Enabling Power Density and Thermal-Aware Floorplanning

Ehsan K. Ardestani, Amirkoushyar Ziabari, Ali Shakouri, and Jose Renau
School of Engineering
University of California Santa Cruz
1156 High Street
Santa Cruz, CA, 95064
{eka, aziabari, ali, renau}@soe.ucsc.edu

***Abstract*—With temperature being one of the main limiting factors in design of high performance processors, early evaluation of thermal effects in design stages is becoming a necessity. Floorplanning is an imperative step in the design process where thermal effects can be taken into account. This work studies a thermal-aware floorplanning scheme, with the goal of increasing both reliability and performance measures of the design. We show that a majority of thermal emergencies can be averted by a) leveraging the lateral heat transfer effects (as has been shown previously), and b) by reducing the power density of thermally critical blocks. The former becomes possible through moving, and modifying the aspect-ratio of the blocks in the floorplanning process. The latter, one of the key contributions of this work, is carried out through resizing of functional blocks in a controlled way. We also propose a selective power map generation method for the floorplanning process. In this method the time windows in which thermal emergencies occur guide the power map generation. As a result, we observed an 8.8% performance improvement, and a 40% reliability increase with the area overhead of just 3%.**

***Index Terms*—Floorplanning, Power Blurring, thermal simulation, architectural level thermal simulator.**

I. Introduction

Floorplanning is an essential component of any successful integrated circuit design, particularly in the design of high performance processors. Today, we are experiencing a shift in the problem formation from focusing primarily on area utilization and timing, to one where design objectives, such as power, temperature and reliability are major concerns. The thermal characteristics affect multiple aspects of processor design including frequency, leakage, performance throttling and cooling cost. In this work, we focus on the thermal impact of floorplanning on performance and reliability measures in an integrated circuit (IC).

Thermal effects on an integrated circuit impact both reliability and performance. Processors with higher thermal cycles have shorter mean time to failures (MTTF) [1]. Also, high thermal gradients across the chip could adversely impact wire delay [2], and narrow the frequency margin. In addition, all modern processors are equipped with dynamic thermal management (DTM) technique to keep temperature in the safe range and prevent catastrophic break down of the integrated circuit. At high temperatures, DTM triggers global or local actions to preemptively lower the power consumption. This typically comes at the cost of performance. Hence, a thermal-aware floorplan could potentially improve both reliability and performance of a processor.

With the shrinking trend in very large scale integration (VLSI) circuits, the ratio of leakage to total power keeps increasing. As a result leakage power is gaining more attention as a first order design parameter. Leakage power is temperature dependent, and reducing the temperature across the chip results in less leakage.

The lateral heat spreading and interaction among adjacent functional blocks impacts the hotspot formation on the chip. To either balance the thermal distribution on the chip, or to reduce the hotspots and peak temperature, thermal aware floorplanning methods have been proposed [3]–[5]. The proposed methods run through iterations of floorplan assignments, and thermal evaluation of the processor, with the goal of finding an optimum floorplan in terms of area, timing and thermal profile.

Not all hotspots can be reduced to safe levels by leveraging the lateral heat spreading effect. Thermal behavior of blocks is highly correlated with power density, and in thermal-aware floorplanning, power density is a simple metric that can be used to guide the exploration of design space. Our technique allows for the resizing of individual blocks in the floorplan to address issues related to high power density blocks. This is an issue that cannot be rectified by simply re-placing individual blocks.

In this work, we aim at reducing the number of thermal emergencies in the processor through floorplanning. Through a simulated annealing based method, we study the floorplanning impact on performance and reliability of the chip. Thermal throttling is a DTM method that we implement. We show how a resulting floorplan improves processor performance by lowering the amount of time it spends in thermal throttling. We also consider a set of temperature dependent reliability metrics [1] to relatively characterize the quality of each floorplan.

