View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Infoscience - École polytechnique fédérale de Lausanne

3D Thermal-Aware Floorplanner for Many-Core Single-Chip Systems

David Cuesta
Complutense University
Madrid, Spain
dcuestag@pdi.ucm.es

José L. Risco-Martin
Complutense University
Madrid, Spain
jlrisco@dacya.ucm.es

José L. Ayala
Complutense University
Madrid, Spain
jayala@fdi.ucm.es

David Atienza
EPFL
Lausanne, Switzerland
david.atienza@epf .ch

***Abstract*—Heat removal and power density distribution delivery have become two major reliability concerns in 3D stacked technology. In this paper, we propose a thermal-driven 3D floorplanner. Our contributions include: (1) a novel multi-objective formulation to consider the thermal and performance constraints in the optimization approach; (2) an efficient *Mixed Integer Linear Programming (MILP)* representation of the floorplanning model; and (3) a smooth integration of the MILP model with an accurate thermal modelling of the architecture. The experimental work is conducted for two realistic many-core single-chip architectures: an homogeneous system resembling Intel's SCC, and an improved heterogeneous setup. The results show promising improvements of the mean, peak temperature and the thermal gradient, with a reduced overhead in the wire length of the system.**

I. Introduction

Current computer architectures have reached a performance barrier. Patterson has formulated the problem as follows: The power wall + the memory wall + the ILP wall = a brick wall for serial performance. This can be understood like the limit in materials and architectures to provide increasing throughput. Therefore, computer architects have been forced to turn to parallel architectures to continue to make progress. Parallelism can be exploited by using the additional transistors (forecasted by Moores law) to add more independent CPUs, data-parallel execution units, additional registers sets for hardware threads, bigger caches, and more independent memory controllers to increase memory bandwidth.

The emergence of heterogeneous many core architectures presents a unique opportunity for delivering order of magnitude performance increases to high performance applications by matching certain classes of algorithms to specif cally tailored architectures. Specif c HPC applications like N-Body Simulations, Molecular Dynamics, and Terrain Rendering [1] can experience order of magnitude or greater speedups when paired with architectures that are specif cally tailored to their needs. Similar examples from the HPC community include the Los Alamos National Labs Roadrunner system.

This trend on executing the target applications in many parallel cores is not only one of the characteristics of the current datacenters, but also multi-processor systems-on-chip (MPSoCs) have reached the category of many-core systems. Intel Labs has created an experimental Single-chip Cloud Computer, (SCC) a research microprocessor containing the most Intel Architecture cores ever integrated on a silicon CPU chip 48 cores. [2] It incorporates technologies intended to scale multicore processors to 100 cores and beyond, such as an on-chip network, advanced power management technologies and support for message-passing.

The exponentially increasing power densities that have been reached in current technologies, the values of leakage currents, the cooling costs and the recent reliability constraints in microprocessor-based systems have motivated the cooling down of the chip temperature to be one of the main concerns in system design. The operating temperature has a signif cant impact on microprocessor design. At higher temperatures, transistors work slower due to the degradation of the carrier mobility. The resistivity of the metal interconnects also increases, causing longer delays and performance degradation.

Reliability is also strongly related to temperature, and increasing the temperature exponentially decreases the chip lifetime. In fact, time to failure is a function of $e^{-Ea/kT}$, where Ea is the activation energy of the failure mechanism and accelerated by temperature increase, k is the Boltzmann's constant, and T is the absolute temperature. When the operating temperature exceeds a threshold, the effect on reliability can be permanent and seriously impact lifetime. Each 10 degrees rise reduces the life of the component by a 50%. Hence, lower operating temperatures of components maximize reliability.

In order to maintain the chip temperature under a certain limit, the power density of the hardware modules can be decremented by increasing the chip area. However, this is not admissible in terms of cost, and the problem of meeting all the geometric constraints should be solved.

