INTEGRATION, the VLSI journal 46 (2013) 10–21

Contents lists available at SciVerse ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

3D thermal-aware floorplanner using a MOEA approximation ☆

David Cuesta *, José L. Risco-Martin, José L. Ayala, J. Ignacio Hidalgo

Complutense University Madrid, Spain

ARTICLE INFO

Available online 10 May 2012

Keywords:
3D architecture
Thermal-aware floorplan
Temperature
Reliability
Hot spots
Evolutionary algorithm
Through silicon vias optimization

ABSTRACT

Two of the major concerns in 3D stacked technology are heat removal and power density distribution. In our work, we propose a novel 3D thermal-aware floorplanner. Our contributions include:

1. A novel multi-objective formulation to consider the thermal and performance constraints in the optimization approach.
2. Two efficient *Multi-Objective Evolutionary Algorithm* (*MOEA*) for the representation of the floorplanning model and for the optimization of thermal parameters and wire length.
3. A smooth integration of the MOEA model with an accurate thermal modeling 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 and peak temperature, as well as the thermal gradient, with a reduced overhead in the wire length of the system.

1. Introduction

Computer architectures are experiencing a performance barrier given by Patterson's [1] problem, formulated 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 Moore's 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.

Nowadays heterogeneous many-core systems are being implemented and they present a unique opportunity to improve (sometimes in some orders of magnitude) the performance of the architecture. This is achieved by increasing the high performance algorithms to specifically tailored architectures. Special HPC applications like N-Body Simulations, Molecular Dynamics, and Terrain Rendering [2] can experience order of magnitude or greater speedups when compared with architectures that are specifically tailored to their needs. Similar examples from the HPC community include the Los Alamos National Labs Roadrunner system [3].

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). These systems 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, with 48 cores. [4] It incorporates technologies intended to scale multi-core processors to 100 cores and beyond, such as an on-chip network, advanced power management technologies and support for message-passing.

However, the exponentially increasing power densities that are being reached in current technologies, and taking into account the values of leakage currents, the cooling cost of the systems and the reliability constraints in microprocessor-based systems, are a major problem in terms of temperature. The operating temperature has a significant 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, therefore, performance degradation [5].

Also, leakage power consumption is comparable to dynamic power consumption in sub-micron technologies. Leakage power is highly dependent on temperature and the efficient management of chip heat dissipation will alleviate the effect of the leakage.

Reliability is also strongly related to temperature, and increasing the temperature will exponentially decrease the lifetime of the chip. The time to failure has been shown to be a function of

☆This work has been supported by Spanish Government grants TIN2008-00508 and MEC Consolider Ingenio CSD00C-07-20811 of the Spanish Council of Science and Technology, and partially supported by the Swiss National Science Foundation (SNF) grant 200021-127282.

* Corresponding author. Tel.: +34 625856911.
E-mail addresses: dcuestag@pdi.ucm.es (D. Cuesta), jlrisco@dacya.ucm.es (J.L. Risco-Martin), jayala@fdi.ucm.es (J.L. Ayala), hidalgo@fis.ucm.es (J. Ignacio Hidalgo).

0167-9260/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.vlsi.2012.04.003

$e^{-Ea/kT}$, where Ea is the activation energy of the failure mechanism being accelerated by the increased temperature, 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 catastrophic, impacting on the lifetime. Each 10 degrees rise reduces the life of the component by a 50%.

Taking this important fact into account, it is much more preferable to keep the components as cool as possible for maximum reliability. However, the absolute temperature of the chip is not the only factor that affects performance; moreover, the thermal gradients that appear on the chip surface degrade the system reliability by creating dangerous electro-migrations.

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 characteristic depends on the placement of the functional units in the chip. This placement must take as guidelines the proximity of the functional units to the chip border, and the proximity of other units that behave as thermal sinks or thermal sources.

Thermal-aware floorplanning algorithms are able to even out the temperature of the hardware modules through spreading of the heat dissipation. This aspect of floorplanning 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 improve the performance of the system by reducing interconnect delays and increasing the density of the logic, making the idea of "many-core single-chip" into a reality. This revolutionary idea allows the integration of multiple and disparate technologies, such as radio frequency and mixed signal components, with traditional computing technologies.

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 [6] 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 MOEA formulation and an efficient solver that manages multiple objectives in the minimization problem, as well as considering a many-core heterogeneous single-chip for the 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 *Multi-Objective Evolutionary Algorithm* (*MOEA*) 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.

2. Related work

Floorplanning is one of the most important solutions to address the problem of thermal impact. This impact is thermically analyzed in real microprocessor-based systems in [7] where the placement of components for Alpha and Pentium Pro is evaluated. Some initial works on thermal aware floorplanning [8] 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 [9].

In the area of floorplanning for microprocessor-based systems, some authors consider the problem at the microarchitectural level [10], where it is shown that significant peak temperature reduction can be achieved by managing lateral heat spreading through floorplanning. Other works [11] use genetic algorithms to demonstrate how to decrease the peak temperature while generating floorplans with area comparable to that achieved by traditional techniques. Han and Koren [12] use 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 floorplanner will optimize 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 [13] 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. [14] proposed a thermal-driven floorplanning algorithm for 3D ICs, which is a natural extension of his previous work on 2D. In [15], Healy et al. implemented a multi-objective floorplanning algorithm for 2D and 3D ICs, combining linear programming and simulated annealing. Some other authors [16] have also considered the placement of thermal vias in these 3D stacks to optimize the thermal profile of ICs.

Our work has more similarities with the reference [17] carried out by Hung, where a thermal-aware floorplanner for 3D architectures is proposed. However, this study does not consider the important fact that the problem must be considered as a multi-objective problem, as we propose in our work, and it 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 tradeoff 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 [18–21] have been proposed, but none has been proposed, and none has taken thermal-aware 3D IC design into consideration. Li and Hong [22] have 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 work, we propose a novel algorithm to thermally optimize the 3D layout. It will eliminate hotspots, reduce the peak and mean temperature and decrease the reliability risks associated with temperature. Given a 3D package and a chip area, we study the thermal-aware problem and we formulate the total wire length problem with two different genetic algorithms. The first one places the functional units throughout the chip according to their power densities trying to minimize temperature parameters. The second algorithms places the communication units in strategic points in order to minimize wire length. Experimental results show that we can reduce the maximum on-chip temperature in 80° in the best case, for two realistic homogeneous and heterogeneous many-core single-chip architectures, outperforming previous thermal-aware floorplan designs.

3. Floorplanner

The optimization phase of our floorplanner is carried out by a Multi-Objective Evolutionary Algorithm (MOEA). We will briefly explain in the following the main characteristics of a MOEA algorithm and why it is the best strategy to solve the floorplan optimization.

Most of the algorithms presented for the 3D thermal aware floorplanning problem are based on a Mixed Integer Linear Program (MILP) [15,23], Simulated Annealing (SA) [14,15] or Genetic Algorithm (GA) [24]. MILP has proven to be an efficient solution. However, when MILP is used for thermal aware floorplanning, the (linear) thermal model must be added to the topological relations and the resultant algorithm becomes too complex [25], specially as the problem size (number of cores, in our case) increases. Regarding SA and GA, the main problem is based on the representation of the solution. Some common representations are polish notation [26], combined bucket array [14] and O-tree [24]. Most of these representations do not perform well, because they were initially developed to reduce area. However our problem is not focused on area optimization because in 3D MPSoCs every layer must have the same area. In the thermal aware floorplanning problem, hottest elements must be placed as far as possible in the 3DIC. In this work, we have developed a straightforward MOEA based on NSGA-II [27], which tries to minimize maximum temperature and total wire length while still fulfills all the topological constraints.

