Microarchitectural Floorplanning for Thermal Management: A Technical Report

†Karthik Sankaranarayanan, ‡Mircea R. Stan and †Kevin Skadron
†*Department of Computer Science,*
‡*Charles L. Brown Department of Electrical and Computer Engineering,*
University of Virginia, Charlottesville, VA
{*ks4kk,skadron*}*@cs.virginia.edu, mircea@virginia.edu*

Abstract

This paper presents research to address the temperature challenge in multicore processors through the lever of thermally-aware floorplanning. Specifically, it examines the thermal benefit in a variety of placement choices available in a multicore processor including alternative core orientation and insertion of L2 cache banks between cores as cooling buffers. In comparison with an idealized scheme that scatters the functional blocks of a multicore across the entire chip area to maximize uniformity, a combination of core orientation and L2 cache bank insertion achieves about 75% of the peak temperature reduction with negligible performance impact. On an average, the improvement in temperature is about 20% of the magnitude above the ambient temperature.

1 Introduction

As transistors scale into tens of nanometers, low-level physical effects which were previously considered second-order and were largely invisible to computer architects have surfaced to become primary concerns. Leakage, temperature, power delivery and parameter variations are a few examples. Of these, temperature has arguably become one of the hardest obstacles to continued technology scaling. The exponential increase in power density across technology generations translates into a corresponding increase in cooling costs in order to prevent it from resulting in higher temperature. The exponential impact of temperature on leakage power and lifetime reliability combined with usability considerations like fan noise and wearability have made high temperature very undesirable in microprocessors.

Early approach to the thermal management problem involved designing the thermal solution (heatsink, fan *etc.*) for the absolute worst-case application behaviour. This has later been complemented by circuit and microarchitectural techniques that adaptively trade-off the performance of applications to suit the thermal needs of the microprocessor. Such Dynamic Thermal Management (DTM) techniques (*e.g.* [12, 17, 3, 23, 30, 16] allow for the thermal solution to be designed for the average-case rather than the worst-case, thereby saving cooling costs. Circuit-based DTM techniques involve either the scaling of the voltage and frequency of the microprocessor or the stopping of the processor clock. Although effective in dealing with temperature, such alterations to the clock are undesirable in server environments as they lead to problems in clock synchronization and accurate time-keeping. Moreover, with non-ideal threshold voltage scaling, such an ability to reduce the voltage might not be easily available. Furthermore, microarchitectural DTM techniques that delay an application in response to a thermal overshoot are problematic in real time systems as they lead to unpredictable slowdowns and hence could

lead to applications missing their deadlines. Hence, there have been research efforts to examine microarchitectural thermal management schemes that do not compromise the latency of applications unpredictably.

Apart from controlling the level of computational activity of an application, another way to handle the thermal management problem is through better distribution of heat in *space*. In a multithreaded environment, this can be accomplished by the scheduling of threads on the hardware substrate in a thermally-aware fashion to distribute heat evenly across the hardware. With the advent of multicore and multithreaded processors, this approach has received research interest [27, 26, 6, 8]. However, orthogonal to both these dynamic methods of thermal management (performance trade-off and scheduling), a static technique to distribute heat spatially is thermally-aware floorplanning at the microarchitectural level. It is not only attractive because of its predictability (which is relevant for real time applications), but also for its ability to complement the dynamic schemes since it is orthogonal to them.

Thermally-aware microarchitectural floorplanning has been studied for single core processors [28, 14, 33, 5, 25]. However, multicore processors have become ubiquitous and they offer a spectrum of placement choices from the functional block level to the core level. Exploiting these choices for thermal benefit is the focus of this paper. Apart from Healy *et. al.*'s research [15] that occurred parallel to this work and Donald and Martonosi's paper [10] that tried out alternative placement strategies in the context of thermal efficiency of Simultaneous Multithreading (SMT) and Chip Multiprocessing (CMP) architectures, we are not aware of previous work that addressed thermally-aware multicore floorplanning at the microarchitectural level. Specifically, this paper makes the following contributions:

- It examines the thermal benefit in changing the relative orientation of cores in a homogeneous multicore chip so as to keep the hottest units of adjacent cores as far apart from each other as possible.
- Since second level caches have much lower computational activity than the cores, they are among the coolest units in a processor. Hence, this work studies the placement of L2 banks between adjacent cores so that they can function as cooling buffers that absorb the heat from the cores.
- As an academic exercise, it investigates the temperature reduction potential of multicore floorplanning by relaxing the requirements that functional blocks should stay within core boundaries and L2 cache banks should stay outside core boundaries.

The remainder of the paper is organized as follows: Section 2 describes previous work related to this paper. Section 3 explains our multicore floorplanning methodology. Section 4 presents the experimental results of our study and Section 5 concludes the paper providing direction to possible future work.

