Temperature-Aware Floorplanning of Microarchitecture Blocks with IPC-Power Dependence Modeling and Transient Analysis *

Vidyasagar Nookala David J. Lilja Sachin S. Sapatnekar
ECE Dept, University of Minnesota, Minneapolis, MN
{vidya,lilja,sachin}@ece.umn.edu

ABSTRACT

Operating temperatures have become an important concern in high performance microprocessors. Floorplanning or block-level placement offers excellent potential for thermal optimization through better heat spreading between the blocks, but these optimizations can also impact the throughput of a microarchitecture, measured in terms of the number of instructions per cycle (IPC). In nanometer technologies, global buses can have multicycle delays that depend on the positions of the blocks, and it is important for a floorplanner to be microarchitecturally-aware to be sure that thermal and IPC considerations are appropriately balanced. This paper proposes a methodology for thermally-aware microarchitecture floorplanning. The approach models the interactions between the IPC and the temperature distribution, and incorporates both factors in the floorplanning cost function. Our approach uses transient modeling and optimizes both the peak and the average temperatures, and employs a design of experiments (DOE) based strategy, which effectively captures the huge exponential search space with a small number of cycle-accurate simulations. A comparison with a technique based on previous work indicates that the proposed approach results in good reductions both in the average and the peak temperatures for a range of SPEC benchmarks.

Categories and Subject Descriptors: B.7.2 [Hardware]: Integrated Circuits - Design Aids

General Terms: Optimization, Experimentation

Keywords: Microarchitecture, Floorplanning, Transient Analysis

1. INTRODUCTION

Due to rapid increases in on-chip power and integration densities, operating temperatures have become an important concern in high performance integrated circuits in nanometer technologies. A high temperature can affect the reliability of a circuit, thus reducing its lifetime [1], through phenomena such as electromigration and Negative Temperature Bias Instability (NBTI). With every process generation, circuit performance becomes more sensitive to thermal effects due to the decreasing limits on the maximum junction temperature [2]. In addition, the temperature dependence of the leakage power results in an undesirable positive feedback, commonly referred to as *thermal runaway*, which could even lead to catastrophic chip failures. While advanced [3] packaging solutions can result in enhanced heat removal capabilities, the costs associated with these solutions are typically prohibitive. Therefore, it is important to develop temperature-conscious design techniques that alleviate on-chip thermal problems.

On-chip temperature distributions depend not only on the total power dissipation, but also on the spatial distribution of the power sources and the material properties of the medium that permit vertical and horizontal heat transfer in a chip. Physical design methods, such as floorplanning and placement, can impact the thermal profile of a chip by altering the spatial distribution of power sources, indicating a scope for improvement through better heat spreading that evens the temperature distribution on the chip. In addition, physical design optimizations can complement other thermal- and power-aware design [4] techniques implemented at a higher, architecture level such as Dynamic Thermal Management (DTM) [5].

The topic of thermally-aware floorplanning/placement has attracted some attention in the last few years, both at the circuit and microarchitecture levels. The primary difference between circuit and architecture level treatments is the level of knowledge about the spatial distribution of power. At the architectural level, the circuit is defined only in terms of large functional blocks and coarse estimates of power are available, while at the circuit level [6, 7, 8], the power consumptions of individual macro cells or blocks are all well known, and more accurate estimations are possible. However, there are many more flexibilities at the architectural level that permit significant design changes that reduce the overall power and temperature distribution.

This work focuses on the interactions between microarchitecture design and physical design, in particular, floorplanning, to explore performance-temperature tradeoffs. In the nanometer regime, the choice of a floorplan can significantly affect the performance of a processor, measured in terms of the number of instructions per cycle (IPC) [9, 10, 11, 12]. The chief culprit is the delay associated with global wires, such as buses, which can have multicycle delays [13] , thus requiring *wire-pipelining* [14] in order to support high operating frequencies. Moreover, the fluctuations in the IPC can change the activity patterns of the blocks, resulting in variations in the power densities. In other words, floorplanning can affect the temperature profile not only through heat spreading but also because the spatial and temporal distributions of power densities vary due to wire-pipelining. A good floorplanning strategy must therefore consider such interaction between IPC and power (and hence temperature) and jointly optimize both the performance and temperature objectives.

A few recent works [15, 16, 17, 18] propose techniques for thermal-aware microarchitecture floorplanning. While these indicate a welcome progress, they suffer from two drawbacks:

- They do not model the IPC-power interaction in the floorplanning step and assume that the block power consumptions are layout independent. Specifically, the power densities that

*This work was supported in part by a gift from Intel Corporation, by the NSF under award CCCR-0205227, by the Minnesota Supercomputing Institute, and by the University of Minnesota Digital Technology center.

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.
ISLPED'06, October 4–6, 2006, Tegernsee, Germany.
Copyright 2006 ACM 1-59593-462-6/06/0010 ...$5.00.

are obtained for a zero-bus-latency scenario, which typically represents the worst case for dynamic power (and the best case for IPC), are assumed to be valid for all floorplans irrespective of the amount of pipelining required by the buses, and this can result in overestimation of the temperature.

