Temperature Aware Floorplanning

Yongkui Han, Israel Koren and Csaba Andras Moritz
Department of Electrical and Computer Engineering
University of Massachusetts,Amherst, MA 01003
E-mail: {yhan,koren,andras}@ecs.umass.edu

ABSTRACT

Power density of microprocessors is increasing with every new process generation resulting in increasingly higher maximum chip temperatures. The high temperature of the chip greatly affects its reliability, raises the leakage power consumed to unprecedented levels, and makes cooling solutions significantly more expensive. The maximum temperature of a block in a chip depends, however, not only on its own power density, but also on the power density of the adjacent blocks. Consequently, the placement of architectural blocks, or a particular floorplan selected for a given chip, can affect the maximum temperature of the chip considerably. This paper analyzes the impact of floorplanning on the maximum temperature by using the Alpha and Pentium Pro microprocessors as examples. We show that the difference between the maximum temperatures of two different floorplans can be as high as $37^{\circ}C$. We have modified a floorplanning tool to include temperature as an objective for block placement to reduce the hot spot temperature. We show that it is possible to find a floorplan that can reduce the maximum temperature of the chip by up to $21^{\circ}C$ compared to the original floorplan while maintaining comparable performance.

1. INTRODUCTION

Power density is increasing in each generation of microprocessors since feature size and frequency are scaling faster than the operating voltage. Power density directly translates into heat, and consequently processors are getting hotter. For example, Pentium 4 chips generate more heat than a kitchen hotplate and the company's projections show that the heat generated by its processors will increase sharply in the coming years, approaching that of the core of a nuclear power plant, unless solutions to this problem can be found [6].

In order to keep the chip temperature below a certain limit, the heat generated by the processor must be removed from the die. Since the cost of removing heat is increasing at about the same rate as power density, reducing the maximum temperature in the chip can reduce the cost of the cooling system, which constitutes a major component of the overall cost.

The high temperature of the chip also greatly affects its reliability. The reliability of the chip reduces exponentially as the temperature increases. The time to failure has been shown to be a function of $e^{(-Ea/kT)}$, where Ea is the activation energy of the failure mechanism being accelerated by the increased temperature, k is Boltzmann's constant, and T is the absolute temperature [2]. At elevated temperatures a silicon device can fail catastrophically. Even if it does not, its electrical characteristics frequently undergo intermittent or permanent changes. The life of an electronic device is also directly related to its operating temperature. Each $10^{\circ}C$ temperature rise reduces component life by 50% [2]. Therefore, it is recommended that computer components be kept as cool as possible for maximum reliability and longevity.

It is also expected that leakage power consumption will be comparable to dynamic power consumption within the next few process generations. Leakage power is highly dependent on temperature. Therefore reducing the temperature of the chip will result in less leakage.

With increases in power density of digital circuits, heat dissipation is fast becoming a limiting factor in microprocessor design. Recently temperature aware designs have been proposed and studied [13]. Skadron et al. propose temperature aware microarchitectures [18][19]. They have developed the HotSpot software [1], which is a tool to calculate the temperature distribution among different blocks on the CPU chip.

Donald et al. study temperature aware design issues for SMT (Simultaneous MultiThreading) and CMP (Chip Multiprocessing) architectures [11]. They find that large temperature gradients are prominent in both architectures but both show promise for temperature aware enhancements to mitigate this problem. Li et al. evaluate the thermal efficiency of SMT and CMP architectures [17]. They show that SMT and CMP exhibit similar peak operating temperatures, but the mechanism by which they heat up are quite different, hence the best thermal management mechanisms are also different for SMT and CMP.

Chaparro et al. study thermal-aware clustered microarchitecures [8]. They propose and evaluate several techniques including temperature aware steering techniques and cluster hopping in a quad-cluster superscalar microarchitecture. They claim that 30% reduction in leakage power and 8% reduction in average peak temperature can be achieved at the expense of a slowdown of only 5%.

Chu et al. introduce a new combinatorial optimization problem, matrix synthesis problem [9], to model the thermal placement problem. They present several provably good approximation algorithms for the solution. Our paper is different from that paper in three ways. First, they focus on a theoretical and simplified floorplanning problem where all blocks have the same size. Secondly, instead of using temperature as an objective, they use the sum of power numbers of a partial floorplan. We calculate the real temperature difference between different floorplans using the HotSpot software. Third, they use randomly generated power numbers in their experiments, while we use simulated power numbers for SPEC2000 benchmarks in our experiments.

Hung et al. study thermal-aware floorplanning using genetic algorithms [14][15]. They demonstrate that their combined area and thermal optimization technique decreases the peak temperature while providing flooeplans which have comparable area as the traditional area-oriented techniques. But they do not explore the performance impact of their algorithms. In our paper, we evalu-

ate the performance of different floorplans using an interconnection model. Their studies are focused on lower circuit level, while our studies are focused on the architectural level, and we use real processors to show the impacts of different floorplans on temperature. They also use randomly generated power numbers in their experimental simulations while we do not.