Orthogonal to the power density of the functional blocks, another important factor that affects the temperature distribution of the chip is the lateral spreading of heat in silicon. This depends on the placement of the units and their proximity to the chip border and other units that behave as thermal sinks or thermal sources. Thermal-aware f oorplanning algorithms are able to even out the temperature of the hardware modules through spreading of the heat dissipation. This aspect of f oorplanning is particularly attractive in comparison with static external cooling, that reduces the temperature of the chip surface by a constant factor (it does not reduce the temperature gradient across the chip).

Three-dimensional (3D) multi-processor chips have been proposed as an effective mechanism to signif cantly improve

978-1-4577-1490-0/11/$26.00 ©2011 IEEE

system performance by reducing interconnect delays and increasing the density of the integrated logic, turning the "many-core single-chip" into a reality. It also allows the integration of multiple and disparate technologies, such as radio frequency and mixed signal components.

A major concern in the adoption of 3D architectures is the increased power densities that can result from placing one computational block over another in the multi-layered 3D stack. Also, the thermal conductivity of the dielectric layers inserted between device layers for insulation is very low compared to silicon and metal. Since power densities are already a major concern in 2D architectures, the move to 3D architectures will accentuate the thermal problem. Consequently, it is mandatory to devise efficient 3D floorplanning mechanisms that optimize the thermal profile of these complex 3D multi-processor architectures.

This work continues the initiated in [3] that proposes a set of design rules for the generation of thermal-aware floorplans for the 3D Niagara architecture. This previous work obtained improvements of the thermal metrics with respect to the baseline architecture and compared to traditional thermal-aware floorplanner. The work presented in this paper outperforms this results with a MILP formulation and an efficient solver that manages multiple objectives in the minimization problem, as well as considering a many-core heterogeneous single-chip for experimental purposes.

This paper specifically makes the following contributions:

1) it provides a novel multi-objective formulation of the floorplanning problem in 3D multi-processor architectures with thermal constraints.
2) it performs an efficient resolution of the optimization problem by the use of a *Mixed Integer Linear Programming (MILP)* framework.
3) it shows good response in terms of the main thermal metrics (mean temperature, peak temperature and thermal gradient) for a many-core homogeneous and heterogeneous single-chip architecture that resembles the Intel's SCC.

II. Related Work

The impact of the floorplanning on the thermal distribution of real microprocessor-based systems is analyzed in [4], where the placement of components for Alpha and Pentium Pro is evaluated. Some initial works on thermal aware floorplanning [5] propose a combinatorial optimization problem to model our problem. However, the simplification of the considered floorplan and the lack of a real experimental framework motivated the further research on the area. Thermal placement for standard cell ASICs is a well researched area in the VLSI CAD community, where we can find works as [6].

In the area of floorplanning for microprocessor-based systems, some authors consider the problem at the microarchitectural level [7], where it is shown that significant peak temperature reduction can be achieved by managing lateral heat spreading through floorplanning. Other works [8] use genetic algorithms to demonstrate how to decrease the peak temperature while generating floorplans with area comparable to that achieved by traditional techniques. [9] uses a simulated annealing algorithm and an interconnect model to achieve thermal optimization. These works have a major restriction since they do not consider multiple objective factors in the optimization problem, as opposed to our work. Our proposed floorplanner optimizes jointly both thermal metrics (mean temperature, peak temperature and gradient) with a strong impact on the reliability of the system, and the performance of the system (through the minimization of the wire length delay). Moreover, the thermal models used in these studies do not reflect the complex diffusion processes that exist in current technologies. More recent works [10] have tackled the problem of thermal-aware floorplanning with geometric programming but, in this case, the area of the chip is not considered constant.

Thermal-aware floorplanning for 3D stacked systems has also been investigated. Cong et al. [11] proposed a thermal-driven floorplanning algorithm for 3D ICs, which is a natural extension of his previous work on 2D. In [12], Healy et al. implemented a multi-objective floorplanning algorithm for 2D and 3D ICs, combining linear programming and simulated annealing.

