines the modules to calculate modified P_{D1}. The modified P_{D1} is derived from M_1, M_2, and M_3, and the first belongs to level 1 and the second and the third belong to level 2. Also, the power density of level 1 ($\tilde{P_{D1}}$) is just P_{D1} and the power density of level 2 ($\tilde{P_{D2}}$) is composed of P_{D2} and P_{D3}.
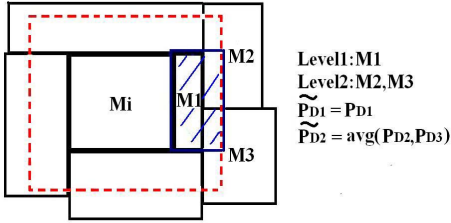

Figure 4: **Illustration of calculation of modified** P_{Dj}

Considering Z potential hottest modules, the total stochastic heat flow then becomes

$$Stochastic\_HeatDiff = \sum_{i\,=\,1}^{Z} W_i \cdot \tilde{H}_i \quad (12)$$

where W_i is the weight proportional to $\overline{P_{Di}}$

3.2.2   Hierarchal clustering

We use K-mean clustering algorithm to find the right number of potential hottest modules. In this paper, the objective is to minimize variance Var

$$Var = \sum_{i\,=\,1}^{k} \sum_{P_{Dj} \in S_i} (|P_{Dj} - \mu_i|)^2 \quad (13)$$

where P_{Dj} is power density of module M_j and μ_i is the average power density within cluster S_i. We use a hierarchical K-mean clustering to find the potential hottest modules. First we set a threshold such as 30% of total modules as the maximum number we have to consider. Then we run K-mean to cluster modules into two clusters and perform the same procedure to the cluster with the higher power density. The above recursive procedure stops when the number of modules in the recursively refined cluster with the higher power density is less than the threshold. Using this

hierarchical method, we can find the optimal number to be considered in the calculation of total heat diffusion.

3.2.3   New objective function

For the new objective function, we replace the term $thermal$ in Equation (1) by $Stochastic\_HeatDiff$, which is calculated from Sub-section *3.2.1* and Sub-section *3.2.2*.

4.   NUMERICAL EXPERIMENTS

4.1   Experiment Setting

Similar to [2], we assume two SuperScalar processors for both $90nm$ and $65nm$ technologies. The settings are shown in Table 1. We treat the blocks as soft and the aspect ratio is between 0.33 and 3 and $L2$ is partition into three modules.

Table 1: **Settings in** $90nm$ **and** $65nm$ **technology.**

	$90nm$	$65nm$
Issue Width	4	8
Die Area(mm^2)	100	200
Die Thickness(mm)	0.5	
Heat Spreader(mm^2)	900	1600
Heat Sink(mm^2)	2500	3600

We use PTscalar [11] to simulate the power consumption for four integer applications $bzip2$, gcc, $gzip$, and mcf and three floating applications art, $equake$, and $mesa$ in SPEC2000 [12]. With these power vectors, we calculate the mean power density (w/mm^2) and standard deviation for each module.
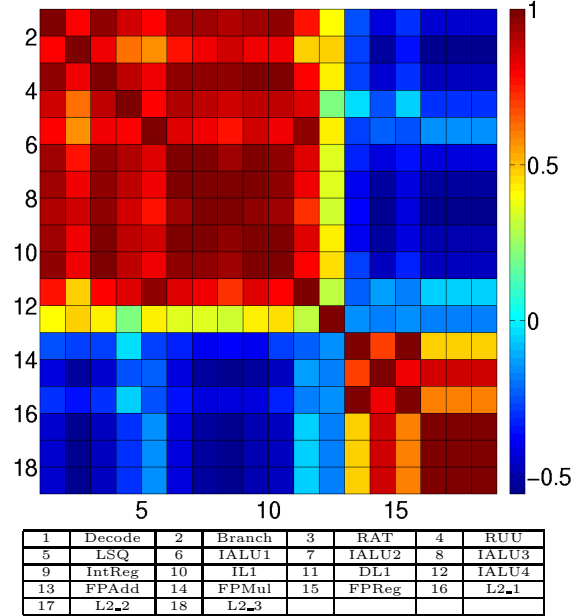