We use an integrated performance, power and temperature toolchain to study the impact of thermal-aware floorplanning on reliability and performance. Previous works do not quantify the reliability of the chip in terms of mean time to failure, nor show the performance impact of the improved floorplan due to thermal effects. This paper has the following contributions:

- It proposes a selective method of generating representative power value for floorplanning,
- Allows for the resizing of functional blocks to add power density exploration capability,
- It quantifies the reliability improvement achieved by the floorplanning scheme, and shows how a floorplan could

improve the performance of a processor by reducing thermal emergencies.

II. Related works

Thermal-aware floorplanning has been studied in several works. Hung *et al.* [5] use a genetic algorithm to explore different floorplans. They aim at reducing the hotspots and distributing the temperature evenly across the chip, while maintaining the area of the chip. Their study is more focused on circuit level floorplanning, and therefor they use a selected circuit-level benchmarks. The power map for the blocks is also randomly assigned. The temperature of blocks is estimated using HotSpot [6].

Han *et al.* [3] and Sankaranarayanan *et al.* [4] use a simulated annealing based approach to explore the floorplan design space. Their focus is architecturally based. They use a processor floorplan and common CPU benchmarks for their experiments. In addition to the basic block moves, they also explore different aspect ratios for each block in a controlled way. However, they maintain the original area of the block. To estimate the temperature, Han *et al.* use heat diffusion between adjacent blocks. Given the distance of any two blocks and their thermal resistance and power map, the heat diffusion measure is computed and used as temperature estimation. Sankaranarayanan *et al.* run Hotspot at each iteration to compute the steady state temperature of each block.

In our work, we use the same simulated annealing approach as Sankaranarayanan *et al.* [4]. We use a processor floorplan. In addition to the *move* and *aspect ratio*, we also allow for a controlled *resizing* of the block area. While these methods use random power values or compute average power across the benchmarks, we choose to compute the representative power map in a different manner. Our temperature estimation is carried out in grid mode by implementing a Power Blurring thermal simulation method.

III. Methodology

A. Thermal Throttling

Designing a package for worse case thermal behavior of the chip could inflate the cost. A package could be designed for the worst typical application [7]. Applications that dissipate more heat than what the designed package can tolerate should trigger a runtime thermal management technique to keep the processor's temperature in the safe range. Among these techniques are power gating, throttling the clock or issue logic, or changing the power state through DVFS[1].

We implement the throttling policy by gating the clock. For that, a configurable trigger threshold is defined. We choose 100°C as it is close to the junction temperature. When a block in the processor reaches the trigger threshold, the processor stops processing. This in turn reduces the power consumption of the chip down to the leakage power. The processor stays throttled until the critical temperature goes below the trigger threshold. The time a processor spends in throttling could have been spent executing instructions. Hence, the throttling adversely impacts the performance of the processor.

[1]Dynamic Voltage and Frequency Scaling

B. Power Map Selection

The power map used to obtain temperature estimation for the chip in the floorplannning process plays a critical role in the quality of resulting floorplan. Previous works mainly use an average of selected benchmarks. A set of power maps are generated by running a set of benchmark applications (i.e from SPEC CPU benchmark suite). Each power map contains average power consumptions for the functional blocks, computed during the execution of the benchmark. Then the resulted average power for all the benchmarks are averaged together one more time to generate a single power map that contains a power value for each functional blocks. We call this method the *average-based* power selection.

Given the fact that each benchmark utilizes the resources in the processor in a different way, averaging the power traces of different benchmarks could hide a great deal of information about thermal distribution across the chip. The accuracy of these methods could be improved by carefully studying the impact of each benchmark and using the detailed temperature distribution results. However, the rather heavy thermal computation, which in turn results in slow simulation, has been preventing the designers from considering each benchmark, leaving them no choice but averaging all power maps together. Our methodology proposes a more efficient way of selecting power values.

Our power selection method differs in two ways. First, for each benchmark, we only compute the average power around the time during which the processor triggers DTM responses to high temperature. In our case, this means we average power values during the time in which thermal throttling happens. This is to identify the power distributions that becomes critical for reliability or performance.