- They attempt to minimize the steady-state temperature of a chip. However, steady-state can only occur when the power dissipation is constant, which may not be true in general since programs tend to exhibit phases of varying activities [19]. In such a case, a transient modeling [20] provides a better picture of the thermal behavior of the chip: the execution times of the standard benchmarks that are used in simulations, such as SPEC [21], are typically in the range of seconds, which are significantly larger than typical thermal time constants, making it imperative to model transients. In addition, transient modeling also captures an accurate depiction of the dependence of leakage current on temperature.

  A better strategy may be to focus on minimizing the peak transient temperature over the entire execution time of a program. Furthermore, besides the peak temperature, it is useful to capture the temporal average of the temperature distribution, since many reliability mechanisms depend on this.

  Although some of the previous approaches do consider the temperature transients, the emphasis is on modeling the impact of temperature on leakage power, only a small portion of the execution time is considered for analysis, and the goal of floorplanning is to minimize the steady-state temperature.

In this paper, we propose a methodology for multiobjective microarchitecture floorplanning, where the objectives are minimizing the temperature (both average and peak), based on transient analysis, and maximizing the performance (IPC). Our approach models the impact of wire-pipelining (i.e., changes in the IPC, on power densities in the floorplanning step) and temperature-leakage power dependencies. For the purposes of a complete transient analysis that considers the entire execution times of the programs, we use a larger timestep than those employed in the limited-time analyses of [15, 16, 17, 18]. Since the floorplanning that we address involves big microarchitecture blocks, which have larger time constants than ordinary cells, the temperatures change at a slow rate, in which case, a large timestep, which reduces the analysis time by a tremendous amount, can be chosen without much loss in accuracy.

2. THERMAL ESTIMATION

A key component of a thermally-aware design methodology is a framework to estimate the temperature distribution of a chip. In the thermal analysis context, a chip can be viewed as a multilayered grid network, essentially a discretization of the chip geometry, where the nodes of the network correspond to the centers of the grids, and the connections between the nodes represent the heat flow paths in the chip. In such a set-up, the power sources $\vec{P}$ are located at the nodes of the network and based on the duality of electricity and heat transfer, the temperature distribution of the network is governed by the following differential equation:

$$\vec{C} \cdot \frac{d\vec{T}}{dt} + G \cdot \vec{T} = \vec{P} \quad (1)$$

where G is the thermal conductance matrix of the network, $\vec{T}$ is the temperature distribution of the nodes of the network. The first term on the LHS of (1) represents the transient behavior of the temperature, with $\vec{C}$ modeling the thermal capacitances. Several techniques for thermal analysis have been proposed in the past, some of which can be found in [22].

3. AVERAGE TEMPERATURE

Figure 1 shows two possible transient scenarios for a circuit, where the maximum transient temperature of the circuit is plotted against time elapsed. Although the curve of Figure 1(a) has a lower peak than that of Figure 1(b), Figure 1(b) offers a better average, where the curve is below that of Figure 1(a) for a majority of the time. As noted in [1], the reliability or mean time to failure (MTTF) decreases exponentially with temperature. Therefore, Figure 1(b) may represent a higher reliable case than Figure 1(a). In such a scenario, attempting to minimize the peak temperature can result in suboptimal thermal profiles. Nevertheless, a higher peak, seen in Figure 1(b), is not desirable due to the constraints it places on the package hardware. Therefore, a better approach may be to consider both the peak and the average temperatures in the optimization objectives, and we do this in our floorplanning methodology.

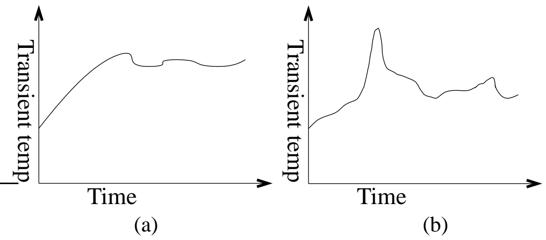

Figure 1: The results of two transient analyses of a circuit under two different implementations.

4. FLOORPLANNING FLOW

Figure 2 shows the flow of the proposed temperature-aware microarchitecture floorplanning methodology. The approach accepts a microarchitecture block configuration, a set of buses, benchmarks and a target frequency as inputs and generates a floorplan of the blocks that is both optimal in both IPC and temperature.

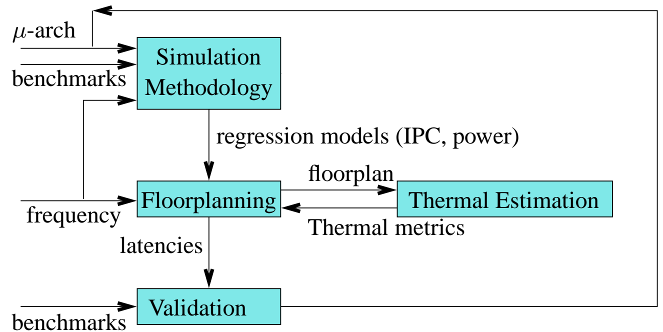

Figure 2: Thermal-aware floorplanning: design flow.