MOEAs are stochastic optimization heuristics where the exploration of the solution space of a certain problem is carried out by imitating the population genetics stated in Darwin's theory of evolution. Selection, crossover and mutation operators, derived directly from natural evolution mechanisms, are applied to a population of solutions, thus favoring the birth and survival of the best solutions. These steps can be seen in Fig. 1. MOEAs have been successfully applied to many NP-hard combinatorial optimization problems. MOEAs encode potential solutions (individuals) to a problem with strings (chromosomes), and combine their codes and, hence, their properties. In order to apply MOEAs to a problem, a genetic representation of each individual has first to be found. Furthermore, an initial population has to be created, as well as defining a cost function to measure the fitness of each solution.

As a second step, we need to design the genetic operators that will allow us to produce a new population of thermal-aware floorplaning solutions from a previous one, by capturing the interdependencies of the different topological constraints working concurrently. Then, by iteratively applying the genetic operators to the current population, the fitness of the best individuals in the population converges to targeted solutions, according to the metric/s to be optimized and the weight of each of them. For an overview of MOEAs the reader is referred to [28].

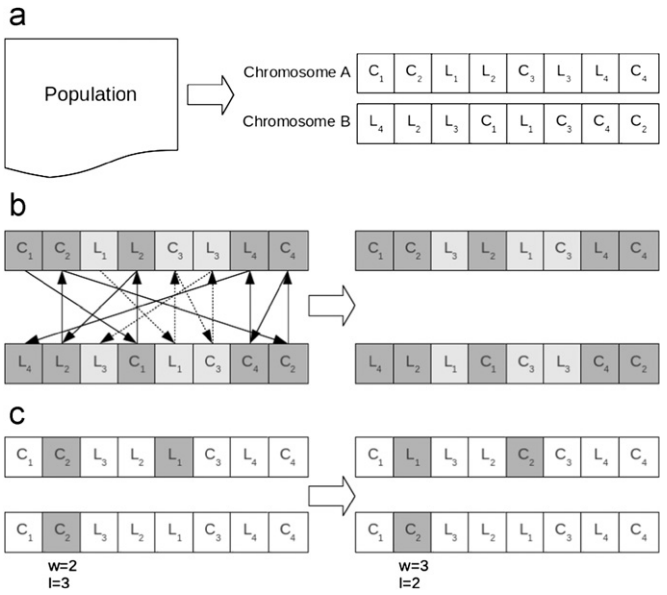

Fig. 1. MOEA operators: tournament selection, cycle crossover and two mutation operators (swap or resize). (a) Tournament selection, (b) cycle crossover and (c) swap mutation or rotation.

4. Genetic representation and operators

The chip is split into small blocks as we will see then in Section 5. Every block i in the model $B_i(i=1,2,\ldots,n)$ is characterized by a width w_i, a height h_i and a length l_i while the design volume has a maximum width W, maximum height H, and maximum length L. We define the vector (x_i,y_i,z_i) as the geometrical location of block B_i, where $0 \le x_i \le L-l_i$, $0 \le y_i \le W-w_i$, $0 \le z_i \le L-h_i$. We use (x_i,y_i,z_i) to denote the left-bottom-back coordinate of block B_i while we assume that the coordinate of left-bottom-back corner of the resultant IC is (0, 0, 0).

In order to apply a MOEA correctly we need to define a genetic representation of the design space of all the possible floorplanning alternatives. Moreover, to be able to apply the NSGA-II optimization process and cover all possible interdependencies of the topological constraints, we must guarantee that all the chromosomes, which are the codification of our final floorplan, represent real and feasible solutions to the problem and ensure that the search space is covered in a continuous and optimal way. To this end, we use a permutation encoding [28], where every chromosome is a string of labels, that represents a position in a sequence. Fig. 1 depicts the three genetic operators used in our MOEA on a floorplanning problem. A chromosome in Fig. 1 is formed by four cores C_i ($i=1, 2, 3, 4$) and four memories L_i ($i=1, 2, 3,4$).

In every cycle of the optimization process (called generation) two chromosomes are selected by tournament (Fig. 1(a). To this end, we select two random chromosomes from the whole population and we select the best of these. This task is repeated twice in order to obtain two chromosomes (called parents). Next, as Fig. 1(b) depicts, we apply the cycle crossover: starting with the first allele of chromosome A (C_1), we look at the allele at the same position in chromosome B. Next, we go to the position with the same allele in A, and add this allele to the cycle. Then, we repeat the previous step until we arrive at the first allele of A. Finally, we

put the alleles of the cycle in the first child on the positions they have in the first parent, and take next cycle from the second parent. As can be seen in Fig. 1(b), dark squares are unchanged blocks from the parents and gray squares represent when the children are swapped. Finally, mutation can be executed in two different ways, both with the same probability (see Fig. 1(c). As a result, some blocks are chosen and swapped as it happens with C_2 and L_1 in the first child, and others are rotated 90°, as it happens in C_2 in the second child where the length (l) and width (w) of the block are changed.

4.1. Fitness function

Each chromosome represents the order in which the blocks are being placed in the design area. Every block B_i is placed taking into account all the topological constraints, the total wire length, and the maximum temperature in the chip with respect to all the previously placed blocks $B_j : j < i$. In order to place a block i, we take the best point (x_i,y_i,z_i) in the remaining free positions. To select the best point we establish a dominance relation taking into account the following objectives in our multi-objective evaluation.

The first objective is determined by the topological relations among placed blocks. It represents the number of topological constraints violated (no overlapping between placed blocks and current area less or equal than maximum area).

The second objective is the wire length. The wire length is approximated as the Manhattan distance between interconnected blocks.

The third objective is a measure of the thermal impact, based on the power consumption. To compute the thermal impact for every power consumption we cannot use an accurate thermal model, which includes non-linear and differential equations. In a classical thermal model, the temperature of a unitary cell of the chip, depends not only on the power density dissipated by the cell but also on the power density of its neighbors. The first factor refers to the increase in the thermal energy due to the activity of the element, while the second one is related to the diffusion process of heat [29]. Taking this into account, we use the power density of each block as an approximation of its temperature in the steady state. This is a valid approximation because the main term of the temperature of a cell is given by the power dissipated in the cell, the contribution of its neighbors does not change significantly the thermal behavior. This approximation is then considered in the optimization process and many calls to the thermal simulator are saved, improving the performance of the optimization loop and its execution time. Thus, our remaining objectives can be formulated as:

$$J_3 = \sum_{i<j \in 1..n} (dp_i * dp_j)/(d_{ij}) \quad (1)$$

where dp_i is the power density of block i, and d_{ij} is the Euclidean distance between blocks i and j.

4.2. TSV Optimization

The last step of our optimization flow is to minimize the wire length in the process of placing TSVs. Technologically, TSVs can only connect two layers. In our work, we have considered connections from the top layer to any other one. In this way inter-layer communication is centralized in the top layer, because every layer in the stack is connected with the top one. This solution simplifies the wire routing and the technology processes currently available for TSV integration in 3D chips. The placement of the TSVs is optimized by another multi-objective genetic algorithm. The problem of placement TSVs in the remaining free cells requires a previous analysis of free available vertical cells.