1	Decode	2	Branch	3	RAT	4	RUU
5	LSQ	6	IALU1	7	IALU2	8	IALU3
9	IntReg	10	IL1	11	DL1	12	IALU4
13	FPAdd	14	FPMul	15	FPReg	16	L2_1
17	L2_2	18	L2_3				

Figure 5: **Temporal correlation matrix of power consumption**

Fig. 5 shows the correlation matrix for $90nm$ processor. We can roughly partition all modules into three groups, the first group is from Decode(1) to DL1(11), the second is IALU4(12), which does not have strong correlation to any module, and the last one is from FPAdd(13) to L2_right(18). Modules in the same group are highly positive correlated and

the correlations between modules in the different groups are either uncorrelated or negative correlated.

We use SA-based PARQUET [10] as our base floorplan solver combined with the CPI model [2] and our stochastic heat diffusion model and run the experiments on a Linux workstation. After completing the whole flow with different objectives, HOTSPOT [5] is used to calculate the temperature for verification purposes only. For each objective, we run ten iterations to acquire the best case and the average case.

4.2 Comparison between Thermal Models

We compare our stochastic heat diffusion model (SHDM) with [4], which calculates the maximal temperature for every iteration in SA to estimate the cost for each new floorplan. The objective function is area and thermal effect with weight 0.6 and 0.3, respectively. Table 2 summarizes the final result. From the table, SHDM can reduce $Tmax$ by up to 3^oC (3.2%) with a 1.34% increase in the area. The above results show our model is quite accurate while it is 27x faster for $90nm$ processor and 19x faster for $65nm$ processor.

Table 2: Comparison between our model and [4]

	90nm			65nm		
	Tmax (^oC)	Area (WS) (mm^2)(%)	Time (s)	Tmax (^oC)	Area (WS) (mm^2)(%)	Time (s)
[4]	93	119 (4.7%)	2300	93.3	217 (4.3%)	2980
SHDM	90	121 (5.6%)	85	93.1	220 (5.8%)	155
impact	-3.2%	+1.34%	1/27x	-0.2%	+1.03%	1/19x

Table 3: Comparison of stochastic and deterministic heat diffusion model between different objectives

90nm						
Obj.	CPI		Tmax^oC		Area(mm^2)WS(%)	
	Best	Avg	Best	Avg	Best	Avg
AC	0.82	0.89	97.7	96.7	118.5(3.05)	122.4(6.89)
ACHd	0.99 +21.3%	1.00 +12.4%	92.0 -5.8%	92.2 -4.7%	122.0(6.67) +2.9%	125.3(9.08) +2.3%
ACHs	0.88 +7.3%	0.95 +7.2%	88.8 -9.1%	88.9 -8.1%	121.1(6.10) +2.2%	123.2(7.36) +0.0%

65nm						
Obj.	CPI		Tmax^oC		Area(mm^2)WS(%)	
	Best	Avg	Best	Avg	Best	Avg
AC	0.73	0.77	102.8	105.6	217.8(4.37)	223.6(7.00)
ACHd	0.79 +8.3%	0.84 +8.9%	97.6 -5.0%	100.7 -4.6%	224.1(7.39) +2.9%	221.5(6.42) -1.0%
ACHs	0.78 +6.6%	0.78 +1.8%	97.2 -5.4%	97.6 -7.5 %	221.2(6.03) +1.6%	223.0(6.98) -0.0%

	90nm			65nm		
Obj.	Time(s)	AR		Time(s)	AR	
		Best	Avg		Best	Avg
AC	212	1.10	1.08	483	1.01	1.08
ACHd	248	1.02	1.09	583	1.02	1.05
ACHs	298	1.00	1.06	634	1.04	1.02