Second, for each functional block, the maximum power among the benchmarks is selected to form the power map for the chip. The reason for selecting the maximum block power is that different benchmarks (*e.g.*, integer vs. floating point benchmarks) exercise different blocks and as a result have different hotspot distributions. Averaging power maps across the benchmarks could result in high power consumption of a block in one benchmark canceled out by low power consumption of the same block in another benchmark.

We call this method the *selective average-max* power generation (SAM). To show the impact of power selection methods, we generate a power map with each of these methods. For each method, we run the simulation for 4 billion (B) instructions, skipping first 1B and simulating the rest to gather the power numbers. Then we run the floorplanner for both power maps. The results show around 20% improvement by having less thermal throttling in the floorplan resulted from SAM compared to the average-based method..

C. Floorplanning

Our floorplanner is based on simulated annealing approach. We use a modified version of HotFloorplan [4], with the same simulated annealing parameters. The cost function is a linear combination of the total area, maximum temperature, and the estimated wire delay. During each iteration, the tool

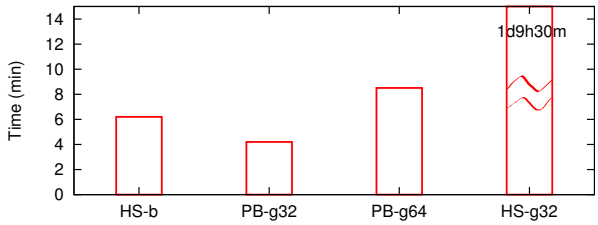


Fig. 1. Execution time for different thermal solver methods.

can change the floorplan through three primitive actions; it can either randomly move, change the aspect ratio or resize a block. The resizing is designed to resolve the blocks with high power density that end up with critical temperature. Those blocks could lead to reliability threatening hotspots which cannot be rectified by re-placing. The current implementation only includes increasing the size up to 50% of the original block size. There is also a constraint of 10% maximum increase in the total area of the chip.

D. Wire Delay

The floorplanning scheme uses the first order wire delay model. In this model, the delay is a linear function of the wire-length. This is simple enough to be used at the architecture level. To consider the impact of the wire delay in performance simulation, we make sure there is no connected blocks with unacceptable wire delay. Should two blocks have high wire delays, we consider the effect in latency of the block in the performance simulator.

E. Power Blurring Thermal Simulation

All the previous works are based on block model temperature estimation. However, the block model approximates the temperature of an entire functional block with a single node, and it could potentially lead to inaccuracy in the estimated temperature when the modeled blocks have very high aspect ratios. To show that, we try a floorplan with two blocks among the entire floorplan having a high aspect ratio. We estimate the steady state temperature with a sample power map. We configure HotSpot once for block model and once for grid model to get the temperature estimations. The results for the two high aspect ratio blocks differ up to 10°C. The error increases at higher temperatures which is the range we are most interested in for thermal-aware floorplanning.

Power Blurring methods has been shown to be fast and accurate for steady state temperature modeling [8]–[13]. This method intrinsically solves the thermal equations in grid mode. With implementation of Power Blurring based thermal modeling, we are able to avoid the aspect ratio sensitive problems that raises by using block model.

Figure 1 shows the time it takes to run the floorplanner using different temperature estimation methods. The experiments are run on an AMD $Opteron^{(tm)}$ Processor 6172. The first part of the labels for each method refers to the solver (HS or PB) and the second part indicates the model type (b for block and gxx for grid with size $xx \times xx$). The PB solver is as fast as the block model solver used in HotSpot. However, using HotSpot with grid model takes two orders of magnetude more time to solve the same equations, which makes it impossible to use for this purpose.