In this paper we study the impact of floorplanning on the temperature of a chip. The insights learned from this paper can also be applied to devise new techniques at the architectural level. Architectural components that often affect the maximum temperature in the chip, e.g., the register file, could be banked/partitioned [10] to allow more flexibility in placement and reduced power density. We have seen similar trends in fact to reduce power consumption in caches [12][16]. Alternatively, one might consider architectures where some of these components are replicated and associated activity is distributed in a temperature-conscious way. Additional provision can be added at the circuit level to reduce the power density of such components.

Our contributions are as follows:

1. We demonstrate how different floorplans affect the maximum temperature of the chip. The temperature difference can be as large as $30^{\circ}C$.

2. We propose temperature aware floorplanning, through which, we can find a floorplan that can reduce the maximum temperature of the chip by up to $21^{\circ}C$ compared to the original floorplan while maintaining comparable performance.

3. We propose to use a heat diffusion measure as an approximation of the temperature. This can considerably reduce the complexity of computing the temperature while still producing good results.

The rest of the paper is organized as follows. In Section 2, we provide the motivation for temperature aware floorplanning and demonstrate the temperature benefits of different floorplans. In Section 3, we propose temperature aware floorplanning and describe the implementation of such floorplanning based on the Parquet software. The experimental results for an Alpha microprocessor are given in Section 4. We provide the experimental results for the Pentium Pro microprocessor in Section 5. Conclusions are presented in Section 6.

2. MOTIVATION

2.1 Alpha Processor Floorplan

The HotSpot software developed at the University of Virginia is a tool that models the temperature of microprocessor chips. HotSpot allows the user to specify a processor floorplan with its functional units. From this floorplan, it creates an equivalent circuit model that represents heat transfer in a processor die with specified thermal packaging. We use the newly released HotSpot 2.0 [1], which accounts for many important effects of the thermal interface material between the die and heat spreader and has been validated against a test chip. The same Alpha processor ($0.13\mu m$ technology) floorplan used by Skadron et al. [19] is used in our experiments (shown in Figure 1). We use all 24 benchmarks from the SPEC 2000 suite [5] in our experiments.

The temperature of each processor block for the *gcc* benchmark is shown in Figure 2. We do not show the temperature of the L2 cache in the figure because the L2 cache has a considerably lower temperature than the other blocks in the processor core. The power density of each block is shown in Figure 3.

From Figure 2, we can see that the block with the maximum temperature in the chip is the integer register file *IntReg*. Its temperature is $120^{\circ}C$, and it has the highest power density, 2.798

Figure 1: The original Alpha floorplan

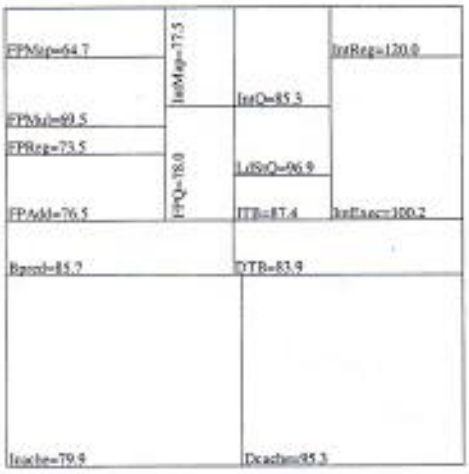

Figure 2: Temperatures of each block in the core area for the *gcc* benchmark in $^{\circ}C$

Figure 3: Power densities of each block in the core area for the *gcc* benchmark in $Watt/mm^2$

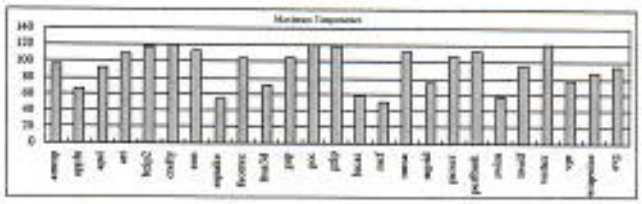

Figure 4: Maximum temperature ($^\circ C$) for the original Alpha floorplan for SPEC2000 benchmarks

$Watt/mm^2$.

Usually the block with the highest power density has the highest temperature, but it is not always true. The temperature of a block in a chip depends not only on its power density but also on the power density of the adjacent blocks. We can take blocks *IntQ* and *FPReg* as an example. The power density of *IntQ* is 0.137 $Watt/mm^2$, and the power density of *FPReg* is 0.623 $Watt/mm^2$, which is nearly 4 times larger than that of *IntQ*. However, the temperature of *IntQ* (85.3$^\circ C$) is higher by about 9$^\circ C$ than the temperature of *FPReg* (76.5$^\circ C$). This is because *IntQ* is placed near the blocks *IntReg*, *LdStQ*, and *IntExec*, all of which are hot blocks. In contrast, *FPReg* is placed near *FPMul*, *FPAdd*, both of which have relatively low power densities. This demonstrates that the placement of a block has a considerable impact on its temperature and has motivated us to study temperature aware floorplanning.

2.2 Maximum Temperature for the SPEC2000 Benchmarks

We show the maximum temperature for SPEC2000 benchmarks in Figure 4. We can see that for 12 out of the 24 benchmarks the maximum temperature of the chip is higher than 100$^\circ C$, for 8 of them the temperature exceeds 110$^\circ C$, and for 2 of them it exceeds 120$^\circ C$.