An important issue of the design flow is estimating the IPC and the block power dissipations required to generate the temperature distribution of the microarchitecture layout. In particular, the number of pipelined latencies required by each bus of the microarchitecture is proportional to its length, and therefore for every floorplan, there is a corresponding bus-latency configuration, and consequently an IPC and a power (and temperature) distribution. However, the large search space explored during floorplanning makes it virtually impossible to use simulations for each floorplan that is to be evaluated. Specifically, if each of n wires on a layout can have k possible latencies, then the cycle-accurate simulator may have to perform up to n^k simulations to fully explore the search space. We use a simulation strategy, first proposed in [12] for IPC-aware floorplanning, that is based on design of experiments (DOE) to limit the number of cycle-accurate simulations to a practical level. This approach, which reduces the number of simulations to a linear function of n, forms the preprocessing step of the flow.

Unlike [17, 18] and also our previous work on IPC-aware floorplanning [12], where the purpose of the simulations is to characterize the variations in the IPC in terms of changes in the bus latencies, the objective of the simulation strategy of Figure 2 is to model the variations in both IPC and power densities, and thus capture the IPC-power dependence. The variations are encapsulated in the

form of regression functions, with the bus latencies as variables, both for IPC and power.

The floorplanner is based on a simulated annealing (SA) framework and uses the regression models to optimize a cost function that is a weighted sum of, besides traditional objectives such as area and aspect ratio, the IPC[1] and the thermal terms, both the peak and average temperatures, as described in section 3.

After every SA move, the floorplanner estimates the block power densities from the regression models and passes them along with the corresponding floorplan to the thermal simulator, which in turn returns the thermal metrics that are part of the cost function. The performance and thermal profile of the resultant layout can then be determined from cycle-accurate simulations. In addition, the entire design flow of Figure 2 may be repeated for several microarchitectural block configurations to identify the optimal configuration.

4.1 Microarchitecture and simulator

The microarchitecture that we employ in this work is based on the DLX architecture [23] and resembles a real processor, Alpha 21362 [24]. The configuration and the corresponding functional blocks are shown in Table 1 and Figure 3, respectively. The instruction fetch and decode blocks are labeled as fet and dec, respectively, while $il1$ and $dl1$ are the level-1 instruction and data caches, respectively. The instruction and data translation look-aside buffers (TLB) are indicated as $itlb$ and $dtlb$, respectively, while $l2$ is the unified level-2 cache. The block ruu is the register update unit, which contains the reservation stations and issue logic, while lsq represents the load store queue. The register file is shown as reg, whereas $bpred$ is the branch predictor. The blocks $iadd1$, $iadd2$, $iadd3$, $imult$, $fadd$ and $fmult$ are the instruction execution units. The figure also shows the 22 buses that can impact the performance (IPC) and block power densities of the processor, when pipelined.

Parameter	Value
Fetch width	8 instrs/cycle
Issue width	8 instrs/cycle
Commit width	8 instrs/cycle
RUU entries	128
LSQ entries	64
IFQ entries	16
Branch pred	comb, 4K table 2-lev 2K table, 11-bit 2K BHT
BTB	512 sets, 4-way
IL1	64K, 64B, 2-way LRU, latency: 1
DL1	32K, 32B, 2-way LRU, latency: 1
L2	2M, 128B, 4-way latency: 12
ITLB, DTLB	128 entries Miss latency: 200

Table 1: Block configuration of the processor.

For estimating the IPC and power data, we use Wattch [25], which is based on `sim-outorder` [23] simulator. The impact of the bus latencies is modeled as dummy pipeline stages in the simulator and the latencies are made configurable. The remainder of this section explains each step of the flow of Figure 2 in detail, and we tie the description to the microarchitecture of Figure 3.

4.2 Simulation strategy

Statistical *design of experiments* is an approach that characterizes the response of a system in terms of changes in the factors which influence the response of the system. The basic idea is to conduct a set of experiments, in which all factors are varied systematically over a specified range of acceptable values, such that the experiments provide an appropriate sampling of the entire search space. The subsequent analysis of the resulting data will identify the critical factors, the presence of interactions between the factors, etc. In this work, the system is a microarchitecture, such as that shown in Figure 3, the response is the IPC/power, and the factors are the latencies of the buses of the microarchitecture.

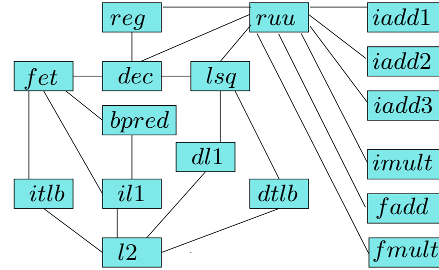

Figure 3: The functional blocks of the microarchitecture. The lines between the blocks represent the buses of the processor.