Our work presents more similarities with the reference [13] by Hung, where they propose a thermal-aware floorplanner for 3D architectures. However, this preliminary work does not consider the multi-objective approach proposed in our work, and does not consider the minimization of those thermal variables with a strong impact on the reliability of the system.

Thus, an efficient model of the optimization problem and an effective solver are required to achieve good trade-offs between thermal optimization and performance constrains. In the case of 3D IC design, incremental optimization is a promising way to handle multi-objective optimization with complicated constraints and facilitate the design reuse technology. Several works concerned with incremental floorplanning for 2D IC design [14], [15] have been proposed, but none has taken thermal-aware 3D IC design into consideration. [16] has recently proposed an incremental MILP algorithm. However the design process could take several iterations, whereas our methodology perform the thermal-aware and total wire length optimization in two steps.

In this paper, we propose a novel algorithm to optimize the 3D layout in order to eliminate the hotspots, reduce the peak temperature and decrease the reliability risks. Given a 3D packing and a chip area, we formulate the thermal-aware and total wire length optimization into two MILP problems. The former is defined in terms of power density, moving hottest blocks until they are as far away as possible from each other. Then, with hottest blocks fixed in space, we perform another optimization trying to move the remaining blocks to reduce total wire length. Experiments results show that we can reduce the maximum on-chip temperature in 54 degrees on average for two realistic homogeneous and heterogeneous many-core single-chip architectures, outperforming previous thermal-aware floorplan designs.

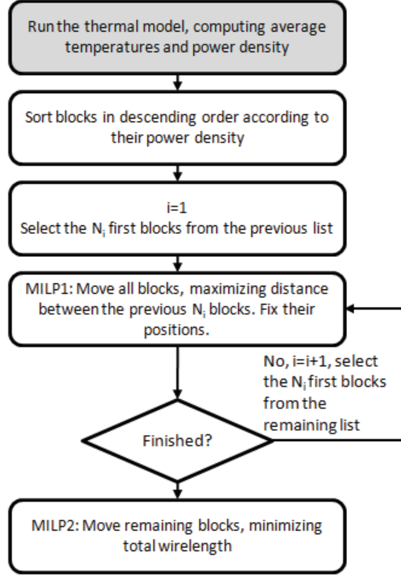

Fig. 1. Iterative f ow of our approach

III. Design flow

In order to reduce maximal on-chip temperature as much as possible, we propose a novel thermal-aware incremental optimization f ow. To this end, we have developed three algorithms. The f rst one performs an accurate analysis of the thermal behavior in the 3D IC. The second moves all the blocks until the hottest ones are as far away as possible from each other. The last one, having f xed the hottest, tries to move the remaining blocks while total wire length is minimized.

Fig. 1 shows the design f ow of our thermal-aware 3D microarchitectural f oorplanner. Such f ow can be divided in two phases. The former is the thermal analysis of an initial conf guration of the 3D IC (gray shaded block in Fig. 1). Since we are studying the Niagara system, such initial conf guration is available in [17]. If not, we can obtain it by running the second MILP algorithm proposed in this work (last block in Fig. 1). The second phase is the optimization loop (rest of the diagram). Next, we describe these two phases in detail.

A. Thermal analysis

As Fig. 1 shows, we f rst perform a thermal analysis. To this end, we have developed an accurate thermal model, which is brief y described in the following.

3D integration consists on placing different active layers using silicon dioxide and joining them with a glue material. If inter layer communication is required, Through Silicon Vias (TSVs) allow it. Some of the goals on the design of 3D stacks are to achieve The 3D stack is built over an adiabatic PCB surface and then, traditional technological dies composed by silicon dioxide and silicon, are placed one over the others.

The heat f ow through the 3D stack is diffusive, hence, it can be characterized with a 3D RC thermal model as the one presented in [18].