When we take a look at the hottest block in the chip, we find that it is *IntReg* for almost all SPEC2000 benchmarks expect for *applu*, *lucas*, and *mgrid*. *FPReg* is the hottest block for *applu* and *FPAdd* is the hottest block for *lucas* and *mgrid*. However, their maximum temperatures are not high (67.2$^\circ C$, 59.1$^\circ C$ and 75$^\circ C$, respectively). Since the temperature distribution among the blocks of the chip for SPEC benchmarks is similar, we select as the representative benchmark in our experiments the *gcc* benchmark, which has a maximum temperature of 120$^\circ C$.

2.3 A Manually Generated Floorplan

We first tried to manually modify the Alpha floorplan to lower the maximum temperature of the chip. Figure 5 is a new floorplan, in which we put blocks with high power density next to blocks with low power density, while maintaining the area of blocks and changing the aspect ratio of the blocks as little as possible. We show the resulting temperature of each block for the *gcc* benchmark in Figure 5. The maximum temperature and the temperature reductions for the new floorplan for SPEC2000 benchmarks are shown in Figure 6. We achieve a reduction of more than 20$^\circ C$ for many benchmarks, and an average reduction of 11$^\circ C$.

For the new floorplan, only the maximum temperature for the *vortex* benchmark is a little higher than 100$^\circ C$. The maximum temperature for all other benchmarks has been reduced to below 100$^\circ C$. This is a significant improvement in temperature.

For three benchmarks, *applu*, *lucas*, and *mgrid*, the maximum temperature has increased. Their hottest blocks in the chip are *FPReg*, *FPAdd*, and *FPAdd*, respectively. Notice, however, that

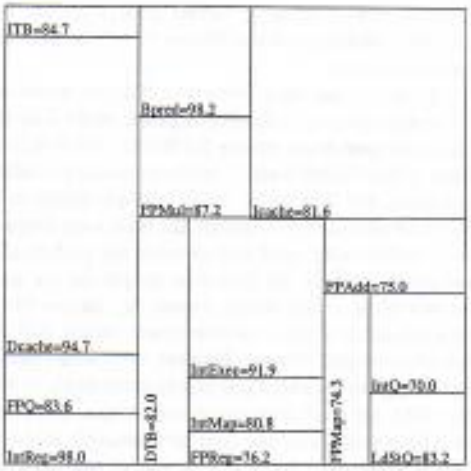

Figure 5: The block temperatures ($^\circ C$) for the manually generated Alpha floorplan

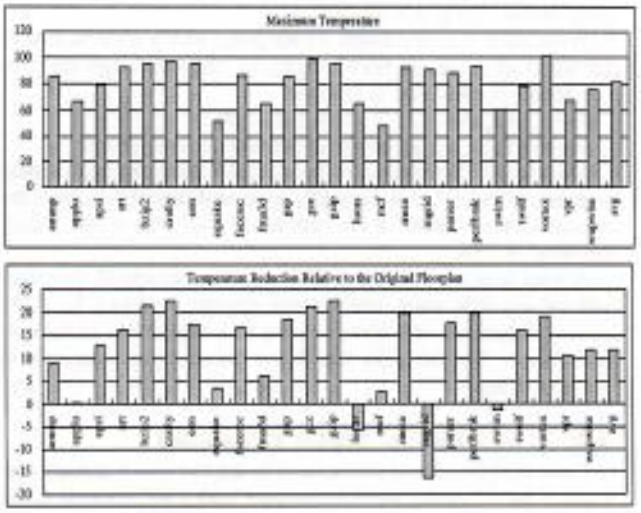

Figure 6: Maximum temperatures and temperature reductions ($^\circ C$) for the manually generated Alpha floorplan

for these benchmarks, the maximum temperature is low, so a small increase in the maximum temperature will still keep the chip reasonably cool. Their maximum temperatures are $67°C$, $65°C$, and $91°C$, respectively.

These results motivated us to further manipulate the floorplan and modify the placement of the blocks to reduce the maximum temperature in the chip.

Any block replacement may, however, affect the performance of the chip. The performance of the chip depends on the wire length of the various interconnections among the blocks. Since only approximate values of inter-block wires' lengths are usually available during floorplanning and in order to obtain a simple metric to be used in the search for an optimal floorplan, the total wire length among all blocks is traditionally used as a measure for performance during floorplanning. Clearly, the total wire length can not accurately reflect the individual signal delays among the various blocks. To increase the relevance of the total wire length metric, it is common to assign higher weights to wires between two blocks which carry timing critical signals making these blocks more likely to be placed adjacently. Still, the total wire length can at best, serve as only a first order approximation for the chip performance. In the absence of exact information regarding the criticality of individual wires in the floorplans which we have analyzed, we use the unweighted total wire length as our measure for performance. We believe however, that the principles of temperature aware floorplanning can still be demonstrated despite the inaccuracies in the wiring length measure.

