

Physical-Aware System-Level Design for Tiled Hierarchical Chip Multiprocessors

Jordi Cortadella Javier de San Pedro Nikita Nikitin Jordi Petit^{*}
Universitat Politècnica de Catalunya
Barcelona, Spain

ABSTRACT

Tiled hierarchical architectures for Chip Multiprocessors (CMPs) represent a rapid way of building scalable and power-efficient many-core computing systems. At the early stages of the design of a CMP, physical parameters are often ignored and postponed for later design stages. In this work, the importance of physical-aware system-level exploration is investigated, and a strategy for deriving chip floorplans is described. Additionally, wire planning of the on-chip interconnect is performed, as its topology and organization affect the physical layout of the system. Traditional algorithms for floorplanning and wire planning are customized to include physical constraints specific for tiled hierarchical architectures. Over-the-cell routing is used as one of the major area savings strategy. The combination of architectural exploration and physical planning is studied with an example and the impact of the physical aspects on the selection of architectural parameters is evaluated.

Categories and Subject Descriptors: B.7.2 [Integrated circuits]: Design Aids—*placement and routing*

General Terms: Algorithms, Design

Keywords: Network-on-chip, floorplanning, wire planning, chip multiprocessor

1. INTRODUCTION

During the past decade many-core chip multiprocessors have become the major trend in designing scalable computing architectures. Multiple processing units with distributed memory combined with power saving schemes are the platforms used today for exploiting application parallelism while keeping power consumption under control.

Tiled CMP architectures facilitate the design process offering a rapid way of assembling platforms with tens or hun-

^{*}This work has been supported by a gift from Intel Corp., the Spanish Ministry of Science and Innovation (project FORMALISM, TIN2007-66523) and the Catalan Government (SGR 2009-1137).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'13, March 24–27, 2013, Stateline, Nevada, USA.
Copyright 2013 ACM 978-1-4503-1954-6/13/03 ...\$15.00.

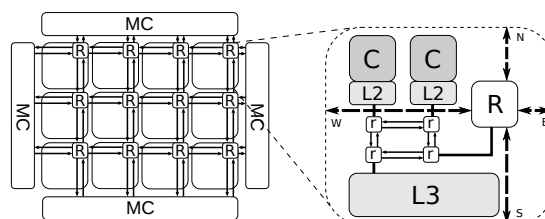


Figure 1: Structural representation of a hierarchical tiled CMP with two-level interconnect: a global mesh and bi-directional rings in clusters.

dreds of units by replicating pre-designed tiles [3, 12, 19]. Nevertheless, large-scale systems obtained by tile replication may suffer from the limited bandwidth of the on-chip interconnect and deliver poor performance because of the communication bottleneck to shared memory. To overcome this problem, *hierarchical* CMP organizations have been proposed to better exploit locality [3, 8].

Figure 1 depicts the block diagram of a tiled hierarchical CMP with 24 cores and distributed L3 cache. The chip is organized as a 4×3 regular grid of tiles (*clusters*), each one including two computing cores (C) with private cache (L2), a distributed shared cache (L3), a router of the global mesh (R) and a local interconnect (bi-directional ring with four routers (r)). The two-level hierarchical interconnect constitutes the backbone of this architecture. The purpose of the global mesh is to provide inter-cluster communication, as well as access to the memory controllers (MC). Intra-cluster communication is supported by low-latency rings that significantly improve the bandwidth of the system given the locality of memory references inherent to the applications.

The problem of *system-level design* for a many-core CMP consists of selecting high-level architectural parameters (e.g., number of cores, size of cache, topology of the interconnect, etc.) so as to maximize system performance for the selected workload and satisfy the design constraints (e.g., area and power). System-level design is performed early in the design cycle. The main complexity of this task is determined by the vast space of potential architectural configurations and the inaccuracy of the models to represent the components of the system and the workload.

To alleviate the problem complexity, most strategies for architectural exploration disregard physical parameters and postpone them to the later stages of design. However, in this paper we show that physical planning has a non-negligible impact on performance and area of a CMP. In this work

we propose methods for *floorplanning* and *wire planning* of tiled hierarchical CMPs and show the impact of physical parameters in the configuration of the architecture.

1.1 Networks-on-chip

Physical planning of a CMP is strictly driven by the organization of its on-chip interconnect. In this section we give a brief overview of the interconnect architecture.