The thermal modeling of the stack is performed after splitting the chip into small cubic unitary cells. These cells are modeled with six thermal resistances and one thermal capacitance as

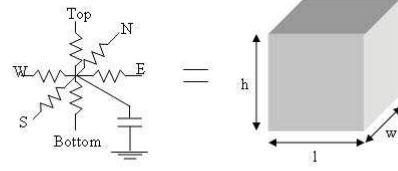

Fig. 2. Equivalent RC circuit of a single cell

can be seen in Figure 2. Four of these resistances connect each cell to its lateral neighbors (those on the same layer), while the two remaining resistances connect the cell with the upper and bottom cell, respectively. The capacitance represents the heat storage inside the cell. The values of the conductances and the capacitance are calculated using these expressions:

$$G_{top/bottom} = k_{th}(l \cdot w)/(h/2) \quad (1)$$

$$G_{north/south} = k_{th}(l \cdot w)/(h/2) \quad (2)$$

$$G_{east/west} = k_{th}(l \cdot w)/(h/2) \quad (3)$$

$$C_{top} = sc_{th}(l \cdot w \cdot h) \quad (4)$$

where north, south, east and west indicate the direction in which heat is diffused; k_{th} and sc_{th} are the thermal conductivity and specif c heat capacity per volume unit of the material, respectively.

The model also considers the heat diffusion to the surrounding environment. The existence of TSVs is considered in the model, also as a resistance element. The most important thermal properties of the material used in the model are listed in Table I.

TABLE I
Thermal properties of materials.

Silicon linear thermal conductivity	295 W/(mK)
Silicon quadratic thermal conductivity	-0.491 W/(mK^2)
Silicon dioxide thermal conductivity	1.38 W/(μmK)
Silicon specif c heat	1.628 x 10^6 J/m^3K
Silicon dioxide specif c heat	4.180 x 10^6 J/m^3K

Once the resistance and capacitance values for every unitary cell are calculated, a set of equations that describe the RC grid is created. After that, an iterative method (Forward Euler) is used to solve it.

The active elements in the 3D stack can be considered as heat sources or heat sinks. Processors are considered strong heat sources, they dissipate power in the die, and this heat is then spread throughout the chip. On the other hand, memories have a lower power activity and they can be considered almost as heat sinks. This is an important consideration to be taken into account since the f oorplanner will try to place both heat sinks and heat sources as close as possible (provided the routing and performance constraints) to balance the thermal prof le.

Once the previous model has been applied to the 3D IC, we obtain mean and peak temperatures, as well as the thermal gradient and power density, which are used later in the optimization phase.

B. Optimization phase

The proposed approach tries to minimize two conflicting objectives: maximum temperature (J_1) and total wire length (J_2). Is this article, instead of minimizing a weighted sum of both objectives like in previous approaches (see for example [16]), we perform a multi-objective optimization. Thus, the first question is: Which search algorithms should be used? The floorplanning optimization problem is NP-hard. As a result, a solver could not find the global optimum in a finite time. There are several approaches to perform such optimization efficiently. Examples are the use of meta-heuristics (like simulated annealing, genetic algorithms, particle swarm optimizers, etc.), as well as classic methodologies based on MILP. We used MILP because of two main reasons: (1) MILP solvers immediately check if a design scenario is or not feasible, and (2) if the problem is correctly formulated, MILP quickly offers feasible solutions.