2.4 Wire Length Overhead

Since the exact number of wires between any two blocks in the Alpha chip has not been available to us, we had to use the estimated interconnect matrix shown in Figure 7. This matrix focuses on data signals and ignores control signals, and is used here for illustration purposes only. The columns (and rows) of the interconnect matrix are in the order *L2_left, L2_bottom, L2_right, Icache, Dcache, Bpred, DTB, FPAdd, FPReg, FPMul, FPMap, IntMap, IntQ, IntReg, IntExec, FPQ, LdStQ, ITB*.

To calculate the wire length we adopt the widely used method of HPWL (Half Perimeter Wire Length), i.e., the wire length between two blocks is calculated as follows:

$WireLength = |x1 - x2| + |y1 - y2|$

where $x1, y1, x2, y2$ are the coordinates of their centers.

The wire length overhead of the manually drawn Alpha floorplan is shown in Table 1. The manually generated floorplan reduces the maximum temperature of the chip by $21.8°C$ with a wire length overhead of 29%.

A wire length overhead of 29% may be considered excessive and the question arises whether any attempt to reduce the maximum temperature will always result in a substantial increase in wire length. We will show that this is not necessarily the case in the next section.

2.5 The Rotated Alpha Floorplan

Since the CPU core is surrounded by the low power density portions of the L2 cache, we experimented with a 90° rotation of the core, a simple floorplan modification that is expected to result in a small wire length overhead. The rotated floorplan and the resulting temperatures for the *gcc* benchmark are shown in Figure 8. The new maximum temperature in the chip and the reduction in maximum temperature for all the 24 SPEC benchmarks are given in Figure 9. The rotated floorplan reduces the maximum temperature of the chip for the *gcc* benchmark by $16.1°C$ with a wire length overhead of only 2.18% (see Table 1). For most benchmarks, we obtain considerable temperature improvements. We can achieve

Figure 8: The block temperature ($°C$) for the rotated core Alpha floorplan

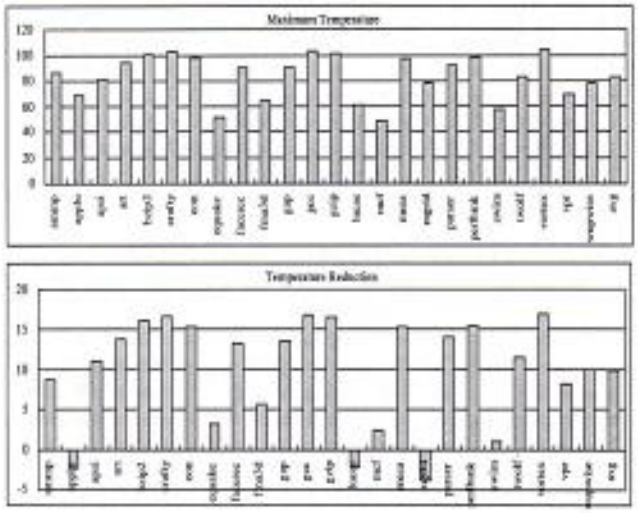

Figure 9: Maximum temperatures and temperature reductions ($°C$) for the rotated core Alpha floorplan

Figure 7: The interconnect matrix

an average reduction of $9^{\circ}C$ for the SPEC2000 benchmarks. This is somewhat smaller than the reduction achieved by the manually generated floorplan.

Manually searching for a floorplan that may further reduce the maximum temperature with a low wire length overhead is time-consuming and inefficient. We decided therefore to modify an existing floorplanning tool to allow us to generate temperature aware floorplans.

3. FLOORPLANNING WITH A TEMPERATURE OBJECTIVE

3.1 Parquet Floorplanner

For our purposes, we have selected Parquet [7], which is a floorplanning tool developed at the University of Michigan. The Parquet floorplanner is a fixed-outline, hierarchical design package and is based on the widely used sequence-pair representation. It is intended to solve multi-objective problems (area and wire length) using the simulated annealing algorithm.

We allow all the blocks of the chip to be "soft" blocks, that is, their aspect ratio can change (in a controlled manner) in each movement but their area is fixed. We next describe our temperature aware floorplanning.

3.2 Problem Description

The temperature aware floorplanning problem is that of placing n rectangular modules in the chip area satisfying the following conditions:

1. Each module i has a fixed area A_i but its height h_i and width w_i can be changed so that $h_i * w_i = A_i$. The aspect ratio of module i, h_i/w_i, must be limited to the range $r_i \leqslant h_i/w_i \leqslant s_i$ $(i = 1, ..., n)$.

2. The wiring length is calculated using an interconnection matrix $C_{n*n} = [C_{ij}]$, where C_{ij} is the number of wires connecting modules i and j.

3. The chip area is $A = H * W$, where H and W are the height and width of the chip, respectively. The chip aspect ratio is also constrained to a given range $R \leqslant H/W \leqslant S$.

4. In order to calculate the maximum temperature T, the power consumption P_i of each module is provided.

5. The objectives of the floorplanning process are: low chip area A, low total wire length L, and low maximum temperature T.

3.3 Objective Function

In Parquet, the global objective is to minimize a linear combination of the total area and the total wire length. We add the temperature to the original objective function as follows:

$Obj = C_A * A + C_W * W + C_T * T$

where C_A, C_W and C_T are the weights of the area, the wire length and the maximum temperature in the chip, respectively.