F. Performance Simulator

To study the impact of a floorplan on performance, thermal profile, reliability, power consumption, and performance we configured the following toolchain. For performance simulation we used a modified version of SESC [14] that uses QEMU [15] as the functional emulator executing arm instructions. We configured SESC to pass activity counters to McPAT [16] (every 100K instructions max) which we used for calculating power. We modified McPAT to save the state that it calculates during initialization so that we could call it many times from our simulator. The power numbers from McPAT were used with a modified version of SESCTherm [17] to scale leakage power consumption according to temperature and device properties, and to generate the thermal metrics.

G. Metrics

We use RAMP [1] as a quantitative basis for reliability. This work describes 5 Mean Time To Failure (MTTF) wear out failure models: Electro Migration (EM), Stress Migration (SM), Time-Dependent Dielectric Breakdown (TDDB), Thermal Cycling (TC) and Negative Bias Temperature Instability (NBTI).

Since MTTF is not additive, the average Failures in Time (FIT) per block is estimated as the application executes. The FIT is proportional to the area. At the end of the execution, we add the area-weighted FITs to report the overall FIT value for the entire processors. Like the RAMP authors, we assume that all the different failure mechanisms have the same contribution to the overall FIT value, which is adjusted to a preset value. In our case, we adjust the FIT value for all the SPEC applications to 10,000. This is approximately equivalent to a MTTF of 11 years which is a short but reasonable lifetime for a processor. Table II shows the selected parameters.

Electro migration: occurs when atoms migrate from one end of the interconnect to the other, eventually leading to increased resistance and shorts. The model used in this work for EM is defined as follows:

$$MTTF_{EM} \propto (J)^{-n} \times e^{\frac{Ea_{EM}}{kT}}. \quad (1)$$

Stress migration: Materials differ in their thermal expansion rate, and this difference causes thermo mechanical stress, referred to as Stress Migration. We use the following SM model:

$$MTTF_{SM} \propto |T_0 - T|^{-n} \times e^{\frac{Ea_{SM}}{kT}}. \quad (2)$$

Time-dependent dielectric breakdown: It is the result of the gate dielectric gradual wear out, which leads to transistor failure. Ramp uses TDDB model

$$MTTF_{TDDB} \propto (\frac{1}{V})^{(a-bT)} \times e^{(\frac{X+\frac{Y}{T}+ZT}{kT})}. \quad (3)$$

Thermal cycling: Thermal Cycling is another reliability factor since the temporal thermal gradients, *e.g.*, power on and

off and high frequency changes in power due to changes in workload behavior, affect the lifetime of the processor. There is no validated model for high frequency thermal cycles, but the effects of low frequency cycling can be modeled via:

$$MTTF_{TC} \propto (\frac{1}{T - T_{amb}})^q. \quad (4)$$

Negative bias temperature instability: NBTI leads to upward shifts in the transistors' threshold voltage that leads to timing violations. Ramp uses NBTI model

$$MTTF_{NBTI} \propto ((ln(\frac{M}{1+2e^{\frac{N}{kT}}}) - ln(\frac{M}{1+2e^{\frac{N}{kT}}} - H)) \times \frac{T}{e^{\frac{-I}{kT}}})^{\frac{1}{\beta}}. \quad (5)$$

We report one augmented reliability metrics. For that, we combine all the reliability metrics. We compute the Fault In Time (FIT) measure for each individual reliability metrics, and then use them to compute $MTTF$ for the chip as follows:

$$MTTF = 1/(FIT_{EM} + FIT_{SM} + FIT_{TC} + FIT_{NBTI} + FIT_{TDDB}). \quad (6)$$

In addition to the reliability metrics, we report the gradient temperature across the chip. This is because of thermal gradient impact on interconnect delay [2]. We also report maximum temperature.

IV. Simulation Setup

While both [3], [4] evaluate the impact of the floorplan on processor performance through wire delay, [4] goes further and studies this impact on the processor performance in terms of IPC[2]. IPC is commonly accepted metric for processor performance in a given clock frequency, and is the main reported metric in all the architectural level evaluations.