Next, we describe the chromosome encoding depicted in Fig. 2. Our first MOEA has already placed the functional units in the first phase, we examine the remaining free cells in the resultant stack and build an array of x-y coordinates of allowed TSVs. Given a 3D IC with N layers, a first region of this array contains the coordinates of TSVs connecting layers *Top* and 1, a second region contains the coordinates of TSVs connecting layers *Top* and 2, and so forth. If the total number of allowed TSVs is M, we next build a chromosome of M0–1 variables. If 1, a TSV is inserted in the corresponding (x,y) position (and it connects the number of layers defined in the corresponding region). In this way, Fig. 2 encodes seven TSVs in four layers (N=4): one TSV connecting layers 4 and 1, two TSVs connecting layers 4 and 2, and four TSVs connecting layers 4 and 3. The corresponding (x,y) coordinates are stored in the array of coordinates. The larger the number of TSVs, the shorter the total wire length.

Using this representation, we run the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [27].

The algorithm returns a set of solutions, considering the number of TSVs and the total wire length. This makes a Pareto front approximation, and it will be the designer who has to select the optimal solution in terms of economic cost and wire length reduction, considering that a minimum number of TSVs must be included in the design in order to fulfill communication constraints. The minimum number of TSVs is calculated considering the communication bandwidth among cores. We have calculated the data that is transferred considering an FM modulation/demodulation application as the one explained in [30]. The maximum number of TSVs is given by the technological parameters of the TSVs and the amount of data that is transferred [31].

5. Thermal model

Once the optimization process has proposed a set of optimized floorplans, these floorplans have to be thermally analyzed by an accurate thermal model which is briefly described in the following.

3D integration consists of placing different active layers with silicon dioxide and joining them with epoxy which works as a glue material. As inter-layer communication is mandatory,

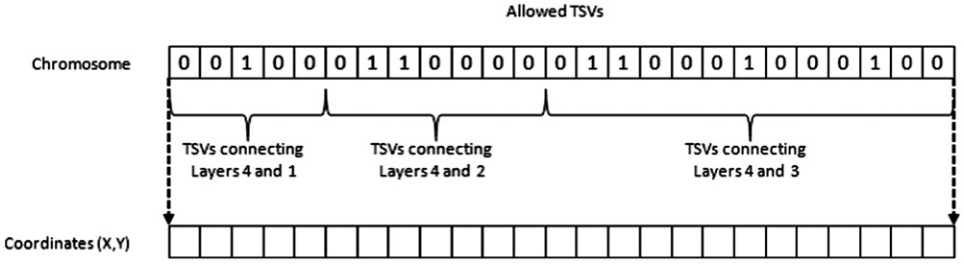

Fig. 2. Chromosome description.

Through Silicon Vias (TSVs) must be deployed in the stack to allow it.

Some of the goals on the design of 3D stacks are to achieve a reduction in area and also to decrease the length of the interconnections, that would be translated into a decrease in the data transfer time and the power consumption.

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 flow through the 3D stack is diffusive, hence, it can be characterized with a 3D RC thermal model as the one presented in [32].

In order to model the thermal distribution in the stack, the chip must be split into small cubic unitary cells. These cells are modeled with six thermal resistances and one thermal capacitance as can be seen in Fig. 3. The resistances in the same layer connect the cubic cell with its neighbors and the remaining two resistances connect the cell with the upper and bottom cell. The capacitance represents self heat storage.

The values of the conductances and the capacitance are calculated using these mathematical expressions:

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

$$G_{north/south} = k_{th}(l \cdot w)/(h/2) \tag{3}$$

$$G_{east/west} = k_{th}(l \cdot w)/(h/2) \tag{4}$$

$$C_{top} = sc_{th}(l \cdot w \cdot h) \tag{5}$$

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

The model also considers the heat diffusion to the surrounding environment. Tuning the resistance and capacitance values of the cells in the border of the stack different chip packages can be modeled.

As the PCB base is considered to be adiabatic, no heat transfer occurs in the bottom direction of the first layer. The vertical diffusion that occurs inside the chip depends on the thermal characteristics of the interface material. This material is modeled as an epoxy layer, a pure resistant material. The existence of TSVs is considered in the model also as a pure resistance element. The most important thermal properties of the material used in the model are listed in Table 1.

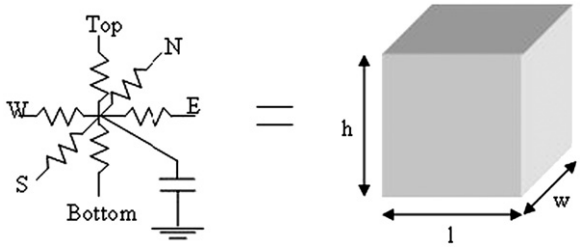

Fig. 3. Equivalent RC circuit of a single cell.

Table 1
Thermal properties of materials.

Si linear thermal conductivity	295 W/(mK)
Si quadratic thermal conductivity	-0.491 W/(mK^2)
SiO_2 thermal conductivity	1.38 W/(μmK)
Si specific heat	1.628×10^6 J/m^3 K
SiO_2 specific heat	4.180×10^6 J/m^3 K

With the values calculated a equation system describing the RC grid is created. After that, an iterative solving method (Forward Euler) is applied.

The functional units can be classified as heat sources or heat sinks. Processor are considered to be strong heat sources because they dissipate power in the die, which is then spread throughout the chip. On the other hand memories, which a have a lower activity and dissipate much less power, can be considered almost as heat sinks. This is an important consideration to be taken into account since the floorplanner will try to place both heat sinks and heat sources as close as possible (provided the routing and performance constraints) to balance the thermal profile.

Once the previous model has been applied to the 3DIC, we obtain mean and peak temperatures, as well as the thermal gradient.

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

6. Experimental Setup

Intel's SCC is the base architecture for our many-core system architecture and the proposed ideas developed in the experimental work. This architecture has been modified to include SPARC cores, like the ones in the Niagara architecture, fabricated in 90 nm and 40 nm technology and whose power density exhibit higher thermal issues than Power cores found in SCC.

The architecture has been also modified in order to include an increased number of cores which are 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 from the top to the other layers.

As we have seen in the previous section, the floorplanner will place the functional units that compose the 3D multi-processor architecture minimizing both, temperature and wire length. The area is set from the beginning of the optimization, and it is the original distribution the one that sets the area of the optimization. The thermal results obtained by our floorplanner will be compared with the stacks composed by these two original layers, based on Niagara 2 and Niagara 3, presented in Fig. 4. These two layers are disposed in order to build a 48, 64 and 128 core system. The cores (C) are disposed in 4, 5 and 9 layers respectively, also the L2 memories (L2), the shared memories (L2B) and the crossbar (Crossbar) can be seen. The original stack with 16 cores in two layers is based on the original Niagara 2 architecture.

Our experimental work will be focused on the analysis of the thermal optimization achieved by the floorplanner in two different scenarios. The first scenario resembles the SCC architecture with a system where 16, 48, 64 and 128 SPARC cores are integrated in the 3D stack.

The second scenario consists of an heterogeneous system where the same number of cores are deployed. The heterogeneous system

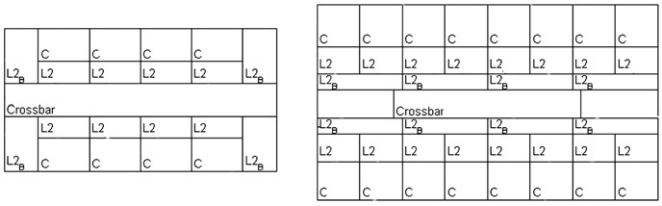

Fig. 4. Original floorplans.

is composed by SPARC and Power6 cores, with a ratio of 3/1. This setup will show the optimized thermal profile that can be expected when multiple core architectures are considered, as well as the extra optimization opportunities that the floorplanner will find.

On the other hand genetic algorithms are configured with different parameters. For the first algorithm, the one in charge of placing the functional units, crossover probability is set to 0.90 and the mutation probability is set to 1/number of blocks as recommended in [27].