The Parquet software performs millions of movements in each simulated annealing run. It is prohibitively time-consuming to calculate the steady state temperature of the blocks for each movement since in order to calculate the steady state temperature, we need to construct a new thermal resistance matrix. We need therefore to find an approximate measure to represent the maximum temperature in the chip.

3.4 Maximum Temperature Estimation

An estimate for the maximum temperature should have the following properties:

1. Reflect the goodness of a floorplan with respect to the maximum temperature, i.e., a floorplan with a lower maximum temperature must have a lower value than a floorplan with a higher maximum temperature.

2. Should be easy to calculate.

The essence of temperature interaction between adjacent blocks is the *heat diffusion* between them. Thus, the heat diffusion between adjacent blocks can serve as a good approximation for the maximum temperature in the chip.

3.5 Heat Diffusion Measure

The heat diffusion between two adjacent blocks is proportional to their temperature difference and the length of the shared block boundary between them.

$H(T1, T2) = (T1 - T2) * shared\_length$

where H is the heat diffusion, $T1$ and $T2$ are the temperatures of the two blocks, and $shared\_length$ is the length of their shared boundary.

Since we do not know the exact temperatures of the blocks at each simulation step, we can not use them to calculate the heat diffusion directly, and we must replace them by estimates. To this end, consider an isolated block whose temperature can be calculated as:

$T = P * R = P * (t/k * A) = (P/A) * (t/k) = (t/k) * d$

where T is the steady state temperature, P is the power consumption, R is the thermal resistance between the block and the environment, t is the thickness of the chip, k is the thermal conductivity of the material, A is the area of the block, and d is its power density.

This expression shows that the temperature of an isolated block depends linearly on its power density. We will therefore, use the power density of a block as an estimate of its temperature.

Thus, we define the following measure as an approximation for the *heat diffusion* between two adjacent blocks:

$H(d1, d2) = (d1 - d2) * shared\_length$

where H is the heat diffusion, $d1$ and $d2$ are the power densities of the two blocks, and $shared\_length$ is the length of their shared boundary.

For each block, its total *heat diffusion* will be:

$H(d) = \Sigma H(d, di)$, for all its neighbors di.

We will calculate the heat diffusion of the chip as an approximation for the chip temperature, but in this calculation we do not consider all the blocks. If the power density of a block is very small, it is impossible for it to become the hottest block and as a result, its position is not important. We only need to care about the heat diffusion of blocks which may become the hottest block in the chip.

To select the *possibly-hot* blocks which may become the hottest blocks, we pick the top m ($1 \leqslant m \leqslant n$) blocks with the highest power density.

If we take too many blocks into consideration, the final result may not place the block with the highest power density and the block with the lowest power density next to each other. If we take too few blocks into consideration, e.g., only the block with the highest power density, then the other blocks with high power density can become the hottest block in the chip. It is important therefore to determine the number of *possibly-hot* blocks carefully.

We tried the selection of the top 1, 2, 3, or 4 blocks with the highest power density as *possibly-hot* blocks in our experiments. We found that the selection of 2 *possibly-hot* blocks produced the best results for the Alpha processor.

We calculate the heat diffusion H of all the selected *possibly-hot* blocks, and add them together. The total *thermal diffusion* D is defined as the sum of the heat diffusion of all *possibly-hot* blocks:

$D = \Sigma H(d)$, for all *possibly-hot* blocks

This D will be used as the approximation of the maximum temperature in our experiments.

Thus, the final objective function for the Parquet floorplanner is:

$Obj = C_A * A + C_W * W - C_D * D$

where C_A, C_W and C_D are the weights of the area, the wire length and the thermal diffusion, respectively. C_D has a negative sign because we want to maximize the *thermal diffusion* D.

In order to reduce the maximum temperature of the chip, we should surround blocks with high power density by blocks with low power density if possible. A block with high power density tends to have a higher temperature while a block with low power density tends to have a lower temperature. If we place them next to each other, we can get maximum heat diffusion between adjacent blocks and thus reduce the maximum temperature.

4. EXPERIMENTAL RESULTS

4.1 Parquet Generated Floorplans

Each run of Parquet takes about 5 seconds, and we ran this tool hundreds of times and then select the best result. For the Alpha floorplan, we only manipulate the positions of the blocks in the core area, while keeping that of the L2 cache fixed. Since the Parquet generated floorplan may have some unused space, the area of some blocks is increased to fill the unused space. As expected, increasing the area of the chip will decrease the power density of the blocks and affect the temperature of the chip. However, the increase in area that we have observed has been very small, usually less than 1%, so its impact is negligible. Also, we only increase the area of *non-possibly-hot* blocks, and never increase the area of *possibly-hot* blocks in order to keep the impact on the maximum temperature as small as possible.