Since it is impractical to fully explore the exponential search space, even when the number of factors (buses) is small ($n = 22$), we employ a *fractional factorial* design [26] to reduce the number of simulations. In addition, since both power and IPC depend on the same set of variables, i.e., bus latencies, a single design can be used to characterize both responses.

An important advantage of factorial designs is the ability to model and estimate interactions between the factors. We have identified a few potential significant interactions, which resulted from the nature of wire-pipelining models integrated into the simulator:

- We have incorporated functional unit scheduling in the simulator. Specifically, the number of latencies inserted on the three buses between the register update unit and the three integer adders can be different, and while issuing an integer add instruction, of all the available units, the one with the least latency is chosen. This indicates possible significant (two and three factor) interactions, which need to be estimated.
- In the decode stage, the number of extra pipeline stages to be inserted is modeled as a maximum function of the latencies of the buses $dec - reg$, $dec - ruu$ and $ruu - reg$ (refer to Figure 3). Such a nonlinear function implies significant (two and three) factor interactions among these three factors.

In this work, we use a two-level resolution III fractional factorial design [26], where the two levels correspond to the lowest and highest values (extremes) for the bus latencies, which can be obtained by assuming worst-case and best-case scenarios for the corresponding wire lengths. For n factors, the number of experiments required is equal to the nearest highest power of two, which turns out to be 32 for our work, since $n = 22$[2].

The floorplanning approaches of [17, 18], although do not model the dependence of power on bus latencies, propose simulation strategies to capture the IPC impact of bus latencies. The method of [17] constructs linear regression models using simulations by varying each latency independently, whereas [18] uses latency-independent models to capture the IPC variations. While these have been demonstrated to work well for IPC since a reasonably accurate relative ordering of variables is sufficient [27], such one-at-a-time approaches may not effectively track absolute variations, required in the case of power, as compared to the DOE approach [28] used in this work. The reason for the requirement of "absoluteness" is that the power and temperature may not have a perfect correlation [29], and power-criticality does not necessarily imply temperature-criticality. This lack of fidelity[3], coupled with the dependence of leakage current on temperature, indicates that any error in power estimation can result in significant inaccuracies in the temperature computations.

[1] The cost function actually includes the CPI, or Clocks Per Instruction, the reciprocal of IPC, since the objective is to maximize the IPC.

[2] Although resolution III designs work under the notion of negligible interactions, since n is less than the number of simulations, we have an *unsaturated design*, and this allows the estimation of a few interactions by projecting them as additional factors.

[3] A well known case where the property of fidelity holds is Elmore delay modeling: although the estimated delays may be inaccurate, the metric accurately tracks the variations in the delays.

4.2.1 *Reducing simulation times*

Cycle-accurate simulations are inherently slow and most SPEC benchmarks with `reference` input sets, when simulated can take days to complete. Therefore, although the resolution III design strategy of section 4.2 requires a small number of simulations, the run time of each simulation is still an issue. To speed up the simulations, we utilize SMARTS [30], a periodic sampling technique, which works well both for throughput (IPC) and power/energy, particularly for the SPEC benchmarks.

4.2.2 *Power/IPC regression models*

The SMARTS technique involves fastforwarding program segments between successive samples chosen for detailed simulation. However, the transient modeling requires that the block power densities be collected periodically for every timestep. For this, we extrapolate the power data collected for each sample for the succeeding fastforwarded portion. While we do not offer a proof, the concept of periodic sampling is inherently based on this assumption, and there is empirical evidence that it works well at least for average power/energy estimation [30].

The total execution time obtained from a simulation is then segmented into slots of size equal to the transient analysis timestep. Therefore, the data collected from the simulation can be arranged as an array P indexed by the timestep and the block number, i.e., the entry $P(a, b)$ of the array corresponds to the power consumption of block b (one of the 17 blocks of Figure 3) during timestep a. Since 32 simulations performed (per benchmark), there are 32 such tables. For each entry $P(a, b)$ (per benchmark), a regression model is constructed from the 32 values [26], based on least-squares approximation, where the variables are the bus latencies. Equation (2) shows one such a model, constructed to estimate the power dissipation at entry $P(a, b)$, where β_is represent the regression coefficients computed from the 32 values obtained for the correspond entry (a, b). Each x variable in (2), say x_i, represents an encoding of the latency of bus i, l_i, where the minimum and the maximum latencies are coded as -1 and +1, respectively, and $\mathcal{I}$ is the set of interactions described in section 4.2.

$$
\begin{aligned}
x_i &= -1 + \left(\frac{2 \cdot l_i}{\min(i) + \max(i)}\right), \quad 1 \le i \le 22 \\
P(a,b) &= \beta_0 + \sum_{i=1}^{22} \beta_i \cdot x_i + \sum_{(ij) \in \mathcal{I}} \beta_{ij} \cdot x_i \cdot x_j + \\
&\quad \sum_{(ijk) \in \mathcal{I}} \beta_{ijk} \cdot x_i \cdot x_j \cdot x_k \qquad (2)
\end{aligned}
$$

An IPC regression model is similarly constructed for each benchmark from the statistics gathered from the 32 simulations. In addition, although we construct separate regression functions for IPC and power, since the associated variables are the same, a direct relation between the power and the IPC estimates can be obtained by composition of the regression functions.