2 Related Work

The work that is most related to this paper is a parallel effort by Healy *et. al.* [15]. They also present a multicore floorplanner that optimizes the temperature profile of a microprocessor. They take a multi-granularity approach that considers both the within-core and across-core floorplans for thermal optimization. Their work employs a floorplanning algorithm that uses simulated annealing [22] based upon the sequence-pair representation [24] with a cost function incorporating both temperature and bus length. Although their work has the advantage of accounting for performance more accurately due to the explicit consideration of bus length, we believe that our approach is complementary to it. The main innovation in their paper is to floorplan an individual core in a multicore-aware fashion with the functional blocks of the core moved inwards from the periphery of the core so that when the cores form a mosaic in the multicore chip, the tiling does not result in the hot blocks being adjacent to each other. On the other hand, our approach achieves the same end through different means: by changing the orientation of the cores and by placing second level cache banks between them. Furthermore, although only an academic exercise,

we perform a limit study of the achievable thermal benefit in multicore floorplanning by a) letting the functional blocks from different cores be close to each other crossing core boundaries and b) inserting L2 cache banks in between the functional blocks of a core. We believe that this is also a significant point of distinction. Moreover, our paper also performs a sensitivity study on core size, core area as a fraction of chip area and L2 power density. Since these variables affect the thermal performance of a multicore floorplanner, it is important to consider them in the evaluation. Also, their work results in floorplans with dead spaces (which might be a consequence of using hard blocks). This is a costly inefficiency in the area *vs.* temperature trade-off since silicon real estate is expensive. Finally, their work employs multiple floorplans for the same microarchitecture, which could lead to a significant replication of design effort for each kind of floorplan used.

Another paper that has considered multicore floorplanning in a limited fashion (with its primary focus being other issues) is from Donald and Martonosi [10]. When studying the thermal efficiency of SMT and CMP architectures, they try an alternative layout strategy to reduce temperature by moving the two cores of a CMP apart, from the center of the chip to its edge. This is similar to the use of L2 cache banks in our work as cooling buffers. However, they approach it as a one-off technique without any generic extensions or applicability to arbitrary floorplans. Since temperature is a serious concern for 3-D architectures, thermally-aware floorplanning for 3-D chips is relevant to our work [11, 20]. Similarly, the wealth of work in thermally-aware placement for ASICs and SoCs(*e.g.* [9, 7, 19, 13]) and microarchitectural floorplanning for thermal optimization [28, 14, 33, 25, 5] is also related to this work. However, none of these papers study the placement choices in multicore architectures.

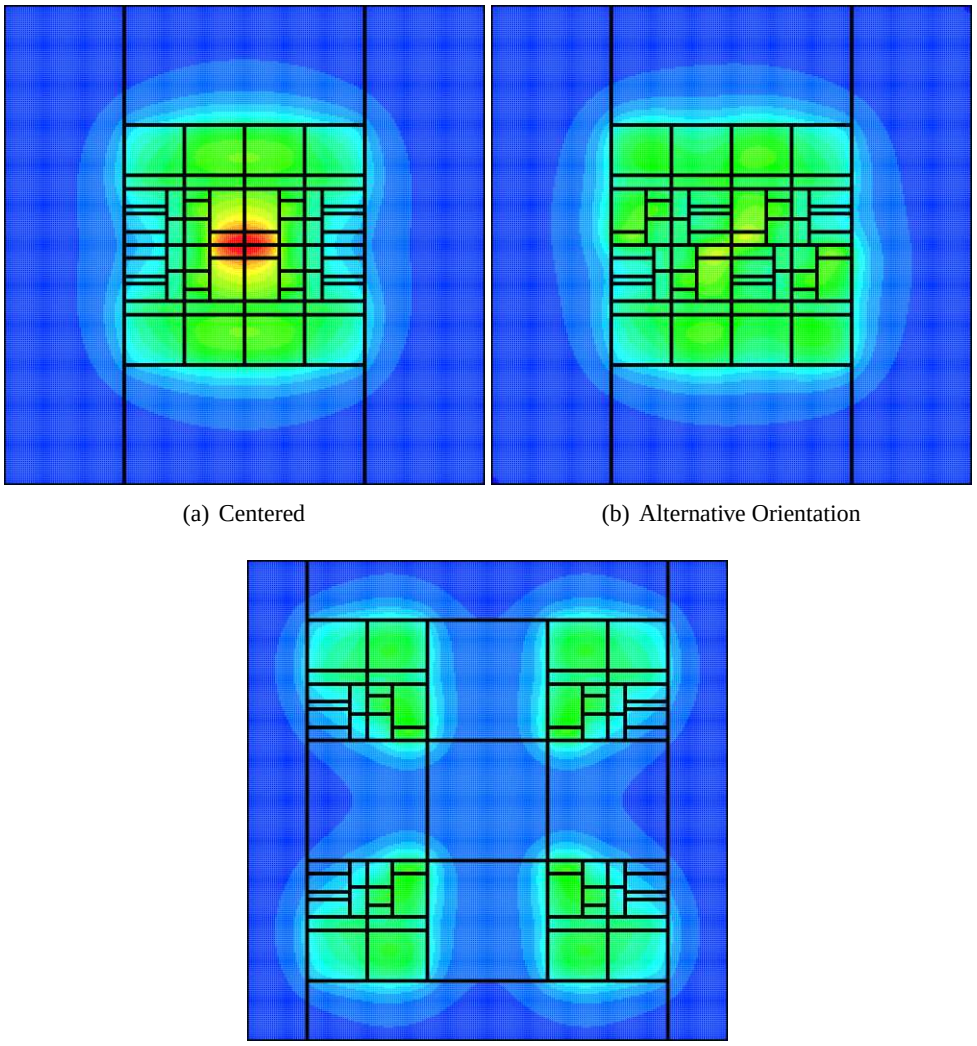
3 Methodology

As multicore architectures have become the norm today, there are many levels of placement choices available for a designer—from the functional block level to the core level. These choices can be exploited to spatially distribute heat effectively. Specifically, following are some of the possibilities we consider for a homogeneous CMP:

3.1 Core Orientation

The advantage of having identical cores in a CMP is physical design re-use. A single core can be designed once and re-used many times. In such homogeneous cores seen today, the orientation of the cores is such that their floorplans are mirror images of each other. This typically leads to hot functional blocks being adjacent to each other and oriented towards the center of the core. Figure 1(a) illustrates the thermal profile of such an arrangement for a homogeneous 4-way CMP with each core resembling that of an Alpha 21364 as in [30]. Without compromising the re-usability of the cores, a simple temperature-aware floorplanning scheme would be to experiment with the orientation of the cores for temperature reduction. Fig 1(b) illustrates such a floorplan with alternative orientations of the cores that result in reduced peak temperature. Specifically, the cores on the bottom right and top left are flipped about the vertical axes passing through their respective centers.

Each core can have eight different orientations. They are the four rotational symmetries and their corresponding four reflections (mirror images). Hence, for a k-way CMP, there are a total of 8^k different possible floorplans. For a small number of cores (*e.g.* $k \leq 6$), this is still within the limits of a brute-force search. Given a set of representative power numbers (which can be obtained through application profiling), we use the HotSpot [30] thermal model to obtain the corresponding thermal profile and choose the floorplan which offers the lowest peak temperature. Once the number of cores crosses the limit of a brute-force search, we employ simulated annealing [22] to search through the vast solution space. Each floorplan is encoded as a k-digit octal number denoting the set of core orientations it is comprised of. The only type of move used in the annealing schedule is a random increment move where a random digit is chosen from the k-digit string and incremented (modulo 8) to the next numerical orientation.



(a) Centered (b) Alternative Orientation

(c) Checkerboard-like

Figure 1. Illustration of different core arrangements for a 4-way CMP with each core resembling an Alpha 21364. (a) shows a typical floorplan with hot units adjacent to each other. (b) shows a floorplan with alternative orientations of the cores. (c) shows a checkerboard-like arrangement with the use of L2 cache banks as cooling buffers between the cores.

3.2 L2 Cache Banks

As second level caches are large, they are already partitioned into many banks. Furthermore, their power density is quite low because of relatively infrequent accesses. Hence, their temperatures are usually among the lowest in a chip. So, a possible strategy for temperature reduction is to use the L2 banks as cooling buffers between cores. However, in doing so, the performance cost of a longer L2 bus must be accounted for. Since we assume a traditional L2 cache with Uniform Cache Access (UCA), the L2 latency already includes the worst-case latency from a core to the farthest L2 bank. Thus, in placing the cache banks between cores, the latency increase is only proportional to the maximum extra distance a core moves within the chip due to the cache bank insertion. For the floorplan configurations considered in this paper, a part of the L2 always wraps around the periphery of the chip. In such

a scenario, for the range of L2-area to chip-area ratios we consider (25-85%), a core can move an extra distance between 7 and 44%. Assuming a linear wire delay model similar to [28], this implies the same percentage increase in L2 latency too. Since L2 latency is tolerated well by the microarchitecture due to the presence of L1 caches that filter out most of the accesses, this translates to less than 2% slowdown for SPEC2000 benchmarks [28, 5]. Hence, in this paper, we consider the performance impact of distributing the L2 cache banks between the cores to be negligible. However, it is to be noted that our simulation setup involves running identical benchmarks on all the cores with no communication between them. Although this modeling methodology is a limitation of this work, we believe it is not a serious one because of two reasons. First, in comparison with an arrangement of the cores adjacent to each other, the cache insertion provides extra space for routing in the vicinity of the cores (over the sub-arrays of the cache). This space could be used to reduce the latency of the interconnection network by using thicker wires, thus minimizing the impact of the latency on coherence traffic. Second, for a large number of cores, Non-Uniform Cache Access (NUCA) is the likely choice and since it already envisions an interconnection network with a distributed arrangement of the cores and the cache banks [21], the performance of a cache-bank inserted layout as suggested in this work is not likely to be much different.

In exploring the placement of the cache banks, we first assume that their size and aspect ratio are flexible. Then, the processor cores and L2 blocks could be arranged to tile the entire silicon real estate in a checkerboard-like fashion to increase the lateral spreading of heat. Since silicon acts as a spatial low-pass filter for temperature [18], maximizing the spatial frequency of the power density distribution is beneficial for temperature reduction. For a checkerboard-like tiling of a multicore chip, this is accomplished by making the high power areas (cores) as small as possible (by separating the cores from one another) and the low power areas between them (caches) as large as possible. Furthermore, since the chip boundaries allow lateral heat conduction only in two or three directions (instead of four), this also means that the cores should be placed away from the chip boundaries to facilitate heat spreading. A sample of such an arrangement is shown in Figure 1(c). Please note that although we use the term *checkerboard-like*, the cache bank insertion is in essence seperating the cores in *both* the x and y directions with cache banks. In Figure 1(c), a true checkerboard arrangement would have had another core in the middle square. However, in this paper, we use the term *checkerboard-like* for the lack of a better alternative. Also, for determining the positions of the cores and the cache banks, we assume (heuristically) that adjacent cores are equidistant from each other and that the distance between a peripheral core and chip boundary is half the distance between adjacent cores.