In what follows we show several Parquet generated floorplans. **Low-temp** (Figure 10) is a floorplan obtained when we optimize

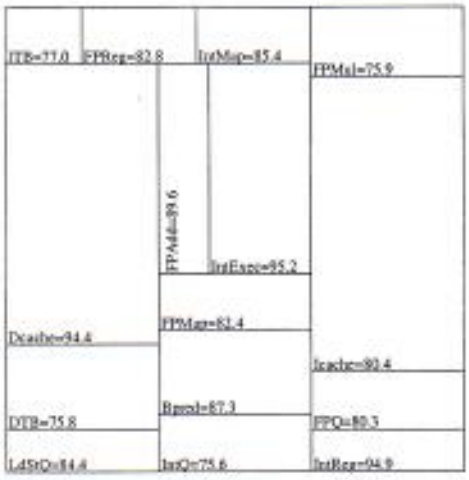

Figure 10: Low-temp: a floorplan with low maximum temperature

area and temperature ignoring wire length ($C_A = 0.4$, $C_W = 0$, $C_D = 0.6$). **Wire-temp** (Figure 12) is a floorplan which takes temperature, area and wire length into account ($C_A = 0.3$, $C_W = 0.4$, $C_D = 0.3$). **Short-wire** (Figure 14) is a floorplan which optimizes only the wire length and area but ignores the temperature ($C_A = 0.4$, $C_W = 0.6$, $C_D = 0$). **High-temp** (Figure 15) is a result of an attempt to generate a floorplan with the highest maximum temperature.

The maximum temperature of the **Low-temp** floorplan for the *gcc* benchmark is $95°C$, which is $25°C$ lower than that of the original floorplan. The steady state temperatures for SPEC2000 benchmark are shown in Figure 11. We can see that the average maximum temperature has decreased from $94°C$ to $81°C$, the average reduction of the maximum temperature is $13°C$, which is $2°C$ better than that obtained by the manually generated floorplan. The maximum temperatures for all the benchmarks have been reduced to below $97°C$. Although the temperatures of the *applu*, *lucas*, and *mgrid* benchmarks have increased by several degrees, their maximum temperatures are still below $80°C$.

The maximum temperature of the **Wire-temp** floorplan for the *gcc* benchmark is $99°C$, which is $21°C$ lower than the original one. The maximum temperatures for the SPEC2000 benchmarks are shown in Figure 13. We can see that the average maximum temperature has decreased from $94°C$ to $82°C$ with an average reduction of $12°C$. The maximum temperatures for all the benchmarks have been reduced to below $100°C$.

The maximum temperature of the **Short-wire** floorplan (see Figure 14) for the *gcc* benchmark is $120.1°C$, which is almost the same as that of the original flooeplan. This shows that in order to get a lower chip temperature, we must include temperature as an objective in Parquet.

If we do not include the temperature in the objective function, Parquet may generate floorplans with very high maximum temperatures. We show such a floorplan **High-temp** in Figure 15. This floorplan has a maximum temperature of $132°C$, which is $12°C$ larger than the original one and $37°C$ larger than the floorplan **Low-temp** with the lowest maximum temperature.

4.2 Area and Wire Length Overhead

The area increase, wire length overhead, and temperature reduction for all the generated floorplans are listed in Table 1. The area

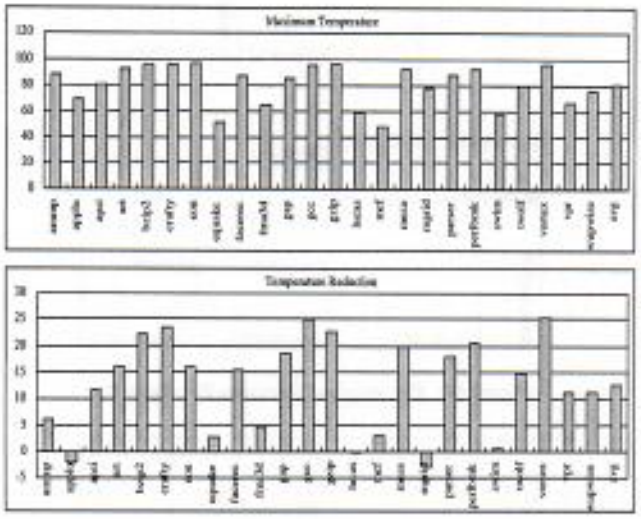

Figure 11: Maximum temperature and temperature reductions ($^{\circ}C$) of Low-temp floorplan for SPEC2000 benchmarks

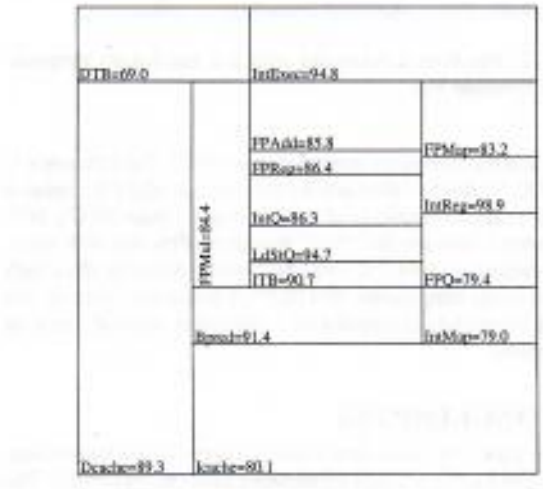

Figure 12: Wire-temp: a floorplan with both short wire length and low maximum temperature