Networks-on-chip (*NoCs*) [7] have been firmly established as a the paradigm to implement on-chip communication for many-core CMPs. In a NoC, each link is a short, point-to-point connection that can be shared among different communication flows. NoCs offer several advantages: they enable higher performance, avoid wire underutilization and facilitate reuse and extendability of designs.

The core component of a NoC is the *router* (or *switch*). Each router implements a crossbar switch in order to commute packets from any input port to any output port. Network links connect either different routers, or routers with the endpoint nodes, such as cores and memories. On-chip communication is realized by means of data packet exchange between the nodes. A source node *injects* a packet into the NoC via the attached router, which forwards the packet to other routers, subject to the established routing policy. The destination router consumes the packet from the network and forwards data to the relevant node.

The network *topology* dictates the arrangement of links, routers and nodes. There are several commonly used topologies, such as meshes, rings, butterflies, fat trees, etc [9]. Hierarchical topologies are also used by combining some of the previous topologies.

1.2 Related work

Floorplanning as a part of the VLSI design flow has been extensively studied for decades. Traditional algorithms often try to minimize a linear combination of area and estimated wire length, and leave actual wire planning to posterior stages in the design process. Hierarchical approaches to floorplanning have already been shown to reduce the algorithm runtime. Quite often hierarchical floorplanning is applied to the design of Systems-on-Chip (*SoCs*), for which every component can be considered as a fixed-size block. These blocks can be generated using fixed-outline floorplanners such as [2], while the system-level floorplanning can be solved using the traditional minimal area techniques such as [6]. In this work, we will instead exploit the regularity of tiled CMPs with hierarchical NoCs.

When floorplanning a CMP, it might also be desirable to optimize factors other than area and wire length. Previous approaches exist that evaluate floorplans based on other qualities such as temperature minimization [20, 15] or power consumption [22], using analytical models. For floorplanning at the system-level, [27] proposes a method that creates tile arrangements which minimize the overall wire length for several 3D NoC topologies.

Adding constraints is also considered in modern floorplanning [24]. These constraints usually restrict valid placement of blocks, e.g. adjacency constraints, distance limits between pairs of blocks, and preplaced objects [28]. However, in this work, we want to satisfy constraints imposed by the CMP interconnect nets. In [26] the authors show how over-simplified models for those constraints (e.g., disregarding pin place-

ment) produces sub-optimal floorplans, but only for classic bus-based interconnects.

1.3 Motivation

The problems of physical planning for CMPs are related to traditional problems in VLSI physical design [21]. CMP floorplanning is similar to classical VLSI floorplanning, while wire planning is more common with global routing. However, there are several aspects inherent to tiled hierarchical CMPs which motivate us to extend existing approaches.

As shown in Fig. 1, the tiled organization of CMPs reduces the floorplanning problem from chip to cluster level. However, the cluster floorplan has to satisfy the property of symmetry in the location of the North/South and East/West ports at the boundaries of the tile. This enables the construction of a full chip by replicating and abutting of tiles.

Floorplanning of the local interconnect introduces another complexity into the design. For example, when considering rings, as in Fig. 1, it is required that the links between the ring routers (r) have *balanced* lengths to guarantee similar hop delays. If the link delays are imbalanced the communication through the ring may have a negative impact on performance.

A special type of constraints, such as *adjacency* or *maximum net delay* constraints are required to prevent certain components be placed far from each other. A typical example may be a core and its L2 cache. Placing a cache far from the core may increase its access delay and result into a significant performance penalty. While adjacency of the two components may appear as a too strict constraint, a weaker requirement of the inter-component distance to be less than one hop will be enough to assure no loss of performance.

An important observation is the recent tendency to design CMPs with wide links. Communication links of the on-chip interconnect may incorporate thousands of wires, aiming at transferring a complete cache line in one cycle. Given the ITRS prediction for minimal wire spacing [1], links of a global mesh can have a width of the order of 10^2 μm , occupying a significant amount of chip area.

One of the possible ways to alleviate the area overhead is to benefit from *over-the-component* routing. Some of the CMP components, such as memories, do not use all the metal layers available in the technology and, therefore, these available resources can be used to implement global nets across the chip.

In this scenario, the most complex components using all metals layers may act as blockages for over-the-component routing. Hence, one of the purposes of wire planning is to verify chip *routability*. Another purpose is the estimation of wire length, which is one of the main parameters when evaluating design quality [23].