Nonetheless, these evaluations are merely based on the wire delay, and are carried out to ensure quality of the placement of blocks in the floorplan. None of these methods show how the improved thermal profile of the chip benefits processor's performance.

To evaluate the impact of the thermally-aware improved floorplan on the performance and reliability of the processor, we start with a manually generated floorplan for a processor as our base configuration. Then our automated floorplanner improves the floorplan. We use 8 SPEC2000 CPU workloads in our experiments, namely *applu*, *crafty*, *gzip*, *mesa*, *mcf*, *mgrid*, *swim*, and *twolf*. *applu*, *mesa*, *mgrid*, and *swim* belong to the floating point workload category. Both the base and improved floorplans are run through an integrated performance, power and temperature estimation simulator. The simulator supports thermal throttling. The simulation estimates the IPC for each of the processor configurations that only differ in their floorplan.

For the evaluation, we study two methods and we compare three floorplans. The original floorplan makes the base configuration and hence is called *base*. The two methods that we evaluate are *MAR-A* and *MARS-SAM*. The first set of abbreviations in the naming stands for the supported primitives in the floorplan scheme. The second set of abbreviations refers to the power selection method. *MAR-A* implements 2 basic primitives in the floorplanning scheme, Move, and Aspect-Ratio, and it uses the Average-based power selection method. *MARS-SAM* has one additional primitive, reSizing of a block area, in addition to the primitives supported by *MAR-A*. It also uses the Selective-Average-Max power generation method. Figure 3(a) shows the floorplan manually generated as the base configuration. Area of each block is also indicated in the figure. The unit for the numbers is MM^2. L2 cache is not shown to save space. It is placed on top of the core with area of 3.90×10^{-6} mm^2.

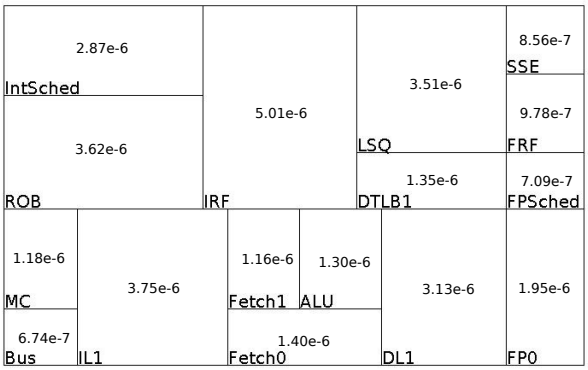


Fig. 2. The processor's core floorplan used as the base configuration. L2 cache is not shown. Total area is 72.36×10^{-4} mm^2

The processor configuration parameters are shown in Table I. Table II shows the reliability metric parameters.

Parameter	Value
Frequency	2.0 GHz
ICache	32KB 2-way (2 cycle hit)
DCache	32KB 8-way (3 cycle hit)
L2Cache	1MB 16-way (12 cycle hit)
Mem. Lat.	180 cycles
Branch Pred.	Hybrid 76Kb total memory
Issue width	4
ROB	192
Inst. Win.	48
Phy. Reg. (I-F)	128-128

TABLE I
ARCHITECTURAL PARAMETERS.

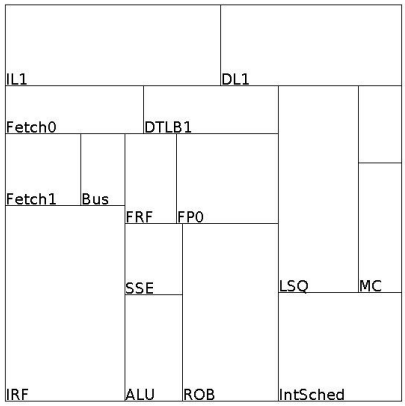
Metric	Parameters
EM	$a_{EM} = 0.9$, $k = 8.617343 \times 10^{-5}$
SM	$n = 2.5, a_{SM} = 0.9$
$TDDB$	$V = 1.1$, $a = 78$, $b = -0.081$,
	$X = 0.759$, $Y = -66.8$, $Z = -8.37$
TC	$T_{amb} = 293$, $q = 2.35$
$NBTI$	$M = 1.6328$, $N = 0.07377$
	$I = -0.06852$, $\beta = 0.3$, $H = 0.01$

TABLE II
THERMAL METRICS CONSTANTS.