3.3 Hierarchical Floorplanning

Since the checkerboard-like arrangement separates the cores from one another, it reduces the lateral thermal coupling between them. This affords us a possibility of floorplanning the functional blocks of the cores independent of their multicore arrangement. Hence, we apply a hierarchical floorplanning algorithm combining a previously proposed single-core floorplanner [28] with both of the above techniques (orientation and tiling). Given a set of functional blocks and their area and aspect ratio constraints, it first floorplans the core using the classic Wong and Liu [32] simulated annealing algorithm with a cost function that includes area, delay and temperature. In assigning the relative importance to architectural wires, we use the modifications suggested by [5] instead of the order originally proposed in [28]. This single-core floorplan is then arranged in a checkerboard-like fashion with L2 cache banks arranged in between the cores as described in Section 3.2. Then, as a final step, the orientation space of the cores is searched using simulated annealing as described in Section 3.1.

3.4 Potential Study

Finally, as an academic exercise, we also investigate a floorplanning strategy that allows for complete flexibility in the placement of functional blocks even disregarding the core boundaries. Since this compromises design

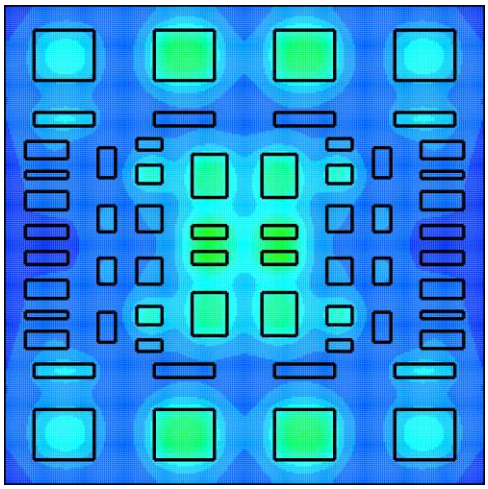


Figure 2. Thermal profile of a floorplan with scattered functional blocks in a 4-way CMP with each core resembling an Alpha 21364. The adjacency of the blocks in the original floorplan is maintained through the scattering.

re-use and performance at all levels, it is not presented here as a practical technique. Such an experiment is useful just as a potential study to measure against the performance of the other techniques mentioned above. We look at two possibilities: in the first, we apply the single core floorplanning algorithm in [28] to a combined list of all the functional blocks in the entire multicore chip. This basically results in a medley of functional blocks from different cores occupying the center of the multicore chip and surrounded by the L2 cache banks. Compared against the alternative core orientation strategy mentioned in Section 3.1, it tells us how much thermal potential is available in distributing the functional blocks within the cores. The second possibility we look at is to insert the L2 cache banks even between the functional blocks within a core. This results in the scattering of the functional blocks through out the entire chip. In the process of scattering, the adjacency of the functional blocks is retained as it was in the original core floorplan. This is illustrated in Figure 2. It is useful to compare the figure against the non-scattered version in Figure 1(a) and the core-level cache inserted version in Figure 1(b). Such a scattering serves as a benchmark for the L2 cache insertion technique described in Section 3.2 to measure against. However, the performance of such a scattered floorplan is likely to be significantly worse than a floorplan in which closely communicating functional units are adjacent to each other. This is especially true if the delay on the architectural wires involved in critical loops increases due to the scattering [2]. Hence, we only consider this scheme to understand its potential for temperature reduction and not as a realistic approach.

3.5 Experimental Setup

In order to evaluate the floorplanning choices described above, we use a simulation infrastructure comprised of the HotSpot [30] thermal model, Wattch [4] power model and Simplescalar [1] performance model. We use the SPEC2000 [31] benchmark suite and run each benchmark for an interval of 500 Million instructions using the reference inputs. The interval that is most representative of the entire program is identified using the SimPoint [29] tool. We extend the HotFloorplan tool from [28] to implement the multicore floorplanning strategies described in Section 3. The floorplanner is fed with the floorplan of a single core and a representative set of power values for each of its functional blocks (which we compute as the average of the power values of all the benchmarks simulated as mentioned above). It uses this information and searches through the solution space to find a floorplan

configuration that is thermally sound. In doing so, it uses the block-based thermal model of HotSpot to compute the temperature of the configurations at every step and chooses the floorplan with the lowest peak temperature. The block-based model is chosen because of its computational speed. However, in evaluating these floorplans, we employ the regular grid-based thermal model of HotSpot which is slower but more accurate. We use a grid resolution of 256 x 256 and perform steady state thermal simulations to obtain the peak temperature of the different floorplan configurations.

Although we model the dependence of leakage power on temperature using the empirical model in [30], this is done only during the computation of the input power densities of each benchmark. Once the peak steady state temperature of a benchmark for a given floorplan is computed, the power densities are not re-evaluated by taking into account the reduced leakage due to temperature reduction. Hence, the temperature improvement because of floorplanning reported here does not include the benefit accrued from reduced leakage. Thus, it really forms a lower bound to what can be expected in practice and the actual enhancement is likely to be higher.