The algorithm which optimizes TSVs deployment is configured with a maximum population of 100, and a maximum number of 250 generations. The probability of mutation is set depending on the number of variables; in this particular case, it is the inverse of the number of available points in the plane. Then, we set a single point crossover with a probability of 0.9 and the tournament selection method, following the guidelines given in [27].

7. Results

This section firstly presents the thermal profile of the two scenarios described in Section 6 calculated 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 profiles exhibited by the outputs of the floorplanner.

The worst case operation environment for power consumption in the Niagara 2 (84 W at 1.1 V and 1.4 GHz [33]) and Niagara 3 (139 W) is considered to extract the power densities of every SPARC unit. Also, the area of the layers has been scaled accordingly to the number of cores and the number of layers is also increased. For the Power 6 cores, the power dissipated by the unit is found in [34] and equal to 2.6 W.

7.1. Scenario 1, homogeneous distribution

This first scenario presents the results for 16, 48, 64 and 128 cores for the homogeneous case.

Figs. 6–9 shows the thermal maps for the simulation of the homogeneous system. Black spots in the figures show the position of the TSVs. The previously defined thermal metrics, the mean temperature, thermal gradient and maximum temperature for every layer of the configuration, have been calculated. The comparison with the baseline homogeneous system (48 core original system depicted in Fig. 5) shows that the floorplanner is capable of optimizing the maximum temperature in 60°, the mean temperature in 14 and the thermal gradient is decreased in 75 in the best case.

This can be explained because our floorplanner spaces 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 floorplanner also takes into account vertical heat spread, and each layer will have a different layout, avoiding placing heat sources one over the other. As can be seen in the optimized floorplans the cores are mainly placed in the first and last layer, leaving inner layers with heat sinks. Using this approach heat is spread equally in all the chip achieving big reductions in main and maximum temperature, as well as a decrease in the thermal gradient.

Also, as shown in Table 2, the deviation of temperatures is clearly reduced in both scenarios. This reduction determines a more homogeneous thermal distribution, which is translated into a reduced reliability risk and diminished leakage currents.

7.2. Scenario 2, heterogeneous distribution

The second scenario presents the same results for the heterogeneous case. These results for 16, 48, 64 and 128 cores can be seen in Figs. 10–13 respectively.

Similarly to the previous setup, the mean temperature, thermal gradient and maximum temperature for every layer of

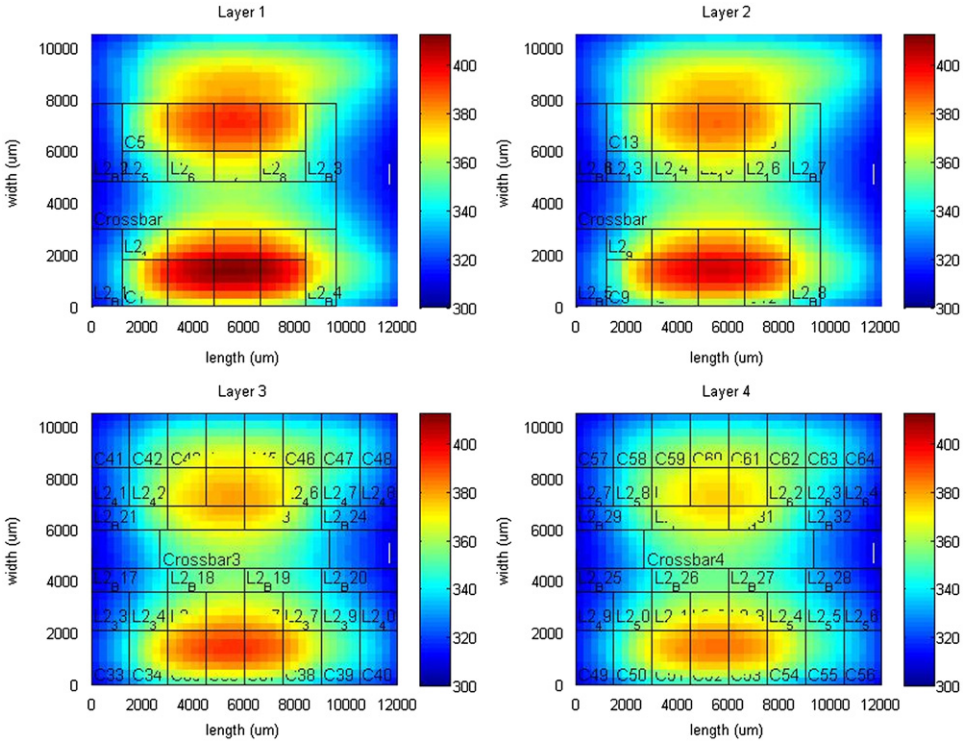

Fig. 5. Thermal maps of the original 48-core homogeneous system.

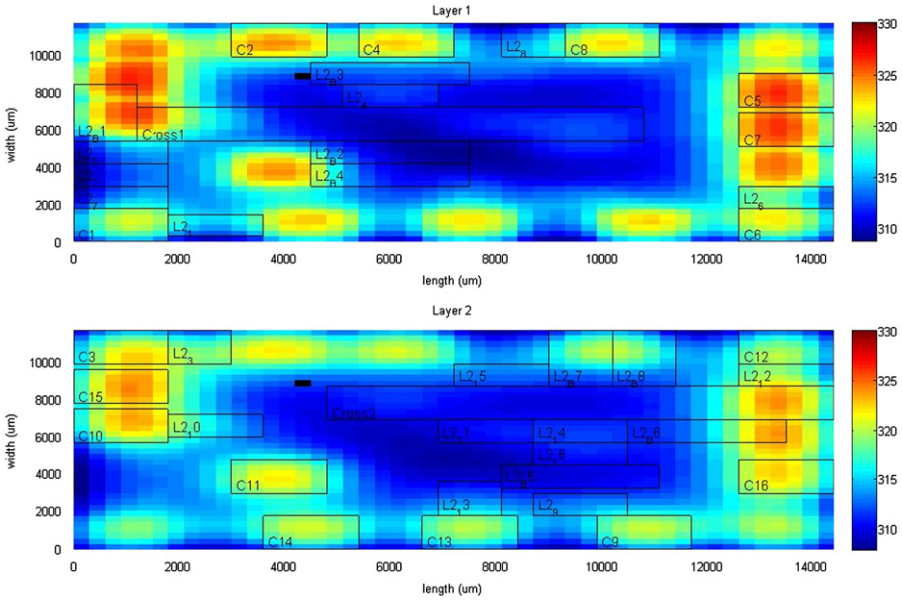

Fig. 6. Thermal maps of the optimized 16-core homogeneous system.