4.3 Temperature estimation

We use HotSpot [29] in this work for thermal analysis. In this approach, the nodes of the multi-layered thermal network described in section 2 are the centers of the blocks of the microarchitecture. The tool also provides a framework for transient modeling, and accepts a floorplan, the length of the timestep, and the block power dissipations averaged over each timestep as inputs. The differential equation (1) is solved at each timestep to estimate the new set of temperatures (with the initial conditions being those of the previous timestep). The leakage power component of the succeeding timestep can then be updated using the new temperatures.

Choice of timestep

In general, the smaller the timestep, the higher is the accuracy of the transient analysis. It is clearly impractical to perform the analysis for every clock cycle of execution, and the authors of HotSpot suggest a size of about 10000 clock cycles at a frequency of 3GHz, i.e., a timestep of about $3.3\mu s$. Although this reduces the analysis time by a significant factor, it still makes it prohibitive to incorporate transient analysis into the iterative scheme of the floorplanning step, where thousands of floorplans are evaluated.

To solve this issue, we choose an interval of one million clock cycles, which amounts to about a few hundreds of microseconds for gigahertz frequencies, and this can possibly affect the accuracy of the computations. However, since the focus of the optimizations involves relatively larger microarchitecture blocks (than the macro cells considered in circuit level optimizations), the thermal RC constants tend to be higher, typically in the range of tens of milliseconds, and this indicates a minimal loss of accuracy since each time constant still involves a high number of timesteps. For instance, ruu, a medium sized block of the microarchitecture of Figure 3, has a time constant of about 120ms. As noted in [29], the temperatures rise slowly, and it takes more than 100,000 clock cycles to observe an increase of as small as 0.1°C in the temperature. In addition, we use a single iteration to solve the differential equation of (1) during each timestep.

4.4 Floorplanning cost function

The floorplanner is based on simulated annealing (SA), which uses the power and IPC regression models built out of the simulation methodology described in section 4.2 in the cost function. We use PARQUET [31], a floorplanner available in the public domain.

The cost function C is a weighted sum of, besides the chip area ($Area$) and the aspect ratio (AR), the average (T_{avg}), and the peak (T_{peak}) transient temperatures, as shown below:

$$C = W_1 \cdot Area + W_2 \cdot AR + W_3 \cdot CPI + W_4 \cdot (T_{avg} + T_{peak}) \quad (3)$$

where the Ws represent the relative weights of the optimization terms. It can be seen that the cost function actually contains CPI, the reciprocal of IPC, since the objective is maximizing IPC. If N_t is the number of timesteps in the transient analysis and T_i is the maximum of the block temperatures at timestep i, the average and the peak temperatures are determined as follows:

$$T_{avg} = \frac{1}{N_t} \sum_i T_i \text{ and } T_{peak} = \max_i T_i \quad (i = 1, 2, \cdots, N_t)$$

5. VALIDATION

5.1 Benchmarks

We choose a set of eight SPEC 2000 benchmarks, which, along with the corresponding instruction counts of the `reference` input sets, are shown in Table 2. The benchmarks are chosen because of their distinct instruction mixes. For instance, `mesa` has a high percentage of conditional branches, while `gcc` has a very large number of memory operations. All benchmarks are complied at optimization level O3 using the SimpleScalar version of gcc.

Benchmark	Type	Instr. (B)
gzip	Integer	63
vpr	Integer	11
gcc	Integer	35
mesa	Floating-point	305
art	Floating-point	54
equake	Floating-point	175
parser	Integer	301
bzip2	Integer	94

Table 2: Benchmarks from the SPEC 2000 suite, along with the reference instruction counts.

5.2 Experimental set up

The areas of the blocks of Figure 3 are estimated using [32]. The total area of the chip is about 2cm^2 at 90nm technology, with the L2 cache consuming about 70% of the area. Only the chip core that also includes the L1 caches is considered during floorplanning, and the L2 cache is wrapped around the core floorplan, just as is done in [17] and Alpha 21362 [24]. We choose a frequency of 4GHz for

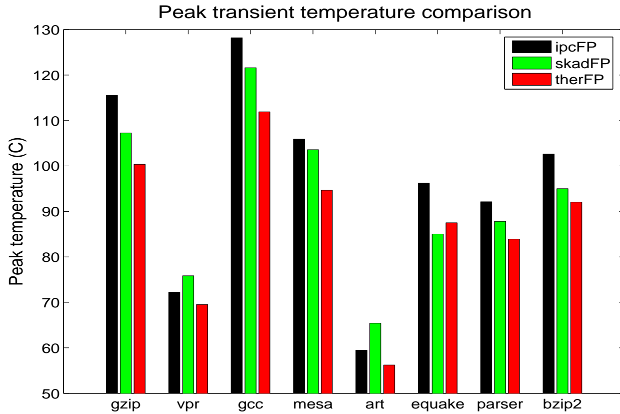

Figure 4: Comparison of the peak transient temperatures obtained from the three floorplanning approaches.