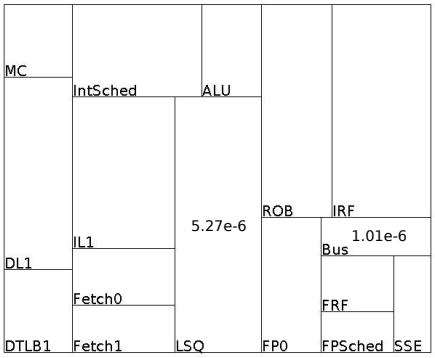
[2]Instruction Per Cycle

V. Evaluation

Table III presents the results in terms of percentage of time the processor spends in thermal throttling, and also the impact on it's performance. The original floorplan results in a processor that on average spends 5.96% of the execution time of the benchmarks being throttled because of high temperature. *MAR-A* improves the results by reducing the time spent in throttling by around 6 fold to 1.1%. However, *MAR-A* cannot completely rectify the thermal throttling issue. *MARS-SAM* on the other hand, is able to completely rectify all the throttling issues. As a result, *MARS-SAM* achieves better performance results. While *MAR-A* improves the performance by 6.2% on average across the selected benchmarks, *MARS-SAM* extends the improvement up to 8.8%.



(a) *MAR-A* floorplan



(b) *MARS-SAM* floorplan

Fig. 3. Floorplan resulted from different methods. L2 is not shown to save space. It is placed on top of the core for *MAR-A*, and on the left for *MARS-SAM*. Blocks that have different area than the base floorplan are indicated by their area inside the block (unit is mm^2).

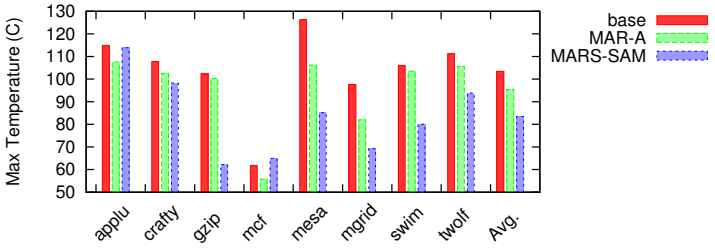
Figure 3 shows the floorplan resulted from each of the methods. L2 is not shown to save space. For *MAR-A*, it is placed on top of the core, and it is located on the left for *MARS-SAM*. Blocks that have different area than the base floorplan are indicated by their area inside the block. *MARS-SAM* could resolve the high power density problem of a block by increasing it's area. As a result the floorplan takes more area. While *MAR-A* increases the total area by 0.06% (due to gaps between some blocks), *MARS-SAM* has to expand two blocks (LSQ and Bus) to resolve the high density problem to the point that there is no thermal throttling. This results

Benchmark	% in Throttling			% IPC Improvement	
	base	*MAR-A*	*MARS-SAM*	*MAR-A*	*MARS-SAM*
applu	1.7	0.2	0.0	1.9	2.3
crafty	11.2	0.5	0.0	13.3	14.0
gzip	0.1	0	0.0	2.2	2.2
mcf	0	0	0.0	0	0
mesa	14.8	2.5	0.0	13.8	22.2
mgrid	0	0	0.0	0	0
swim	3.5	0.5	0.0	4.1	6.3
twolf	16.4	5.1	0.0	14.3	23.4
Avg.	5.96	1.1	0.0	6.2	8.8

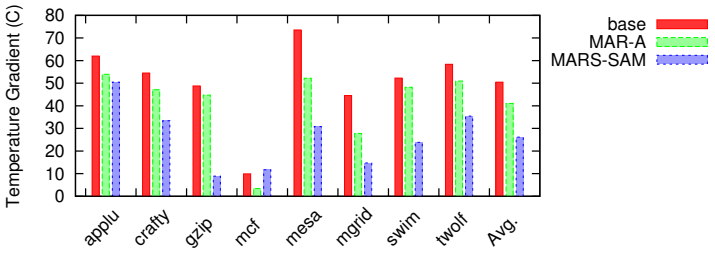
TABLE III
IMPACT OF EACH FLOORPLAN ON THE PERFORMANCE. TT STANDS FOR THERMAL THROTTLING.