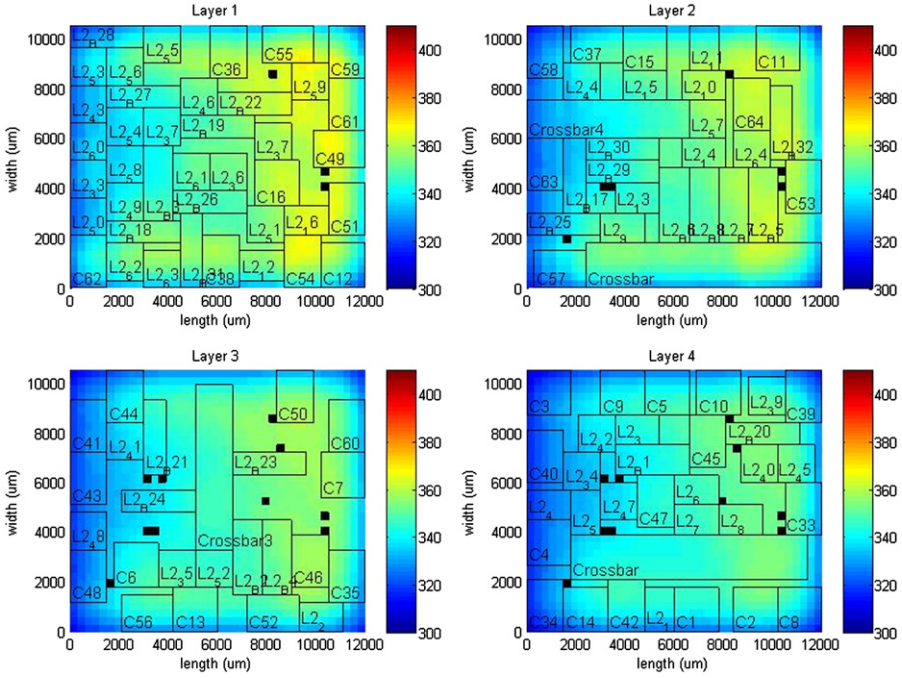

Fig. 7. Thermal maps of the optimized 48-core homogeneous system.

the configuration have been calculated. The comparison with the baseline heterogeneous system shows that the floorplanner is capable of optimizing the maximum temperature in 80°, the mean temperature in 14 and the thermal gradient is decreased in 85, again for the best case. Also, the heterogeneous architecture outperforms the results of the homogeneous system in 21° for the maximum temperature, 14 for the mean temperature and 18 for the thermal gradient in the case of the 128 cores, where the differences are better appreciated.

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 heat sources anymore 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 2, 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.

The optimization of the placement of the TSVs is carried out using a multi-objective genetic algorithm as explained in Section 3.

The algorithm gives the designer a Pareto front approximation with the number of TSVs and chip wire length. As was said, it will

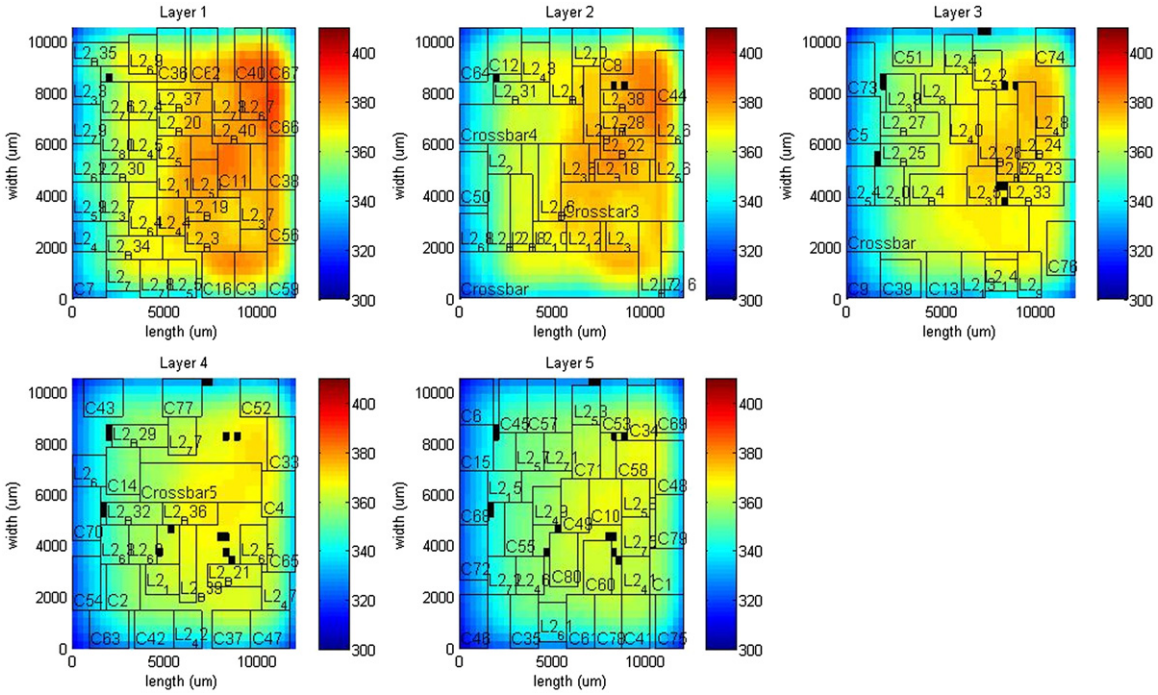

Fig. 8. Thermal maps of the optimized 64-core homogeneous system.

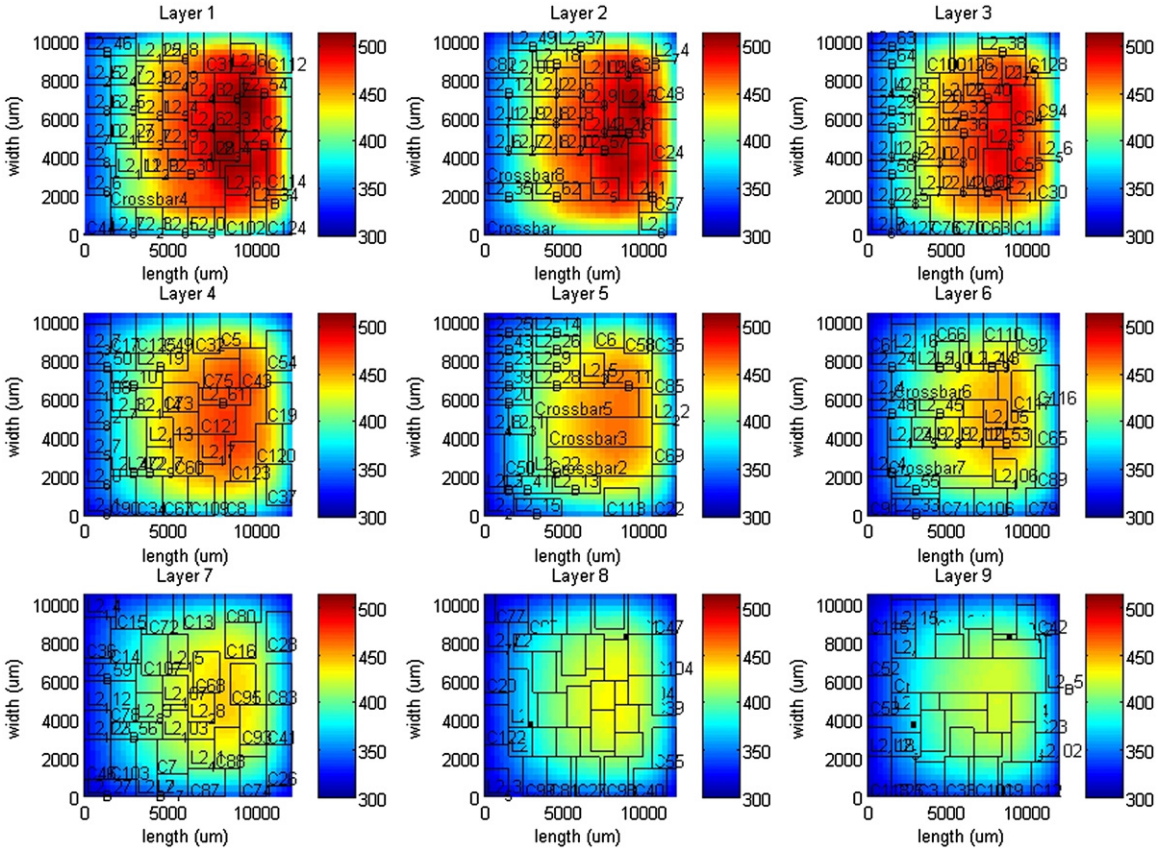

Fig. 9. Thermal maps of the optimized 128-core homogeneous system.

be the designer who will choose which solution is more convenient in every case, depending on the economic cost and technological issues.