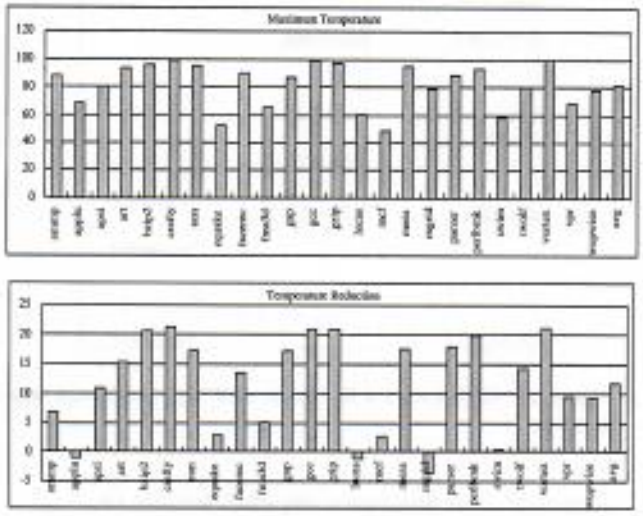

Figure 13: Maximum temperatures and temperature reductions ($^{\circ}C$) of Wire-temp floorplan for SPEC2000 benchmarks

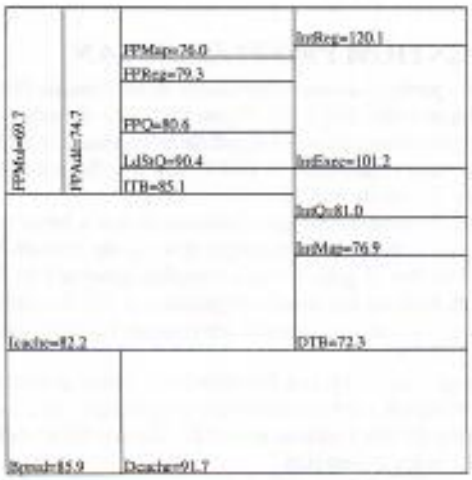

Figure 14: Short-wire: a floorplan with short wire length

Figure 15: High-temp: a floorplan with high maximum temperature

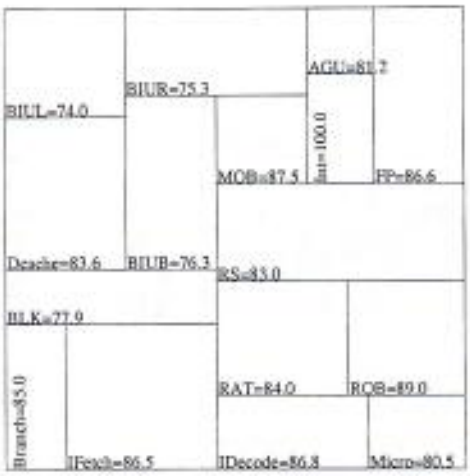

Figure 16: The original Pentium Pro floorplan

overhead of all the floorplans is very small, less than 1%.

The floorplan **Low-temp** reduces the maximum temperature by $25°C$ with a wire length overhead of 38%. If this wiring overhead is considered excessive, we can instead select the floorplan **Wire-temp** which reduces the maximum temperature by $21°C$ while keeping the wire length practically unchanged. The small reduction in wire length for the floorplan **Short-wire** shown in Table 1 does not necessarily mean that this floorplan is better than the original one with respect to wiring cost but can probably be attributed to the inaccuracies in the total wire length calculation.

Through careful block replacement, we succeed in greatly improving the temperature distribution of the chip while keeping the total wire length of the chip almost the same as before. Since many chips are facing the thermal problem, our temperature aware floorplanning can be very useful in alleviating this problem.

5. PENTIUM PRO FLOORPLAN

We also performed some experiments on the Pentium Pro processor ($0.35\mu m$ technology) [4]. Figure 16 shows the original floorplan for this microprocessor [3] and the temperature of each block. The maximum temperature is $100°C$ and it is the temperature of the integer execution unit *Int*.

We use the modified Parquet software to find a better floorplan with respect to the maximum temperature for the Pentium Pro processor. **Pro-low** (Figure 17) is a floorplan generated by Parquet. The block with the maximum temperature is still the integer execution unit *Int*, but the maximum temperature has been reduced by $6.3°C$ to $93.7°C$.

Pro-high (Figure 18) is a floorplan that we have generated in an attempt to obtain a higher maximum temperature. The maximum temperature for this floorplan is $110°C$, which is $10°C$ higher than that of the original floorplan.

The area of the floorplan **Pro-low** has increased by 1.5%, and its total wire length has increased by 13%. These are the penalties we pay for the improvement in the maximum temperature.