To perform a multi-objective optimization in MILP, we proceed as follows. In general terms, all the problem objectives, except one, are introduced into the set of constraints to arbitrarily give a value in the right side of each new constraint (one per objective). Let us suppose that V_2, corresponds to the value of J_2 (total wire length) when J_1 (maximum temperature) is minimized. If the constraint $J_2 = V_2$ is added and the problem is solved again, it would once again obtain the same solution for J_1. But if the constraint $J_2 \leq V_2 - \epsilon$ is added where ϵ is a relatively small positive value, and the problem is solved, it is possible that the new solution of J_1 is superior or equal, but obviously never inferior, since when adding a new constraint, the number of feasible solutions is reduced. Therefore, as the value of J_2 is decreased and new problem instances are solved, new solutions for J_1 are generated. The process stops when the right side of the constraint, $V_2 - \epsilon$, reaches the optimal value of J_2. The problem resides in finding the value of ϵ adapted to being able to generate the maximum number of efficient points in the space of objectives. In our case, since we are trying to outperform an initial configuration (i.e., the input to the thermal model) obtaining better solutions for both maximum temperature and wire length, the value of ϵ can be easily obtained without running the MILP algorithms several times.

To develop a MILP model, we must perform several linear approximations. The first one is related to the thermal model, which includes non-linear and differential equations. The temperature of an element in an integrated circuit depends on the power density of the element and the proximity of its neighbors. The first factor of the equality refers to the increase on the thermal energy due to the activity of the element, while the second factor is related with the diffusion process of heat [19]. Thus, we use the power density of each block as an approximation of its temperature in the steady state. The second approximation is the distance between elements, which is approximated as the Manhattan distance.

Continuing with the optimization phase (second block of Fig. 1) and after analyze the temperature contribution from different blocks, we first sort the blocks in descending order according to their power density. Next, we select the first $N_i, i = 1$ blocks from the previous list, which will be the hottest ones. The selection of these N_i blocks is user-defined. In our experiments we selected all the cores, because their positions decide the final temperature distribution along the 3D IC. Next, the first search algorithm (called MILP1 in Fig. 1) allocates all the blocks, trying to maximize the Manhattan distance (d_m) between the aforementioned N_i modules:

$$\text{maximize } J_1 = \sum_{i,j \in N_i, i<j} d_m(b_i, b_j) \quad (5)$$

Next, the optimization can be repeated several times allocating the following set of $N_i, i = 2$ blocks of the remaining sorted list (having the previous N_1 blocks fixed in the final design). This procedure can be repeated until the sorted list is empty. Finally, we move the remaining blocks (considered those with lowest power density) using a second algorithm (called MILP2 in Fig. 1). In this phase, we do not maximize distance. Instead, we minimize total wire length, approximated as the Manhattan distance between connected blocks (C).

$$\text{minimize } J_2 = \sum_{i,j \in C, i<j} d_m(b_i, b_j) \quad (6)$$

In the following, we will define the experimental set-up, showing the floorplans that will be thermally analyzed and compared with the results obtained by our floorplanner.

IV. Experimental Set-Up

The 3D multiprocessor systems studied in our experimental work are based on the Intel's SCC architecture but their processing units are SPARC cores, like those in the Niagara architecture, fabricated in 90nm technology (these cores are much more powerful than the Power cores found in SCC and can also easily exhibit higher thermal issues). This architecture has been modified to include an increased number of cores, placed in several layers of the 3D stack. Since our floorplanner can place a variable number of cores in every layer, the area and power consumption of the crossbar is scaled accordingly to the number of cores found in every layer and their required bandwidth. The inter-layer communication is resolved with a set of TSVs that route the communication signals.

The floorplanner will place the functional units that compose the 3D multi-processor architecture targeting both temperature and wire length optimization. The thermally-optimized floorplans proposed by the floorplanner will be compared with the original configuration presented in Figure 3. The 64 cores (C) are disposed in 5 layers, where also the L2 memories (L2), the shared memories (L2B) and the crossbar (Cross) can be seen.

The experimental work will analyze the thermal optimization achieved by the floorplanner in two different scenarios. The first scenario resembles the SCC architecture with a system where 64 SPARC cores are integrated in the 3D stack.

The second scenario models an heterogeneous system where the 64 cores are composed of 48 SPARC and 16 Power6. The

Power cores are placed instead of cores C2,7 and 12 in the case of inner layers, and C2,7,11 and 12 for the top layer. This setup will show the optimized thermal prof le that can be expected when multiple core architectures are considered, as well as the extra optimization opportunities that the f oorplanner will f nd.