Fig. 14(a)–(d) shows the Pareto front approximation for the optimized homogeneous case. As can be seen in the figures, the tendency is like a negative exponential.

Every thermal improvement entails an overhead in the performance of the IC because of communication delays caused by the increase in wire length, however, this overhead in the wiring is not directly translated into an increase in 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 diminished. On the other hand, the big savings reached in the maximum temperature (65 K and 80 K for the Scenario 1 and 2, respectively) justify the overhead in wiring.

Table 2
Thermal deviation (K).

Scenario	16 cores	48 cores	64 cores	128 cores
S.1 Original	15.10	21.58	24.70	38.67
S.1 Floorplanner	11.06	10.72	14.34	41.45
S.2 Original	28.65	18.71	21.22	29.81
S.2 Floorplanner	14.61	8.98	15.41	21.15

Table 3 shows the wirelength associated with the 3D system when the minimum number of needed TSVs is chosen. As can be seen, the overhead incurred by the floorplanner in the worst case scenario (128 cores) has been a 63% when compared to the original homogeneous distribution and 51% for the heterogeneous system.

MOEA optimization is a real fast method for multi-objective problems. Table 4 shows the execution time of the algorithm.

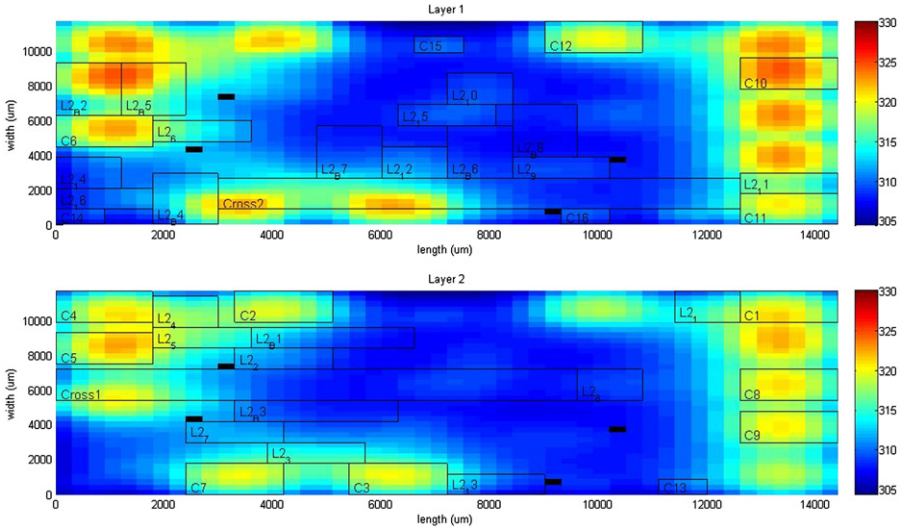

Fig. 10. Thermal maps of the optimized 16-core heterogeneous system.

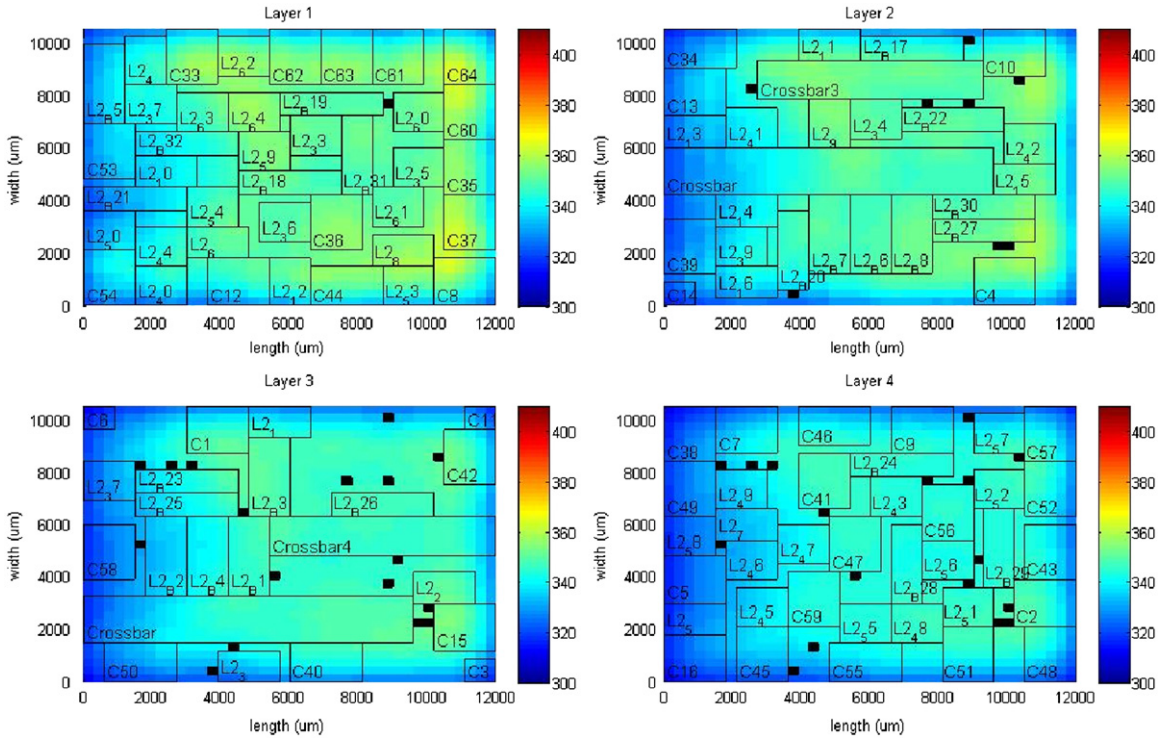

Fig. 11. Thermal maps of the optimized 48-core heterogeneous system.