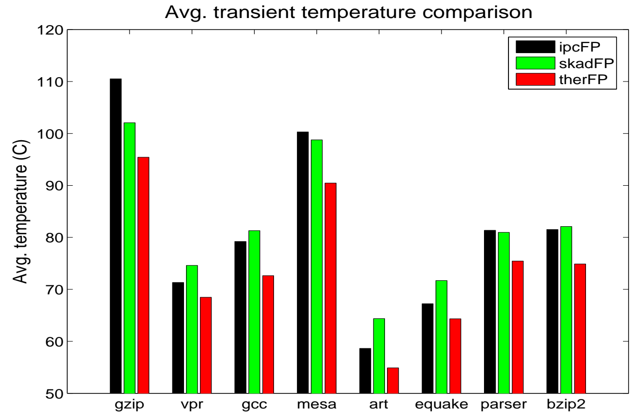

Figure 5: Comparison of the average temperature metric for the three floorplanning scenarios.

our experiments, and therefore, a timestep of 250μs. For the bus latency ranges that are to be used in the resolution III design, the low value is chosen to be 0, depicting the best case placement of the connecting blocks. The high value is chosen to equal the corner-to-corner latency of the chip core, which is found to be 6 clock cycles at 4GHz, based on the computations of [17].

For each of the eight SPEC benchmarks of Table 2, 32 cycle-accurate simulations are performed, as prescribed by the resolution III design. Although the floorplan can be optimized for each of the benchmarks, in practice, a processor must be optimized so that it performs well over a range of benchmarks. In other words, one must generate a single floorplan for the processor that is, on average, optimal over all benchmarks. For this purpose, the IPC and power regression coefficients are averaged over the eight benchmarks to generate a new set of regression models that are used in the optimization process to generate a single floorplan. In addition, for the purposes of transient analysis, we use an initial temperature of 40°C for the blocks of the architecture.

We integrate HotSpot with Wattch to enable thermal analysis during simulations. *Although we use SMARTS to speed up the simulation strategy of section 4.2, detailed cycle-accurate simulations, without fastforwarding any program portions, for the entire execution times of the benchmarks are performed for validating the floorplanning solutions.* In addition, we use a relatively smaller timestep of 10000 clock cycles, as compared to that of 1000000 cycles used during optimization, for transient analysis, i.e., the power data are averaged over every 10000 clock cycles and are provided to the HotSpot solver to determine the set of temperatures.

5.3 Results

We compare our proposed thermal floorplanning technique with two other approaches. The long run times of the simulations is the main obstacle that limits the number of comparisons that can be made. The floorplanners that are compared listed below:

- **ipcFP:** IPC only floorplanning, the cost function of the floorplanning does not consider any thermal issues.
- **therFP:** Our proposed temperature-aware floorplanning, where the cost includes IPC and both the average and peak transient temperatures, along with the core area and aspect ratio.
- **skadFP:** A temperature-aware floorplanning approach based on [17]: the block power densities are assumed to be independent of the bus latencies. In addition, the cost includes only the peak transient temperature, along with the IPC, area and aspect ratio[4].

[4] We choose to include the peak transient temperature in our implementation of [17] for convenience. Moreover, although the original implementation attempts to minimize the steady-state temperature, the authors use peak transient temperature as a metric of their validation process.

For **therFP** and **skadFP**, we choose a weight of 0.4 for both IPC and temperature, and 0.1 for area and aspect ratio, i.e., $w_1 = w_2 = 0.1, w_3 = w_4 = 0.4$ in (3). For the IPC-only floorplanner **ipcFP**, we have $w_1 = w_2 = 0.1, w_3 = 0.8, w_4 = 0$. The idea is to provide a greater emphasis on the primary issues, the IPC and the temperature, while still attempting to limit the total area.

The white spaces (WS) and the aspect ratios (AR) of the floorplans obtained using the three approaches, shown in Table 3, imply that all of the three result only in a small increase in the area. For instance, a core WS of about 6% in **therFP** indicates an overall increase of 1.5% in the chip area (equivalent to 2.03cm^2). Besides, both **skadFP** and **therFP** produce floorplans of almost perfect AR.

Case	Core WS (%)	Core AR
ipcFP	5.33	1.15
skadFP	7.60	1.02
therFP	6.21	1.03

Table 3: Comparison of white space (WS) and aspect ratio (AR) for the three floorplanners.