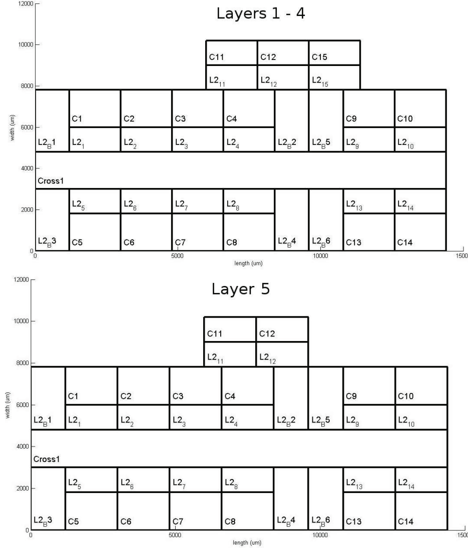

Fig. 3. Original f oorplan.

V. Results

This section f rstly presents the thermal prof le of the two scenarios described in Section IV estimated by the thermal model.

The metrics considered for the analysis of the experimental results are the wire length, mean and maximum temperature of the layer, and the maximum thermal gradient. These metrics are usually found in all the thermal-related analysis.

Then, these results will be compared with the thermal prof les exhibited by the outputs of the f oorplanner.

The worst case of power consumption in the Niagara2 (84W at 1.1V and 1.4GHz [17]) is considered to extract the power densities of every SPARC unit. Also, the area of the layers has been scaled according to the number of cores and the number of layers has bem increased. The power of the Power6 cores is 2.6 W, as found in [20].

A. Scenario 1

Figure 4 shows the thermal maps for the simulation of the homogeneous system. The previously def ned thermal metrics, the mean temperature, thermal gradient and maximum temperature for every layer of the conf guration, have been calculated. The comparison with the baseline homogeneous system shows that the f oorplanner is capable of optimizing the maximum temperature in 55 degrees, the mean temperature in 24 and the thermal gradient is decreased in 70.

This can be explained because our f oorplanner separates heat sources (cores) as much as possible, trying to place them at the border of the chip, helping on the cooling down of the cores. The f oorplanner also takes into account vertical heat spread, and each layer will have a different layout, avoiding placing heat sources one over the other.

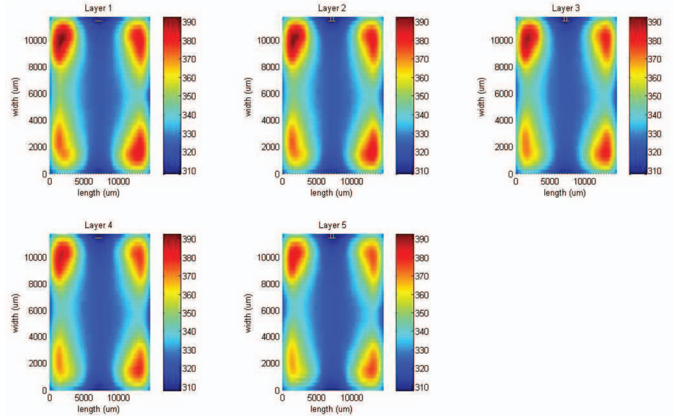

Fig. 4. Thermal maps of the optimized homogeneous system.