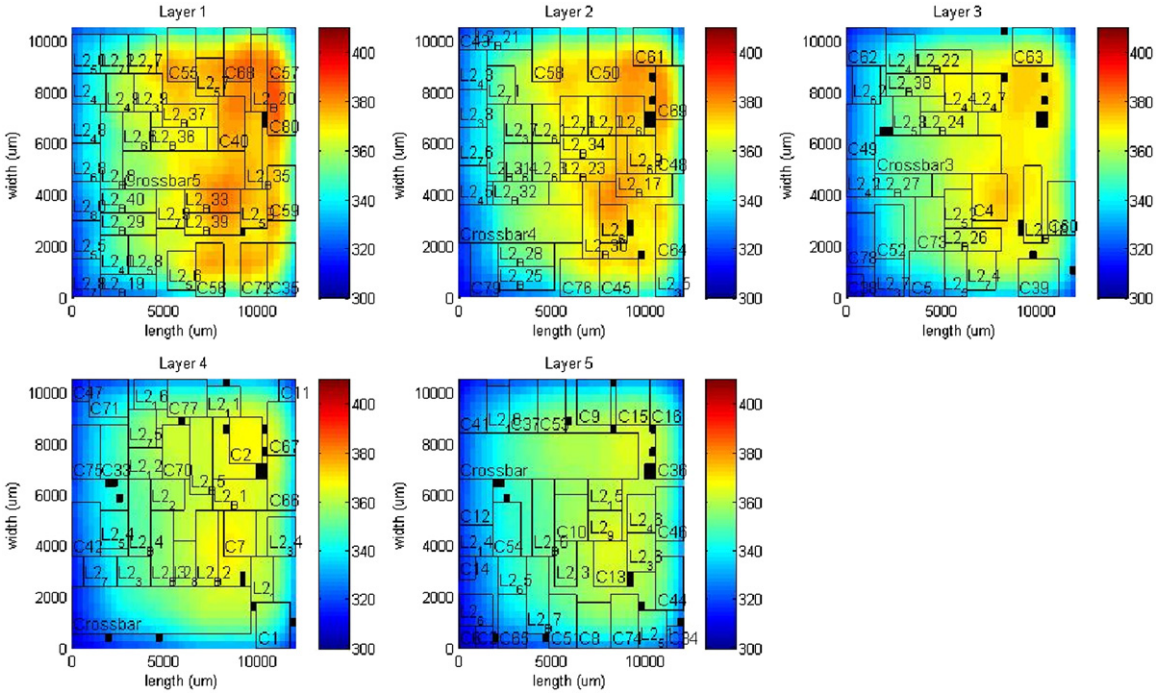

Fig. 12. Thermal maps of the optimized 64-core heterogeneous system.

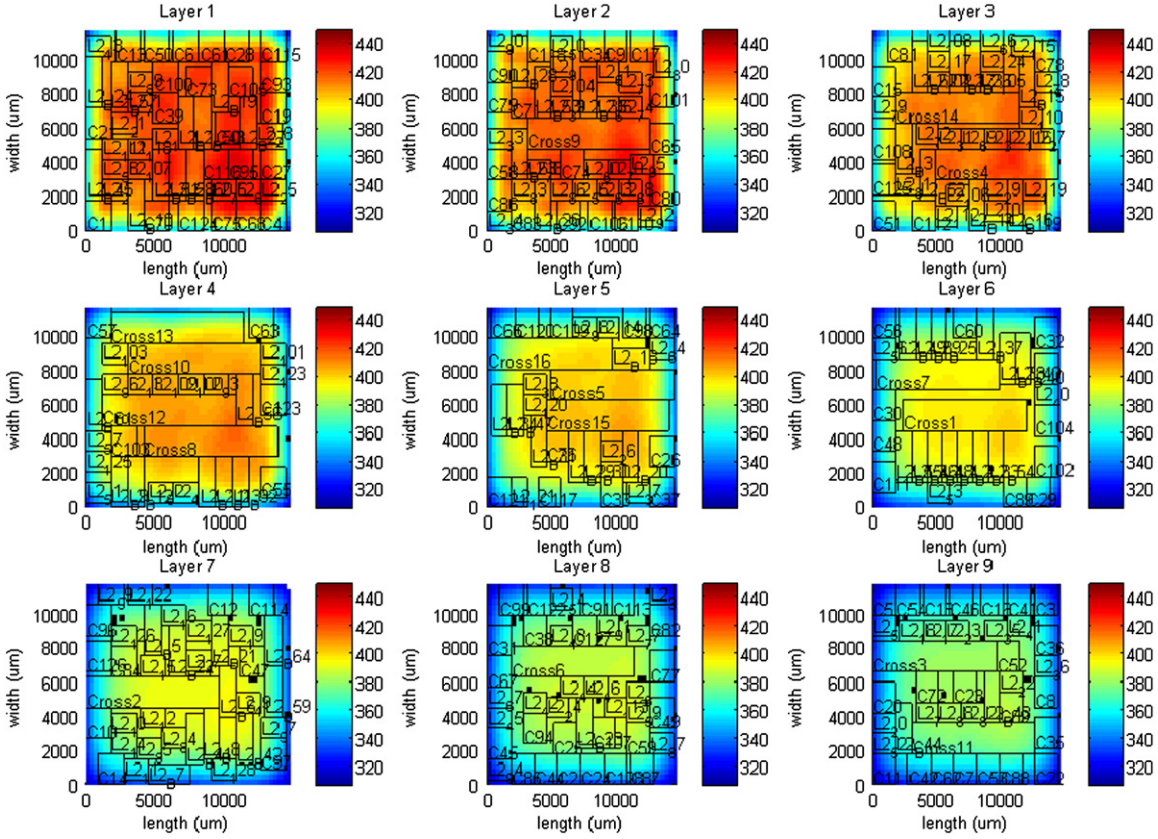

Fig. 13. Thermal maps of the optimized 128-core heterogeneous system.

Execution time is not a critical metric in our work. Since the optimization is done in the design phase of the stack, some time can be invested in order to decrease further costs. Our optimizations have been run in a Intel Core2 Quad CPU Q8300 @ 2.50 GHz.

8. Conclusions

This work has proposed a novel and an effective MOEA formulation to cope with the problem of thermal-aware floorplanning

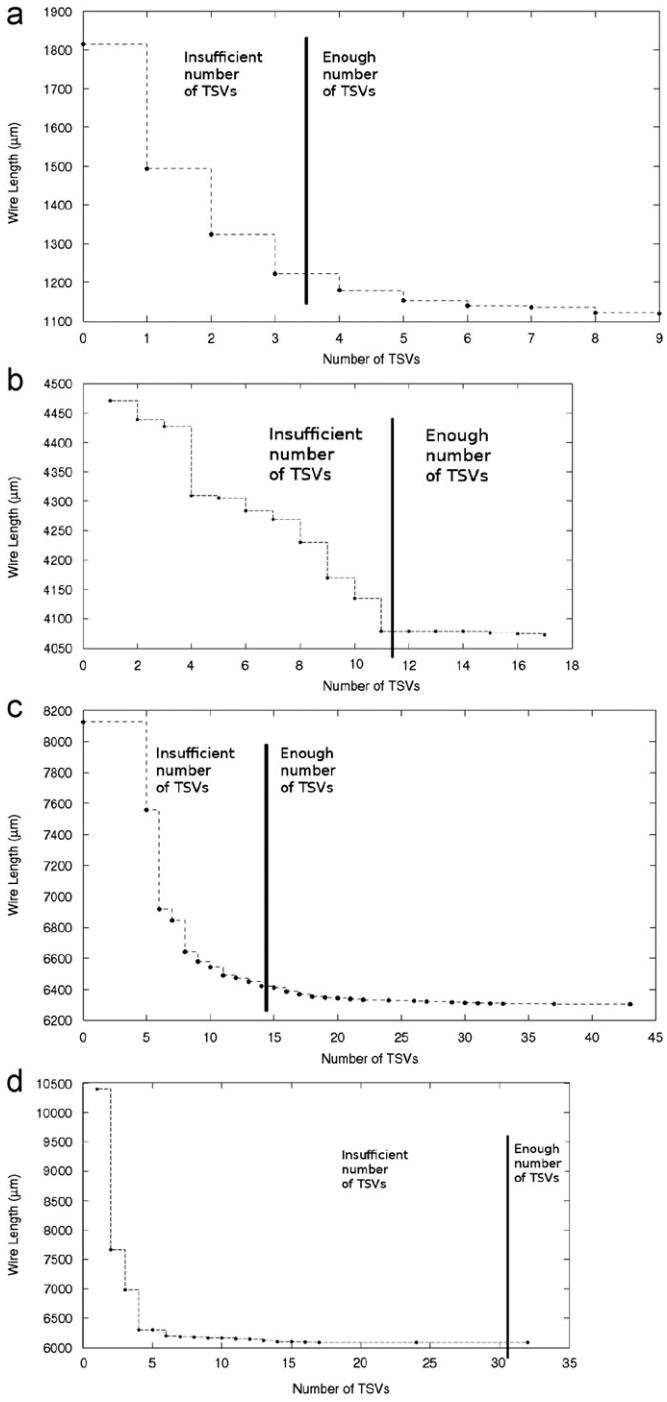

Fig. 14. Pareto front approximation for 16, 48, 64 and 128-core systems. (a) 16-core system. (b) 48-core system. (c) 128-core system.

in 3D many-core single-chips. This floorplanner provides the optimization of the floorplan and it interfaces with an accurate thermal model, which calculates the promising results in the minimization of the main thermal and reliability 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 which used traditional techniques such as MILP formulation or Simulated Annealing.