The benefits of modifying the block placement to improve the temperature for the Pentium Pro processor are not as impressive as those for the Alpha processor, because the temperature difference between blocks in the Pentium Pro chip is not as large. For the original Pentium Pro floorplan, the maximum temperature is

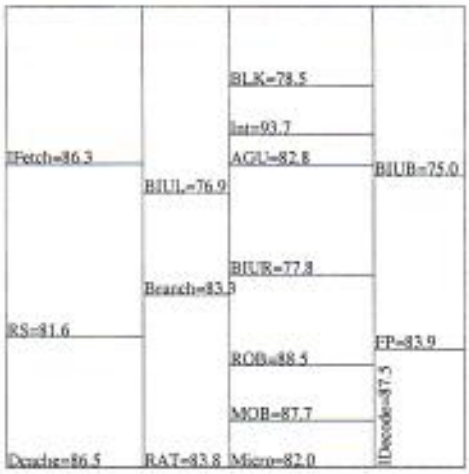

Figure 17: Pro-low: a floorplan with low maximum temperature for Pentium Pro

$100°C$, and the minimum temperature is $74°C$. The difference is only $26°C$, while the difference for the original Alpha floorplan is about $70°C$ (the temperature of the L2 cache is about $50°C$). Still, the difference between the "best" floorplan (**Pro-low** with maximum temperature of $93.7°C$) and the "worst" floorplan (**Pro-high** with maximum temperature of $110.5°C$) shows that even in this case it is worthwhile to consider the temperature when deciding on the floorplan.

6. CONCLUSIONS

In this paper, we have shown how to improve the temperature distribution of a chip through temperature aware floorplanning. Through experiments on the Alpha and Pentium Pro microprocessors, we have shown that we can obtain a temperature reduction of $21°C$ while keeping a comparable wire length for the Alpha processor, or a $6.3°C$ reduction in the maximum temperature for the Pentium Pro processor with a penalty of 13% in terms of the total wire length. In future designs based on deep sub-micron technology, chip temperatures are expected to further increase, making the benefits of temperature aware floorplanning even more prominent.

Table 1: Area, wire length and maximum temperature for Alpha floorplans

Floorplan	Area (mm^2)	Increase	Wire length (m)	Increase	Temp ($°C$)	Reduction ($°C$)
Original	253.1	0%	17.93	0%	120.0	0
Manual	253.1	0%	23.21	29.45%	98.2	21.8
Rotated	253.1	0%	18.32	2.18%	103.1	16.9
Low-temp	254.1	0.4%	24.77	38.15%	95.2	24.8
Wire-temp	255.1	0.8%	18.05	0.67%	98.9	21.1
Short-wire	256.3	0.9%	17.20	-4.07%	120.1	-0.1
High-temp	255.1	0.8%	19.07	6.36%	132.3	-12.3

Figure 18: Pro-high: a floorplan with high maximum temperature for Pentium Pro

7. REFERENCES

[1] http://lava.cs.virginia.edu/hotspot.

[2] http://www.dibeneditto.com/resources/20011105/.

[3] http://www.faculty.iu-bremen.de/birk/lectures/pc101-2003/02pentium/pentiumebpage/manufacturing.htm.

[4] http://www.sandpile.org/impl/p6.htm.

[5] http://www.spec.org/.

[6] Intel tries to keep its cool. *PC World*, april 2004.

[7] S. N. Adya and I. L. Markov. Fixed-outline floorplanning : Enabling hierarchical design. *IEEE Trans. on VLSI*, 11(6):1120–1135, December 2003.

[8] P. Chaparro, J. González, and A. González. Thermal-aware clustered microarchitectures. In *International Conference on Computer Design (ICCD)*, pages 48–53. ACM Press, 2004.

[9] C. C. N. Chu and D. F. Wong. A matrix synthesis approach to thermal placement. In *Proceedings of the 1997 international symposium on Physical design*, pages 163–168. ACM Press, 1997.

[10] J.-L. Cruz, A. González, M. Valero, and N. P. Topham. Multiple-banked register file architectures. In *Proceedings of the 27th annual international symposium on Computer architecture*, pages 316–325. ACM Press, 2000.

[11] J. Donald and M. Martonosi. Temperature-aware design issues for smt and cmp architectures. In *Proceedings of the Workshop on Complexity-Effective Design (WCED)*. ACM Press, 2004.

[12] K. Ghose and M. B. Kamble. Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation. In *Proceedings of the 1999 international symposium on Low power electronics and design*, pages 70–75. ACM Press, 1999.

[13] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam. Compact thermal modeling for temperature-aware design. In *Proceedings of the 41st annual conference on Design Automation*, pages 878–883, 2004.

[14] W.-L. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Thermal-aware IP virtualization and placement for networks-on-chip architecture. In *International Conference on Computer Design (ICCD)*, pages 430–437. ACM Press, 2004.

[15] W.-L. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Thermal-aware floorplanning using genetic algorithms. In *International Symposium on Quality Electronic Design (ISQED)*, 2005.

[16] T. Juan, J. J. Navarro, and O. Temam. Data caches for superscalar processors. In *Proceedings of the 11th international conference on Supercomputing*, pages 60–67. ACM Press, 1997.

[17] Y. Li, K. Skadron, Z. Hu, and D. Brooks. Evaluating the thermal efficiency of SMT and CMP architectures. In *IBM T. J. Watson Conference on Interaction between Architecture, Circuits, and Compilers*, October 2004.

[18] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th annual international symposium on Computer architecture*, pages 2–13. ACM Press, 2003.

[19] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, 2004.