Figure 4 plots the peak transient temperatures obtained using the three floorplanners for various benchmarks. The graphs show that, for a majority of the benchmarks, both our **therFP** and **skadFP** obtain good reductions in the peak temperatures when compared to **ipcFP**, and this is particularly true for those which exhibit high temperatures. In addition, **therFP** outperforms **skadFP** for almost all benchmarks despite not explicitly attempting to minimize the peak temperature as is done in **skadFP**. For instance, for the benchmark `gcc`, the floorplan generated by **therFP** reduces the peak by about 16°C as compared to **ipcFP**, while it is about 7°C for **skadFP**.

Figure 5 compares the average transient temperatures obtained using the three approaches. The plots indicate that **therFP** outperforms both **ipcFP** and **skadFP** by significant amounts for all benchmarks. Reductions of about 9°C and 6°C are obtained over **ipcFP** and **skadFP**, respectively, for `gcc`. In addition, since the floorplans are optimized for the average cases and not specifically for each benchmark, the optimization potential for each benchmark may not be fully exploited. Furthermore, benchmarks that have low power profiles such as `art` and `vpr` do not offer much scope for optimization, the resultant improvements tend to be small, and in fact, **skadFP** worsens the thermal profiles obtained for `art` and `vpr`, where both the average and the peak temperatures are higher than those of **ipcFP**, as shown in Figures 4 and 5.

Finally, Figure 6 depicts the performance (IPC) degradation obtained in **therFP** and **skadFP** due to the inclusion of thermal issues in the cost function, besides performance. On an average, both **therFP** and **skadFP** result in almost identical IPCs, and these are about 6% less than the **ipcFP** case, where no thermal metrics are considered in the cost function.

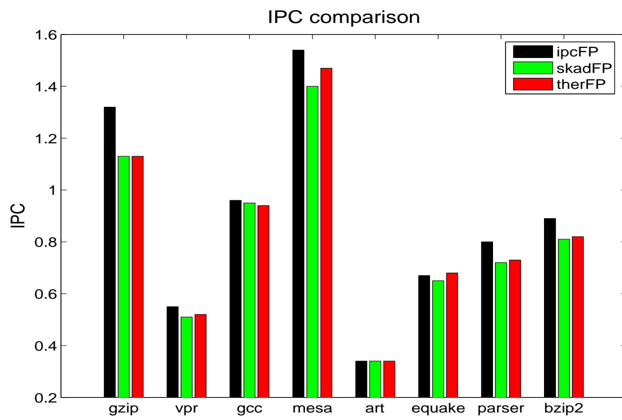

Figure 6: Comparison of the throughput (IPC) metric for the three floorplanning scenarios.

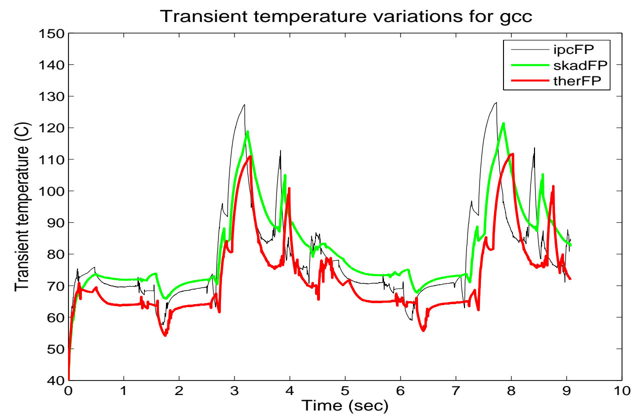

Figure 7: Transient curves for the benchmark `gcc` obtained using the three floorplans.

Figure 7 depicts the temporal distributions of the temperature for a the entire execution time of a benchmark, `gcc`, for the three floorplanners. The figure shows that **therFP**, where the transient curve is below those of the other cases, produces the best profile among the three cases, as is also shown in Figures 4 and 5. In addition, two observations can be made from this figure:

- A steady-state never occurs in all cases, even after a time as long as 10 seconds, and this is true for most of the benchmarks, except `art` and `vpr`, which have low averages, as seen in Figure 5, and do not exhibit significant variations.
- Although **skadFP** exhibits a lower peak than **ipcFP**, the corresponding curve is consistently higher than that of **ipcFP**, resulting in a higher average. This discrepancy, which is also observed for the benchmark `equake` as illustrated in Figures 4 and 5, indicates that the peak and average temperatures may not have a perfect correlation, which underlines the importance of including temporal average in the floorplanning objectives.

6. CONCLUSION

Thermal issues have become an important concern in microprocessors designed in nanometer technology nodes. This paper presented a strategy for thermally-aware floorplanning for microprocessors, where the optimization objectives also include the throughput (IPC) issues. The approach also models the IPC-power interaction, and uses a complete transient analysis that captures a thermal profile of a chip in a better way than the steady-state approach, during the floorplanning optimization. The results indicate good improvements both in the average and peak temperatures when compared to an approach derived from a previous work.