We model a microarchitecture similar to the Alpha 21364 as in [30] but scaled to 90 nm for the single core case. In order to model a multicore configuration, we scale both the area of each core and its power consumption such that the power density remains constant. The thermal model parameters are set to the default values of HotSpot except the convection resistance and TIM thickness, which are assigned values of 0.5 $\frac{K}{W}$ and 30μm respectively in order to model a moderate cooling solution. The default configuration modeled is a 4-core processor with 75% of the die-area occupied by L2 cache. To reduce simulation complexity, each core is assumed to run the same benchmark.

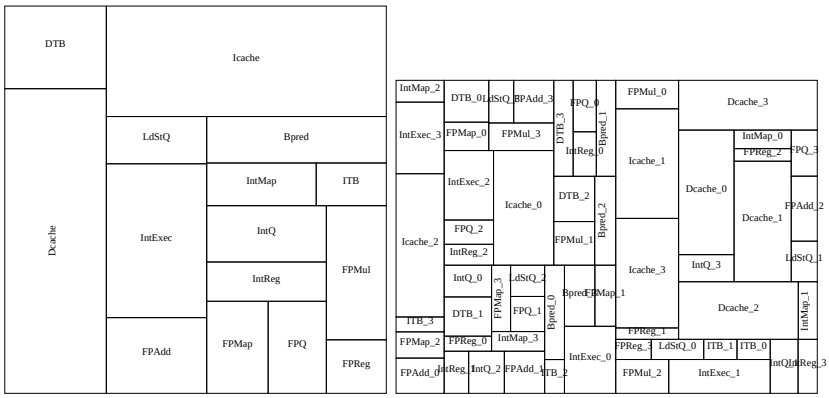
4 Results

We will now describe the various placement choices evaluated. The first configuration is with the four cores arranged at the center of the chip wrapped around by the L2 cache. The cores are oriented in such a manner that their hottest units, the integer register files, are touching each other. This is similar to the illustration in Figure 1(a). Such a configuration is chosen to simulate the worst-case behaviour. We call this arrangement the *hot* scheme. Next is the configuration that forms the base-case of our evaluation. It is similar to *hot* in that the cores are arranged at the center of the chip but the orientation of all the cores is the same — pointing upwards. We call this the *base* scenario.

The next floorplan evaluated is the result of searching the orientation space as described in Section 3.1. For four cores, a brute-force search is performed. The floorplan with the lowest peak temperature is considered for evaluation. This scheme is called *orient*. Next, the cache bank insertion described in Section 3.2 is performed over the *base* configuration. This is called *base+l2*. When the same is done on top of the *orient* scheme, it is called *orient+l2*.

In order to perform the first of the two limit studies described in Section 3.4, we scatter the blocks in the *base* configuration in between the L2 cache banks as shown in Figure 2. This scheme is called *base+scatter*. Such a scatter performed for the *orient* configuration is called *orient+scatter*.

In order to evaluate the hierarchical floorplanning algorithm described in Section 3.3, we first use the *base* configuration with the floorplan of each core derived from the single core floorplanner in [28]. Apart from assigning the relative weights of the architectural wires as per [5], we also incorporate the core aspect ratio into the cost function of simulated annealing. This results in a floorplan with less than 0.05% dead space and a better wire length metric when compared to the base-case Alpha 21364-like floorplan. This alternative floorplan is shown in Figure 3(a). Its aspect ratio is close to 1. We call the 4-way multicore scenario obtained by replicating this alternative floorplan as *altflp*. It is to be noted that the alternative floorplan is computed in isolation, without being mindful of the core's location in a multicore processor. Hence, hot units are steered away from the boundaries as much as possible to minimize the peak temperature. This is not necessarily beneficial in a multicore environment — especially with the L2 cache bank insertion because units near the boundaries of a core are closer to the L2



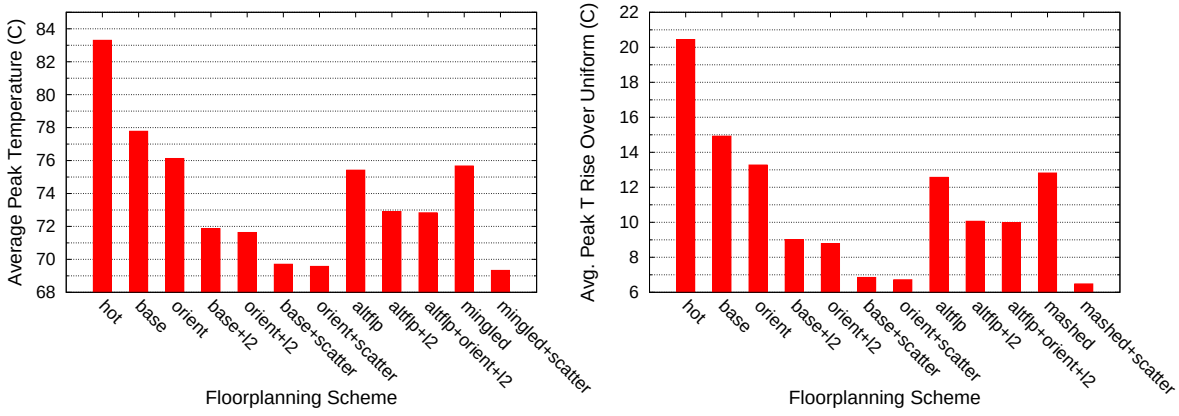
(a) Alternative (b) Mingled

Figure 3. Floorplans used for the *altflp* and *mingled* studies

cache, which is relatively cooler.