Table 3
Wire Length (mm).

Scenario	16 cores	48 cores	64 cores	128 cores
S.1 Original	329	794	1481	3314
S.1 Floorplanner	399	1116	1830	5336
S.2 Original	338	796	1481	3314
S.2 Floorplanner	361	1265	1567	5546

Table 4
Execution Time (s).

16 cores	48 cores	64 cores	128 cores
3200	13,000	31,000	258,100

References

[1] Hennessy and Patterson, Computer architecture: a quantitave approach.
[2] V.T.B. Minor, G. Fossum, Terrain Rendering Engine (TRE), Technical Report, IBM, 2005.
[3] Los ALamos HPC Web Page, 〈http://www.lanl.gov/orgs/hpc〉.
[4] Intel Labs, SCC External Architecture Specification (EAS), May 2010.
[5] H. Momose, H. Iwai, Analysis of the temperature dependence of hot-carrier-induced degradation in bipolar transistors for bi-cmos, IEEE Transactions on Electron Devices 41 (1) (1994) 978–987.
[6] D. Cuesta, J. Risco, J. Ayala, 3D thermal-aware floorplanner for many-core single-chip systems, in: IEEE LATW, 2011.
[7] I.K.Y. Han, C. Moritz, Temperature aware floorplanning, in: Workshop on Temperature-Aware Computer Systems, 2005.
[8] C.C.N. Chu, D.F. Wong, A matrix synthesis approach to thermal placement, in: International Symposium on Physical Design, 1997.
[9] G. Chen, S. Sapanetkar, Partition-driven standard cell thermal placement, in: International Symposium on Physical Design, 2003.
[10] K. Sankaranarayanan, S. Velusamy, M.R. Stan, K. Skadron, A case for thermal-aware floorplanning at the microarchitectural level, Journal of Instruction Level Parallelism 7 (2005) 8–16.
[11] W.-L. Hung, Y. Xie, N. Vijaykrishnan, C. Addo-Quaye, T. Theocharides, M.J. Irwin, Thermal-aware floorplanning using genetic algorithms, in: International Symposium on Quality Electronic Design, 2005.
[12] Y. Han, I. Koren, Simulated annealing based temperature aware floorplanning, Journal of Low Power Electronics 3 (2) (2007) 1–15.
[13] Y.-C. Chen, Y. Li, Thermal-aware floorplanning via geometric programming, Mathematical and Computer Modelling 51 (2010) 927–934.
[14] J. Cong, J. Wei, Y. Zhang, A thermal-driven floorplanning algorithm for 3D ics, in: ICCAD, 2004.
[15] M. Healy, et al., Multiobjective microarchitectural floorplanning for 2-D and 3-D ics, Transactions on CAD 26 (2007) 38–52.
[16] E. Wong, S. Lim, 3D floorplanning with thermal vias, in: DATE, 2006, pp. 878–883.
[17] W.-L. Hung, et al., Interconnect and thermal-aware floorplanning for 3D microprocessors, in: ISQED, 2006, pp. 98–104.
[18] J. Cong, M. Sarrafzadeh, Incremental physical design, in: ISPD, 2000.
[19] P.B.J. Creshaw, M. Sarrafzadeh, P. Prabhakaran, An incremental floorplanner, in: GLSVLSI, 1999.
[20] M.L.S. Liao, D. Mehta, Contrained polygon transformations for incremental floorplanning, ACM Transactions on DAES 6 (3) (2001).
[21] R.T.X. Tang, M. Wong, Optimal redistribution of white space for wire length minimization, in: ASP-DAC, 2005.
[22] Y.M.X. Li, X. Hong, A novel thermal optimization flow using incremental floorplanning for 3D ics, in: ASPDAC, 2009, pp. 347–352.
[23] X. Li, Y. Ma, X. Hong, A novel thermal optimization flow using incremental floorplanning for 3D ics, in: Proceedings of the 2009 Asia and South Pacific Design Automation Conference, Piscataway, NJ, USA, IEEE Press, 2009, pp. 347–352.
[24] M. Tang, X. Yao, A memetic algorithm for vlsi floorplanning, IEEE Transactions on Systems, Man, and Cybernetics, Part B 37 (1) (2007) 62–69.
[25] M. Ekpanyapong, M.B. Healy, C.S. Ballapuram, S.K. Lim, H. hsin, S. Lee, Thermal-Aware 3d Microarchitectural Floorplanning, Georgia Institute of Technology Center for, Technical Report, 2004.
[26] J. Berntsson, M. Tang, A slicing structure representation for the multi-layer floorplan layout problem, in: EvoWorkshops, 2004, pp. 188–197.
[27] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[28] C. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer, 2002.
[29] L.B.G. Paci, F. Poletti, P. Marchal, Exploring temperature-aware design in low-power mpsocs, International Journal of Embedded Systems 3 (1) (2007) 43–51.
[30] D. Cuesta, J.L. Ayala, J. Hidalgo, D. Atienza, A. Acquaviva, E. Macii, Adaptive task migration policies for thermal control in mpsocs, in: ISVLSI, 2010, pp. 110–115.
[31] A.W. et al., Bandwidth optimization for through silicon via (TSV) bundles in 3D integrated circuits, in: Design, Automation and Test in Europe, 2009.
[32] J. L. Ayala, et al., Through silicon via-based grid for thermal control in 3D chips, in: Nano-Net, Series Lecture Notes in Computer Science, vol. 1(1), Springer, 2009, pp. 1–7.
[33] 〈http://www.opensparc.net/pubs/preszo/07/n2isscc.pdf〉.
[34] IBM Systems Energy Estimator, 〈http://www.ibm.com/systems/support/tools/estimator/energy〉.

David Cuesta completed MS in Physics and MS in Electrical Engineering from the Complutense University of Madrid, both in 2008. He is currently a Ph.D. student at the same university and his research interests include thermal-aware run-time techniques for multi-processors, thermal simulation and thermal-aware floorplanning.

José L. Risco-Martín is a Assistant Professor at the Computer Architecture and Automation Department of Complutense University of Madrid (UCM), Spain. His research interests focus on computational theory of modeling and simulation, with emphasis on Discrete Event Systems Specification (DEVS), dynamic memory management for embedded systems and evolutionary computation.

Jose L. Ayala got his MS in Telecommunication Engineering and his Ph.D. in Electrical Engineering in 2001 and 2005, respectively, both from the Politecnica University of Madrid. He also has a MS in Physics from the Open University of Spain. Prof. Ayala is currently an Associate Professor in the Complutense University of Madrid (Spain) and a Permanent Visiting Professor at EPFL (Switzerland). His research interests include thermal-aware electronic design and thermal-aware compilation, thermal-modeling and power-efficient computer architectures.?

J. Ignacio Hidalgo is a Associate Professor at the Computer Architecture and Automation Department of Complutense University of Madrid (UCM). His research interests include processor design with special emphasis on the application of novel timing methodologies, memory hierarchy optimization and management, thermal-aware designs, and the application of bio-inspired optimization techniques in CAD problems.