The main contribution of this paper is the incorporation of physical planning during the architectural exploration of hierarchical CMPs. The algorithms for floorplanning and wire planning are customized to support constraints for tiled configurations. To demonstrate the viability of this methodology, a case study for exploration is presented and the influence of physical planning on the exploration is evaluated.

2. OVERVIEW

This section gives an overview of the main contributions of the paper by using a simple example. The impact of physical planning on architectural exploration will be illustrated.

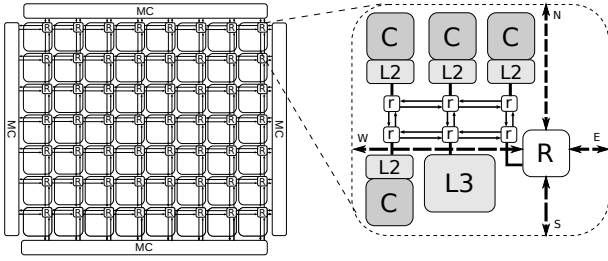


Figure 2: Block diagram of a CMP configuration.

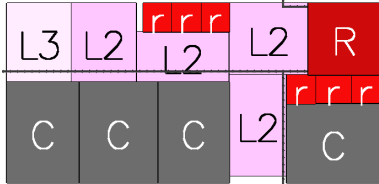


Figure 3: Minimum-area floorplan

Selection of the sample configuration. For this example, let us assume that an architectural exploration tool has generated a configuration such as the one shown in Fig. 2. This configuration has a total of 224 identical cores. The cores are pre-designed and each one occupies an area of 1.2 mm^2 . The layout of the cores can be flipped and rotated.

Each core is assumed to include an internal L1 cache and has an associated private L2 cache. We assume that L2 caches take an area of 1 mm^2 and can be implemented as soft blocks, i.e., various layouts with different aspect ratios are allowed. The L3 cache is shared among all cores (a total of 56 MB for the entire chip) and is also a soft block with the same area as the L2 caches.

Conventional floorplanning. When we consider the floorplan the entire system, we face a problem with about 800 components, including cores, L2 and L3 caches and routers. However, in this work we deal with tiled hierarchical CMPs, which have several proven benefits by enabling a divide-and-conquer design strategy. Floorplanning, placement, routing, and timing closure are processes that can be applied to a single tile while guaranteeing correctness for the global system. For this reason, we will center on the floorplanning of a single tile.

The hierarchical CMP in Fig. 2 has 56 identical clusters interconnected with an 8×7 mesh. Each cluster contains 16 components, including cores, caches and routers. The L3 cache is distributed uniformly among all clusters. The L2 and L3 caches and the router are interconnected with a bidirectional ring. The total area of a cluster is 12.52 mm^2 .

Figure 3 depicts a minimum-area floorplan that could be obtained by a conventional floorplanner such as CompaSS[6]. From the point of view of a hierarchical CMP, this floorplan has some undesirable problems:

1. Some cores are not adjacent to their private L2 caches, potentially increasing the communication latency between them. Similarly, there are long distances between some caches and the corresponding ring routers.

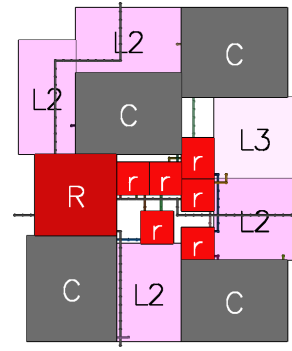


Figure 4: NoC-aware floorplan

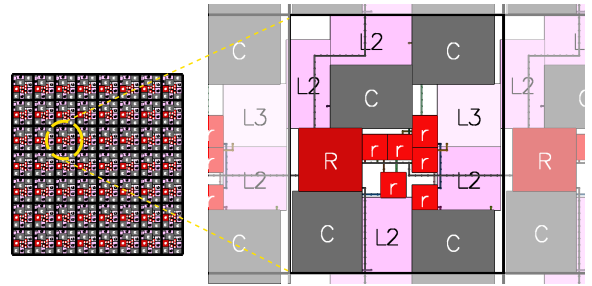


Figure 5: NoC-aware floorplan (full cluster)

2. Ring routers for the local interconnect are not evenly separated. In a ring, the wire length of the longest hop dictates the maximum speed for the entire ring. If this distance is too long, some timing constraints might be violated. Therefore, it is desirable to minimize the length of each link hop separately instead of minimizing the total link length.
3. Assuming that the cores (C) and the router (R) use all metal layers, the two rightmost ring routers (r) have no available routing area in their boundaries. Thus, the design cannot be routed without whitespace insertion.