Method	Area (mm^2)	overhead
base	72.36×10^{-4}	-
MAR-A	72.36×10^{-4}	-
MARS-SAM	74.51×10^{-4}	Total:3% LSQ:50% Bus:50%

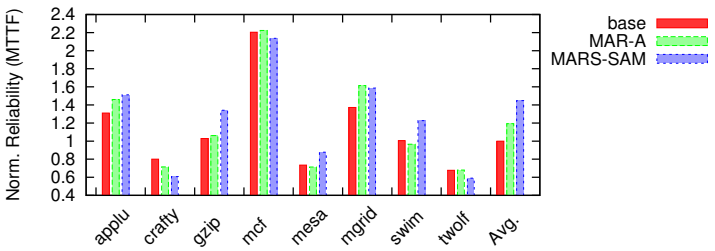
TABLE IV
AREA OF EACH FLOORPLAN AND THE OVERHEAD COMPARED TO THE *base*.



(a) Max Temperature



(b) Temperature Gradients



(c) Reliability

Fig. 4. Thermal metric results.

in 3.0% increase in the total area. However, the silicon real state is less critical factor in the design of microprocessors compared to parameters like clock frequency. Hence trading off performance and reliability for a small increase in the area is acceptable. Table IV summarizes the area measures for all the evaluated methods.

Figure 4(a) shows around 20°C reduction of the maximum temperature on average across the benchmarks. Gradient temperature across the chip could influence both reliability and performance. High thermal gradient across the chip could cause timing failure. Figure 4(b) shows the same range of reduction on the thermal gradients across the chip on average.

Estimation of exact mean to time failure of a chip in a simulation environment is not simple. But normalized results can be used to relatively compare two methods. Figure 4(c) shows the normalized reliability metrics. *MARS-SAM* results in a floorplan that on average has more than 40% longer mean time to failure compared to *base*. Note that the average reliability of the chip is computed as the inverse of average FIT measure for each benchmark. One might argue that the reliability of the chip is determined by the worse case benchmark. But since the collection of these benchmarks represent the typical usage of the chip, averaging the FIT measure of all of them shows the overall effect.

Not all the benchmark workloads show improvement. The reason for that is two fold. First, thermal throttling limits the maximum temperature, and caps the reliability degradation. Second, lowering maximum temperature in hotspots might come at the cost of increase in the average temperature in the region. To show the reliability of the chip without capping it through throttling, we disable the throttling and rerun the experiments. The average normalized reliability measure for the chip degrades down to 0.3 compared to the *base* configuration.

Leakage is the temperature dependent component of power consumption. The $Leakage/Total power$ ratio increases as the technology size shrinks. In our simulation setup, leakage accounts for around 30% of the total power consumption. Our results show that *MARS-SAM* reduces the leakage by 7% compared to *base*.

VI. Conclusion

This study shows the importance of thermal-aware floorplanning in improving both reliability and performance of a processor. The improvement in the reliability comes from less hotspots across the chip. Lower temperature leads to less thermal emergencies on the chip that would have caused the processor performance to suffer. We use a Power Blurring thermal model to estimate temperature in the floorplanning process. The Power Blurring method fundamentally works in grid model, and it is as fast as the block model solver used in HotSpot. We implement a simulated annealing based floorplanning scheme, empowered with three primitive actions: move, aspect-ratio, and resizing. While the first two primitives leverage lateral heat transfer, the last one helps averting critical hotspots by expanding area of the block in a controlled way. We introduce a more efficient method of selecting a power map for the floorplanning process. Our experiments show around 8.8% increase in performance as a result of averting all thermal emergencies, at the cost of 3% increase in the chip total area. The reliability of chip is improved by 40% as a result of lower hotspots. Power consumption is another design parameter that has been improved by 7% reduction in the leakage component.