Retaining the nomenclature above, we call the cache bank inserted version of *altflp* as *altflp+l2* and an orientation space search for *altflp+l2* as *altflp+orient+l2*.

Finally, we examine the potential of mingling the functional blocks from different cores as described in Section 3.4. The result of the simulated annealing performed on a single list of functional blocks from all the cores is shown in Figure 3(b). The amount of dead space for this floorplan is less than 0.1% of the total area of the cores. All the blocks are assumed to be soft with the constraints on their aspect ratio specified in the input file. Hence, the same functional blocks (*e.g.* L1 data cache) from different cores can have different aspect ratios. This scheme is called *mingled*. We also employ the insertion of cache banks in between the functional blocks of the *mingled* scheme. This is called *mingled+scatter*.



(a) Peak Temperature (b) Rise Over Uniform

Figure 4. The peak temperature of the different floorplan configurations averaged across all the SPEC2000 benchmarks. (a) shows the peak temperature while (b) shows the peak rise in temperature in comparison with a uniform power distribution

Figure 4 shows the results of our evaluation. It shows two graphs. Figure 4(a) plots the peak temperature of

each floorplan scheme described above (and listed on the x-axis of the figure), averaged over all the 26 SPEC2000 benchmarks. Since we are also interested in the exploration of limits, Figure 4(b) plots the same data differently. Given the total power consumption on a die, the absolute maximum thermal benefit achievable due to floorplanning is bounded by the case where all that power is dissipated uniformly across the entire die. Figure 4(a) shows the difference between the peak temperature of the floorplanning schemes and the peak temperature of such a uniform power distribution. The data shown is the average across all benchmarks. Such a *delta-T* metric (rise in temperature over that of uniform power distribution) expresses the thermal proximity of a floorplanning scheme to the maximum possible thermal benefit.

Clearly, from the figure, the *hot* configuration with the four hottest blocks adjacent to each other has the highest average peak temperature. Exploiting the orientation of the cores is beneficial when the cores are adjacent to one another as can be seen from the difference between the *base* and *orient* bars. However, when the cores are already separated by the L2 cache banks, the orientation of the cores matters to a much lesser degree. This is the reason the *base+l2* and *orient+l2* bars are pretty similar to each other.

The insertion of L2 cache banks between the cores reduces peak temperatures significantly. There is a 6.1 degree difference between the *base* and *orient+l2* bars, with a large portion of it coming from L2 insertion. For the 11 hottest benchmarks, this improvement is about 8.3 degrees on an average. For an ambient temperature of $45^{\circ}C$, this translates to about a 20.2% improvement over the temperature in excess of the ambient. The *base+scatter*, *orient+scatter* and *mingled+scatter* bars indicate the thermal spreading potential available in multicore floorplanning. Comparing the *orient+l2* bar against these, we can see that a combination of core orientation and L2 cache insertion is able to achieve a significant portion of that potential (about three-fourths).

It can also be seen that although the alternative floorplan and the mingled floorplan are able to achieve temperature reduction, much of that reduction can be achieved by a simple orientation space search (*orient*). Furthermore, since the alternative floorplan has the hot functional blocks towards its center, the use of L2 cache banks as cooling buffers does not benefit it as much as it does the default floorplan. This is the reason *altflp+l2* and *altflp+orient* bars are higher than *base+l2* and *orient+l2*.

4.1 Sensitivity Studies

In this section, we investigate how the conclusions of the previous section are affected by our assumptions about the core size, occupancy and L2 power density respectively. This is done through sensitivity studies that vary the above-mentioned parameters.

4.1.1 Effect of Core Size

Figure 5 plots the effect of varying the size of each core (and hence the total number of cores in a chip). It plots the *delta-T* metric mentioned above for the practical (non-ideal) schemes from the previous section against the number of cores. It is to be noted that the power density of the functional blocks is maintained in changing the size of the cores. The lines shown are decreasing because silicon acts as a spatial low-pass filter for temperature [18]. Hence, for the same power density, smaller cores (high frequency) are cooler than larger cores (low frequency). It can be noted that the trends observed in the previous section still hold. Much of the thermal benefit comes from L2 cache insertion. The only reversal in trend is that *altflp* performs even worse than *orient* for higher number of cores.

4.1.2 Effect of Core Occupancy

Another important parameter in our study is the ratio of core area to the total area of the chip. We call this ratio the core occupancy. Figure 6 plots the result of an experiment varying the core occupancy from 15% to 75%. Actually, two competing factors determine the temperature in this experiment. First is that as the core occupancy decreases,

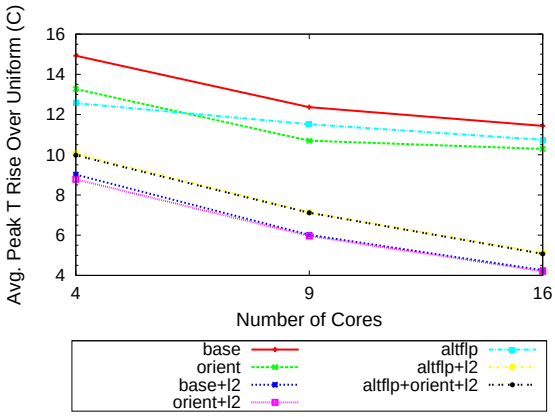


Figure 5. Effect of varying the number of cores. In scaling the cores, the power density of the functional blocks is kept constant.