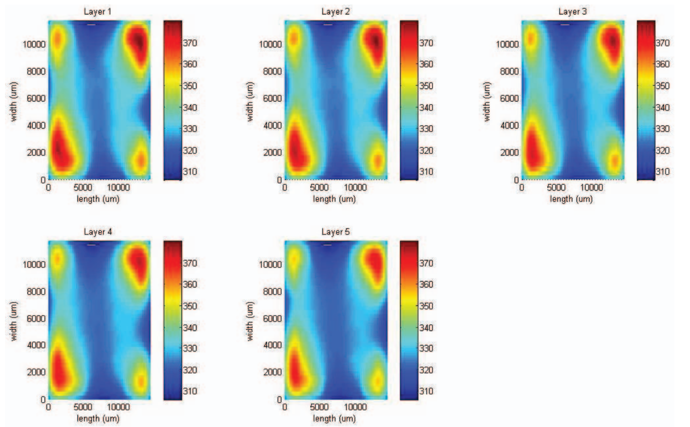

Fig. 5. Thermal maps of the optimized heterogeneous system.

TABLE II
Thermal deviation (K)

Scenario	*Layer 1*	*Layer 2*	*Layer 3*	*Layer4*	*Layer 5*
$S.1Original$	33.74	33.34	32.55	31.37	29.79
$S.1Floorplanner$	17.75	17.46	16.90	16.51	15.89
$S.2Original$	28.65	28.31	27.62	26.60	25.21
$S.2Floorplanner$	14.61	14.32	13.89	13.67	13.31

Also, as shown in Table II, the reduced deviation of temperatures across the layers (or spatial thermal gradients) determines a more homogeneous thermal distribution, which is translated into a reduced reliability risk and diminished leakage currents.

B. Scenario 2

Figure 5 shows the thermal maps for the simulation of the heterogeneous system. Similarly to the previous setup, the mean temperature, thermal gradient and maximum temperature for every layer of the conf guration have been calculated. The comparison with the baseline heterogeneous system shows that the f oorplanner is capable of optimizing the maximum temperature in 52 degrees, the mean temperature in 23 and the

thermal gradient is decreased in 78. Also, the heterogeneous architecture outperforms the results of the homogeneous system in 13 degrees for the maximum temperature, 7 for the mean temperature and 10 for the thermal gradient.

In this case, our floorplanner will try again to maximize the distance between the heat sources. However, the Power cores will not be considered as hotspot by the optimizer since their temperature is much lower than the SPARC's temperature. Therefore, the floorplanner will place the warm Power cores between actual hotspots, achieving a better thermal profile.

Also, as shown in Table II, the reduced deviation of temperatures across the layers determines a more homogeneous thermal distribution, which is translated into a reduced reliability risk and diminished leakage currents.

Table III shows the wirelength associated with the 3D system. As can be seen, the overhead incurred by the floorplanner has been a 24% when compared to the original distribution for Scenario 1 and 31% for Scenario 2. This overhead in the wiring is not directly translated into an increase of the communication delay because core-to-core communication is regulated by the crossbar. As the crossbar is the module that limits the bandwidth and speed of the link, this overhead is seen minimized. On the other hand, the big savings reached in the mean temperature (35% and 41%for the Scenario 1 and 2, respectively) justify the overhead in wiring.

TABLE III
WIRE LENGTH

Number of Layers	*Floorplan*	*Wire Length (um)*	Mean *Temperature (K)*
$Scenario1$	Original	3742	366
	Floorplanner	4644	343
$Scenario2$	Original	3742	356
	Floorplanner	4918	333

C. Comparison of results

The results obtained by our algorithm have been compared with those extracted by traditional placement algorithms. For these comparison purposes, a Genetic Algorithm (GA) with reverse polish notation that performs the placement of units has been selected, as seen in previously published works. Table IV shows the results obtained by our algorithm and the GA for two chip configurations with 16 and 64 cores, respectively.

TABLE IV
COMPARISON OF RESULTS

# cores	*Mean T MILP*	*Wire MILP*	*Mean T GA*	*Wire GA*
16	325	595	329	738
64	333	4644	NA	NA

As can be seen, our algorithm not only achieves better results in terms of mean temperature and wiring length, but also it is able to target more complex systems. The GA was not able to converge to a solution in the case of a chip with 64 cores after 6 days of running.