VII. Acknowledgment

We would like to Thank Dr. Xi Wang and Rigo Dicochea for their valuable comments and feedback toward the improvement of the paper. This work was supported in part by the National Science Foundation under grant 1059442; Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the NSF.

References

1. J. Srinivasan, A. Sarita, P. Bose, and R. Jude. *Lifetime reliability: toward an architectural solution*, In IEEE MICRO, volume 25, pages 70-80, 2005.
2. A. H. Ajami, K. Banerjee and M. Pedram, *Modeling and Analysis of Non-uniform Substrate Temperature Effects on Global ULSI Interconnects*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 24, no. 6, pp. 849-861, 2005.
3. 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.
4. K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron. *A case for thermal- aware floorplanning at the microarchitectural level*, The Journal of Instruction-Level Parallelism, 7, 2005.
5. 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.
6. K. Skadron and M. R. Stan and W. Huang and S. Velusamy and K. Sankaranarayanan and D. Tarjan,*Temperature-Aware Microarchitecture*, in Proceedings of the 30th Annual International Symposium on Computer Architecture, (ISCA), 2003, pp. 2-13,
7. D. Brooks and M. Martonosi, *Dynamic Thermal Management for High-Performance Microprocessors*, Proc. 7th International Symposium on High Performance Computer Architecture (HPCA01), IEEE CS Press, 2001, pp. 171-182.
8. T. Kemper, Y. Zhang, Z. Bian, and A. Shakouri, *Ultrafast Temperature Profile Calculation in IC chips*, Proc. of 12th International Workshop on Thermal investigations of ICs (THERMINIC), Nice, France, pp. 133-137, 2006.
9. J. H. Park, X. Wang, A. Shakouri, and S.-M. Kang, *Fast Computation of Temperature Profiles of VLSI ICs with High Spatial Resolution*, Proc. 24th Semi-Therm, San Jose, CA, pp. 50-54, 2008.
10. J. H. Park, A. Shakouri, and S.-M. Kang, *Fast Evaluation Method for Transient Hot Spots in VLSI ICs in Packages*, International Symposium on Quality Electronic Design, San Jose, CA, pp. 600-603, 2008.
11. J. H. Park, S. Shin, J. Christofferson, A. Shakouri, and S.-M. Kang, *Experimental Validation of the Power Blurring Method*, Proc. 26th Semi-Therm, Santa Clara, CA, pp. 240-244, February 21-25, 2010.
12. A. Ziabari, Z. Bian, and A. Shakouri, *Adaptive Power Blurring Techniques to Calculate IC Temperature Profile under Large Temperature Variations*, International Microelectronic and Packaging Society (IMAPS) ATW on Thermal Management, Sept 28-30, 2010.
13. A. Ziabari, E. K. Ardestani, J. Renau, and A. Shakouri, *Fast Thermal Simulators for Architecture Level Integrated Circuit Design*, SEMI-THERM 2011.
14. J. Renau, F. Basilio, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, and P. Montesinos, *SESC simulator*, January 2005, http://sesc.sourceforge.net.
15. F. Bellard, *QEMU, a fast and portable dynamic translator*, in Proceedings of the USENIX Annual Technical Conference, ATEC, 2005, Anaheim, CA, pp 41-41
16. Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi, *Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures*, in Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, 2009, MICRO 42, pp. 469-480.
17. J. Nayfach and J. Renau, *SOI, Interconnect, Package, and Mainboard Thermal Characterization*, International Symposium on Low Electronics and Design (ISLPED), pp. 327-330, Aug. 2009.