7. REFERENCES

[1] J. Srinivasan *et al.*, "The impact of technology scaling on lifetime reliability," in *Proc. Dependable Systems and Networks*, pp. 70–80, Jun. 2004.

[2] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors," 2001. Available at `http://public.itrs.net`.

[3] S. H. Gunther *et al.*, "Managing the impact of increasing microprocessor power consumption," *Intel Tech. Journ.*, vol. 5, pp. 1–9, May 2001.

[4] D. M. Brooks *et al.*, "Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors," *IEEE Micro*, vol. 20, pp. 26–44, Aug. 2000.

[5] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. ACM HPCA*, pp. 171–182, Feb. 2001.

[6] W.-L. Hung *et al.*, "Thermal-aware floorplanning using genetic algorithms," in *Proc. IEEE ISQED*, pp. 634–639, Mar. 2005.

[7] J. Cong *et al.*, "Thermal-driven floorplanning algorithm for 3D ICs," in *Proc. IEEE ICCAD*, pp. 306–313, Nov. 2005.

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

[9] M. Ekpanyapong *et al.*, "Profile-guided microarchitectural floorplanning for deep submicron processor design," in *Proc. ACM DAC*, pp. 634–639, Jun. 2004.

[10] C. Long *et al.*, "Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects," in *Proc. ACM DAC*, pp. 640–645, Jun. 2004.

[11] A. Jagannathan *et al.*, "Microarchitecture evaluation with floorplanning and interconnect pipelining," in *Proc. ACM ASPDAC*, pp. 1–8, Jan. 2005.

[12] V. Nookala *et al.*, "Microarchitecture-aware floorplanning using a statistical design of experiments approach," in *Proc. ACM DAC*, pp. 579–584, Jun. 2005.

[13] P. Saxena *et al.*, "Repeater scaling and its impact on CAD," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 23, pp. 451–463, Apr. 2004.

[14] P. Cocchini, "Concurrent flip-flop and repeater insertion for high performance integrated circuits," in *Proc. IEEE ICCAD*, pp. 268–273, Nov. 2002.

[15] Y. Han *et al.*, "Temperature aware floorplanning," in *Proc. Temperature-Aware Computer Systems*, Jun. 2005.

[16] Y. W. Wu *et al.*, "Joint exploration of architectural and physical design spaces with thermal consideration," in *Proc. ACM ISLPED*, pp. 123–126, Aug. 2005.

[17] K. Sankaranarayanan *et al.*, "A case for thermal-aware floorplanning at the microarchitectural level," *Journ. ILP*, vol. 8, pp. 8–16, Oct. 2005.

[18] M. Ekpanyapong *et al.*, "Thermal-aware 3D microarchitectural floorplanning," Technical Report GIT-cercs-04-37, Georgia Institute of Technology, Atlanta, Apr. 2004.

[19] C. Iski and M. Martonosi, "Phase characterization for power: Evaluating control-flow-based and event-counter-based techniques," in *Proc. ACM HPCA*, pp. 281–291, Feb. 2003.

[20] T. Wang and C. C. Chen, "3-D thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 21, pp. 1434–1445, Dec. 2002.

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

[22] Y. Zhan and others, "Electrothermal analysis and optimization techniques for nanoscale integrated circuits," 2006. Available at `http://www.aspdac.com/aspdac2006/archives/2d/index.html`.

[23] D. C. Burger and T. M. Austin, "The SimpleScalar tool set, version 2.0," Technical Report CS-TR-97-1342, The University of Wisconsin, Madison, Jun. 1997.

[24] P. Bannon, "Alpha 21364: A scalable single-chip SMP," 1998. Available at `http://www.digital.com/alphaoem/microprocessorforum.htm`.

[25] D. Brooks *et al.*, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. ACM ISCA*, pp. 83–94, Jun. 2000.

[26] D. C. Montgomery, *Design and analysis of experiments*. New York, NY: John Wiley, 1991.

[27] V. Nookala *et al.*, "Comparing simulation techniques for microarchitecture-aware floorplanning," in *Proc. IEEE ISPASS*, pp. 80–88, March 2006.

[28] V. Czitrom, "One-factor-at-a-time versus designed experiments," *The American Statistician*, vol. 53, pp. 126–131, May 1999.

[29] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *Proc. ACM ISCA*, pp. 2–13, Jun. 2003.

[30] R. E. Wunderlich *et al.*, "SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling," in *Proc. ACM ISCA*, pp. 84–97, Jun. 2003.

[31] S. N. Adya and I. L. Markov, "Fixed-outline floorplanning through better local search," in *Proceedings of IEEE ICCD*, pp. 228–334, Oct. 2001.

[32] M. Steinhaus *et al.*, "Transistor count and chip-space estimation of simplescalar based microprocessor models," in *Proc. Complexity-Effective Design*, Jun. 2001.