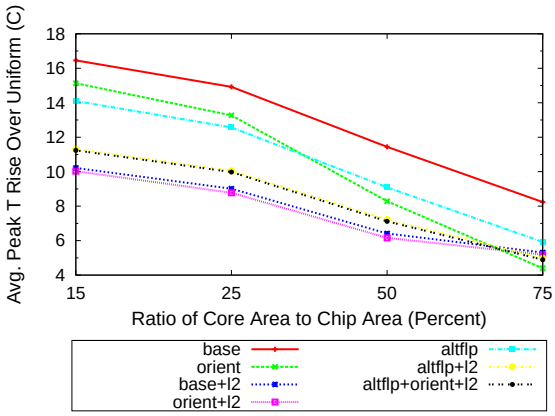


Figure 6. Impact of the ratio of core area to chip area. The size of each core remains the same while that of the L2 cache is increased, keeping its power density constant.

in order to keep the core and L2 power densities constant for an apples-to-apples comparison, the total chip area increases. Hence, the total power dissipated also increases with decreasing occupancy. The second factor is the reduced availability of the relatively cooler L2 space to act as a thermal buffer as occupancy increases. Depending on which factor predominates, sections of the curves decrease or increase. It is evident from the graph that as the occupancy increases, the importance of core orientation increases. This is the reason the *orient* line decreases quickly. At 75% core occupancy, it even performs marginally better than the L2 cache insertion techniques.

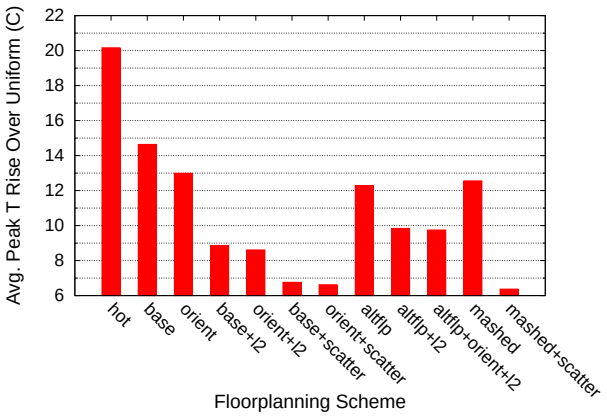


Figure 7. Thermal performance of the floorplanning schemes on doubling the L2 cache power density.

4.1.3 Effect of L2 Power Density

Since the floorplanning schemes presented in this paper involve the L2 cache banks, the power density of L2 cache is an important factor to be considered. Hence, we perform an experiment replicating the results in Figure 4 with the power density of the L2 cache being double of what it was in that figure. The results of this are presented in Figure 7. Clearly, the trends remain the same as before. The actual peak temperatures are slightly higher than those in Figure 4 (by less than two degrees on average) for all the floorplanning schemes in this experiment (including the baseline) due to the increased power density.

5 Conclusion

This paper investigated the temperature reduction potential of multicore floorplanning. It advocated the exploitation of various placement choices available in a multicore processor ranging from the functional block level to the core level. It proposed the exploration of alternative core orientations in order to separate hot units from being adjacent to each other in a multicore chip. It also presented the idea of inserting L2 cache banks between the cores as cooling buffers for better heat distribution. Furthermore, it studied the potential of multicore floorplanning by letting functional blocks and L2 cache banks cross core boundaries. The most important conclusion from this work is that L2 bank insertion achieves significant thermal benefit — about 20% of the temperature above the ambient on an average for SPEC2000 benchmarks. Furthermore, a combination of core orientation and L2 bank insertion is able to achieve about three-fourths of the temperature reduction achievable by an ideal floorplanning scheme that mingles functional blocks from multiple cores and disperses them amidst a sea of L2 cache banks.

With the advent of SIMD processors including GPUs, future work in this direction could examine the applicability of floorplanning techniques in reducing their peak temperature. Furthermore, since heterogeneous multicore architectures offer additional levels of placement options, exploiting them for thermal benefit is another interesting possibility.

References

[1] T. Austin, E. Larson, and D. Ernst. Simplescalar: An infrastructure for computer system modeling. *IEEE Computer*, 35(4):59–67, Feb. 2002.

[2] E. Borch, E. Tune, S. Manne, and J. Emer. Loose loops sink chips. In *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*, pages 299–310, 2002.

[3] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the Seventh IEEE International Symposium on High-Performance Computer Architecture*, pages 171–82, Jan. 2001.

[4] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual ACM/IEEE International Symposium on Computer Architecture*, pages 83–94, June 2000.

[5] A. Chakravorty, A. Ranjan, and R. Balasubramonian. Re-visiting the performance impact of microarchitectural floorplanning. In *Third Workshop on Temperature-Aware Computer Systems(TACS-3), held in conjunction with ISCA-33*, June 2006.

[6] P. Chaparro, J. González, and A. González. Thermal-aware clustered microarchitectures. In *Proceedings of the 2004 IEEE International Conference on Computer Design*, Oct. 2004.