4.3 Comparison between Different Objectives

In this section, we compare the thermal impact on the floorplan with different objectives and we also compare the results with [7]. We summarize the results in Table 3. Since our model is stochastic, we denote our model with objectives area, CPI, and heat diffusion as $ACHs$ and [7] with the same objectives as $ACHd$. The weight for area, CPI, and heat diffusion is 0.6, 0.3, and 0.2, respectively, in the objective function. We first compare the results with or without considering thermal effect based on our stochastic model. As shown in Table 3, considering the thermal effect with objective AC in the best case, the maximal temperature is reduced by 9.1% from 97.7^oC to 88.8^oC for $90nm$ and by 5.4% from 102.75^oC to 97.2^oC for $65nm$ with negligible area overhead and increase of CPI up to 7.31%. Clearly,

there is a trade-off between lowering the temperature and reducing CPI.

The work in [7], produced up to 2.9% area overhead and up to 21.3% increase of CPI with similar or less temperature reduction compared with our model, which shows that ours is more accurate and robust. Although our runtime is longer than that in [7], since the run time is just less than a few minutes, it does not have much practical impact.

5. CONCLUSIONS

We have proposed a stochastic thermal-aware floorplanning with consideration of micro-architecture level throughput optimization. First, we have convincingly shown that there are correlations between power for modules over different microprocessor applications. Second, considering dead space, border effect, and the geometry of modules, we have developed a stochastic heat diffusion model and implemented this model on microprocessor floorplanning. Compared with the existing floorplanning using deterministic heat diffusion model, our model obtains up to 3.2^oC reduction of the on-chip peak temperature, 1.25% reduction of the area, and 1.125x better CPI performance, respectively. Moreover, compared with temperature aware floorplanning in the HOTSPOT toolset that ignores interconnect pipelining, our algorithm is up to 27x faster and reduces the peak temperature by up to 3^oC with a negligible area overhead.

6. REFERENCES

[1] M. Ekpanyapong, J. R. Minz, T. Watewai, H. S. Lee, and S. K. Lim, "Profile-guided microarchitecture floorplanning for deep submicro processor design," in *IEEE/ACM Design Automation Conf.*, 2004.

[2] C. Long, L. Simonson, W. Liao, and L. He, "Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects," in *IEEE/ACM Design Automation Conf.*, 2004.

[3] A. Jagannathan, H. H. Yang, K. Konigsfeld, D. Milliron, M. Mohan, M. Romesis1, G. Reinman, and J. Cong1, "Microarchitecture evaluation with floorplanning and interconnect pipelining," in *Asia South Pacific Design Automation Conf.*, pp.1-8 2004.

[4] K. Sankaranarayanan, S. Velusamy, M. R. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," in *Journal of Instruction-Level Parallelism*, 2005.

[5] K. Skadrona, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2003.

[6] V. Nookala, D. J. Lilja, and S. S. Sapatnekar, "Temperature-aware floorplanning of microarchitecture blocks with ipc-power dependence modeling and transient analysis," in *Int. Symp. on Low Power Electronics and Design*, 2006.

[7] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in *Temperature aware Computing Systems*, 2005.

[8] Technical Report EE of UCLA, 2007, http://eda.ee.ucla.edu.

[9] N. Sherwani, "Algorithms for vlsi design automation," Kluwer, 3rd ed., 1999.

[10] S. N. Adya and I. L. Markov, "Fixed-outlined floorplanning through better local research," in *IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 328-334, 2001.

[11] W. Liao, L. He, and K. Lepak, "Temperature and supply voltage aware performance and power modeling at microarchitecture level," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1042 – 1053, 2005.

[12] J. L. Henning, "SPEC CPU 2000:Measuring CPU performance in the new millennium," *IEEE Trans. on Computers*, pp. 28–35 vol.33, 2000.