VI. CONCLUSIONS

This paper has proposed a novel MILP formulation to cope with the problem of thermal-aware floorplanning in 3D many-core sigle-chips. Also, the efficient solver that provides the optimization of the floorplan, interfaces with an accurate thermal model, providing promising results in the minimization of the main thermal- and reliability-related metrics (peak and mean temperature, thermal gradients) with low performance overhead. The experimental results have been obtained for two realistic many-core single-chip architectures: an homogeneous system resembling Intel's SCC, and an improved heterogeneous setup. These results outperform previous results obtained by traditional thermal-aware floorplanners.

ACKNOWLEDGMENT

This work is partially supported by the Spanish Government Research Grant TIN2008-00508. Also, it is partially supported by the Nano-Tera RTD project CMOSAIC (ref.123618), financed by the Swiss Confederation and scientifically evaluated by SNSF, and the PRO3D EU FP7-ICT-248776 project.

REFERENCES

[1] V. T. B. Minor and G. Fossum, "Terrain Rendering Engine (TRE)," IBM, Tech. Rep., 2005.
[2] Intel Labs, *SCC External Architecture Specification (EAS)*, May 2010.
[3] D. Cuesta, J. L. Ayala, J. Hidalgo, M. Poncino, A. Acquaviva, and E. Macii, "Thermal-aware floorplanning exploration for 3D multi-core architectures," in *GLSVLSI*, 2010, pp. 99–102.
[4] I. K. Y. Han and C. Moritz, "Temperature aware floorplanning," in *Workshop on Temperature-Aware Computer Systems*, 2005.
[5] C. C. N. Chu and D. F. Wong, "A matrix synthesis approach to thermal placement," in *International Symposium on Physical Design*, 1997.
[6] G. Chen and S. Sapanetkar, "Partition-driven standard cell thermal placement," in *International Symposium on Physical Design*, p. 2003.
[7] 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*, vol. 7, pp. 8–16, 2005.
[8] W.-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, and M. J. Irwin, "Thermal-aware floorplanning using genetic algorithms," in *International Symposium on Quality Electronic Design*, 2005.
[9] Y. Han and I. Koren, "Simulated annealing based temperature aware floorplanning," *Journal of Low Power Electronics*, vol. 3, no. 2, pp. 1–15, 2007.
[10] Y.-C. Chen and Y. Li, "Thermal-aware floorplanning via geometric programming," *Mathematical and Computer Modelling*, vol. 51, pp. 927–934, 2010.
[11] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3d ics," in *ICCAD*, 2004.
[12] M. Healy *et al.*, "Multiobjective microarchitectural floorplanning for 2-d and 3-d ics," *Trans. on CAD*, vol. 26, pp. 38–52, 2007.
[13] W.-L. Hung *et al.*, "Interconnect and thermal-aware floorplanning for 3d microprocessors," in *ISQED*, 2006, pp. 98–104.
[14] J. Cong and M. Sarrafzadeh, "Incremental physical design," in *ISPD*, 2000.
[15] P. B. J. Creshaw, M. Sarrafzadeh and P. Prabhakaran, "An incremental floorplanner," in *GLSVLSI*, 1999.
[16] Y. M. X. Li and X. Hong, "A novel thermal optimization flow using incremental floorplanning for 3d ics," in *ASPDAC*, 2009, pp. 347–352.
[17] http://www.opensparc.net/pubs/preszo/07/n2isscc.pdf.
[18] J. L. Ayala *et al.*, "Through Silicon Via-Based Grid for Thermal Control in 3D Chips," in *Nano-Net*, ser. Lecture Notes in Computer Science, vol. 1, no. 1. Springer, 2009, pp. 1–7.
[19] L. B. G. Paci, F. Poletti and P. Marchal, "Exploring temperature-aware design in low-power mpsocs," *IJES International journal of embedded systems*, vol. 3, no. 1, pp. 43–51, 2007.
[20] "IBM Systems Energy Estimator," http://www.ibm.com/systems/ support/tools/estimator/energy.