[7] G. Chen and S. Sapatnekar. Partition-driven standard cell thermal placement. In *ISPD '03: Proceedings of the 2003 international symposium on Physical design*, pages 75–80, 2003.

[8] J. Choi, C.-Y. Cher, H. Franke, H. Hamann, A. Weger, and P. Bose. Thermal-aware task scheduling at the system software level. In *ISLPED '07: Proceedings of the 2007 international symposium on Low power electronics and design*, pages 213–218, 2007.

[9] C. N. Chu and D. F. Wong. A matrix synthesis approach to thermal placement. In *ISPD '97: Proceedings of the 1997 international symposium on Physical design*, pages 163–168, 1997.

[10] J. Donald and M. Martonosi. Temperature-aware design issues for smt and cmp architectures. In *Proceedings of the 2004 Workshop on Complexity-Effective Design*, June 2004.

[11] M. Ekpanyapong, M. B. Healy, C. S. Ballapuram, S. K. Lim, H. S. Lee, and G. H. Loh. Thermal-aware 3d microarchitectural floorplanning. Technical Report GIT-CERCS-04-37, Georgia Institute of Technology Center for Experimental Research in Computer Systems, 2004.

[12] S. Gunther, F. Binns, D. M. Carmean, and J. C. Hall. Managing the impact of increasing microprocessor power consumption. In *Intel Technology Journal*, Q1 2001.

[13] A. Gupta, N. Dutt, F. Kurdahi, K. Khouri, and M. Abadir. Stefal: A system level temperature- and floorplan-aware leakage power estimator for socs. In *VLSID '07: Proceedings of the 20th International Conference on VLSI Design*, pages 559–564, 2007.

[14] Y. Han, I. Koren, and C. A. Moritz. Temperature aware floorplanning. In *Second Workshop on Temperature-Aware Computer Systems(TACS-2), held in conjunction with ISCA-32*, June 2005.

[15] M. B. Healy, H.-H. S. Lee, G. H. Loh, and S. K. Lim. Thermal optimization in multi-granularity multi-core floorplanning. In *ASP-DAC '09: Proceedings of the 2009 Conference on Asia and South Pacific Design Automation*, pages 43–48, 2009.

[16] S. Heo, K. Barr, and K. Asanovic. Reducing power density through activity migration. In *Proceedings of the 2003 ACM/IEEE International Symposium on Low Power Electronics and Design*, Aug. 2003.

[17] W. Huang, J. Renau, S.-M. Yoo, and J. Torellas. A framework for dynamic energy efficiency and temperature management. In *Proceedings of the 33rd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 202–13, Dec. 2000.

[18] W. Huang, M. R. Stan, K. Sankaranarayanan, R. J. Ribando, and K. Skadron. Many-core design from a thermal perspective. In *DAC '08: Proceedings of the 45th annual conference on Design automation*, pages 746–749, June 2008.

[19] W. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T.Theocharides, and M. J. Irwin. Thermal-aware floorplanning using genetic algorithms. In *Sixth International Symposium on Quality of Electronic Design (ISQED'05)*, March 2005.

[20] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Interconnect and thermal-aware floorplanning for 3d microprocessors. In *ISQED '06: Proceedings of the 7th International Symposium on Quality Electronic Design*, pages 98–104, 2006.

[21] C. Kim, D. Burger, and S. Keckler. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 211–222, 2002.

[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[23] C.-H. Lim, W. Daasch, and G. Cai. A thermal-aware superscalar microprocessor. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 517–22, Mar. 2002.

[24] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. Vlsi module placement based on rectangle-packing by the sequence-pair. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15(12):1518–1524, 1996.

[25] V. Nookala, D. J. Lilja, and S. S. Sapatnekar. Temperature-aware floorplanning of microarchitecture blocks with ipc-power dependence modeling and transient analysis. In *ISLPED '06: Proceedings of the 2006 international symposium on Low power electronics and design*, pages 298–303, 2006.

[26] M. D. Powell, M. Gomaa, and T. N. Vijaykumar. Heat-and-run: Leveraging SMT and CMP to manage power density through the operating system. In *Proceedings of the Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2004.

[27] E. Rohou and M. Smith. Dynamically managing processor temperature and power. In *Proceedings of the 2nd Workshop on Feedback-Directed Optimization*, Nov. 1999.

[28] K. Sankaranarayanan, S. Velusamy, M. R. Stan, and K. Skadron. A case for thermal-aware floorplanning at the microarchitectural level. *Journal of Instruction-Level Parallelism*, 7, 2005. (`http://www.jilp.org/vol7`).

[29] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder. Automatically characterizing large scale program behavior. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 45–57, October 2002.

[30] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th Annual ACM/IEEE International Symposium on Computer Architecture*, pages 2–13, Apr. 2003.

[31] Standard Performance Evaluation Corporation. SPEC CPU2000 Benchmarks. http://www.specbench.org/osg/cpu2000.

[32] D. F. Wong and D. L. Liu. A new algorithm for floorplan design. In *Proccedings of the ACM/IEEE 23rd Design Automation Conference*, pages 101–107, June 1986.

[33] Y.-W. Wu, C.-L. Yang, P.-H. Yuh, and Y.-W. Chang. Joint exploration of architectural and physical design spaces with thermal consideration. In *ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design*, pages 123–126, August 2005.