NoC-aware floorplanning. An alternative floorplan is shown in Fig. 4. This floorplan has been generated using all the constraints and enhancements discussed in this work. Since area minimization is no longer the only objective, this floorplan has a 53% area increase (19.12 mm^2). However, all of the cores are now adjacent to their private L2 caches. Additionally, a route can be found between all the ring routers so that the link length for each hop is always between 0.2 and 0.7 mm, and the distance between a component and its attached ring router is strictly less than 0.4 mm.

As an example, Fig. 5 shows a floorplan for the entire system, including all clusters, based on the cluster floorplan from Fig. 4.

Note that a 53% increase in area may induce an unacceptable overhead in manufacturing cost. This fact may encourage a designer to select an alternative architectural configuration, with a slightly lower performance, although with better floorplan properties.

3. ARCHITECTURAL EXPLORATION

This section overviews the *flow for architectural exploration of CMPs* and introduces the context for physical planning. Consider the problem of maximizing CMP performance (throughput) subject to a resource budget, i.e. constraints on area and power. The given formulation is an example of the architectural exploration problem with the objective of efficiently distributing the chip resources among the components of a multi-core system, e.g. cores, memories and interconnect.

The design space for exploration is specified through a set of models and design constraints. The *models* describe the behavior of individual components. There can be different models for cores characterizing different micro-architectural features that trade-off area, power and performance (in-order/out-of-order execution, multi-threading, etc). The memory models define the size, area and latency of different memory modules. The models for the interconnect topologies define their physical and performance properties (latency, contention, etc).

The expected workload for the CMP requires another type of models that characterize the observable behavior produced by the generated memory patterns (memory locality, burstiness, etc). *Constraints* on power consumption and area are typically defined to confine the design space.

Exploration is a complex optimization problem due to the vast discrete space of architectural variables that determine the configuration of a CMP (e.g. number of cores, cache sizes, interconnect topology, link width). To handle this complexity, in this work we resort to a three-stage divide-and-conquer approach to solve the exploration problem. Figure 6 illustrates our methodology, with the main stages being the *architectural exploration*, *physical planning* and *validation*.

Architectural exploration. During the first stage, analytical models are used to rapidly prune the design space and generate a set of promising configurations in the area/power/performance space. The analytical model from [16] is used to evaluate CMP configurations and discriminate those with poor performance. Static and dynamic power are also evaluated using analytical approximations based on the area and activity of the CMP components. The area is approximated as the sum of the areas of all components on chip.

Analytical models are used as a cost estimator for an iterative metaheuristic-based search to efficiently navigate through the design space. This space is described with a set of architectural variables and a set of *transformations* is defined to explore the neighborhood of any particular configuration. Some examples of transformations include modifying the dimensions of the top-level mesh, the number of cores per cluster or the topology of the local interconnect, among others. *Simulated Annealing* [13] and *Extremal Optimization* [5] are used to explore the design space by probabilistically applying transformations and tracking the best discovered solution.

Physical planning. The objective of this stage is to evaluate wire length and give a more accurate area estimation. The floorplanning and wire planning algorithms at this stage consider physical constraints for individual CMP components, such as the aspect ratio and the number of metal layers. The accuracy in estimation comes at the expense of a higher algorithmic cost, which is however tolerated by

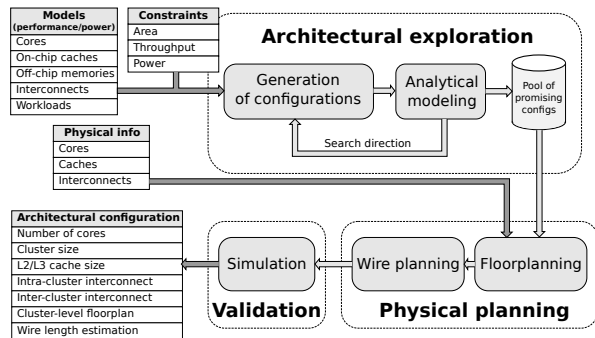


Figure 6: The CMP exploration flow.

performing the planning for a moderate number of configurations, selected during the first stage.

Validation. Finally, the validation phase of the flow is aimed at verifying performance and power, which may differ from the initial analytical estimates. In the current setup we use a cycle-accurate simulation for CMP interconnect, supplied with probabilistic automata models for cores and memories.

This paper focuses on the algorithms for physical planning of hierarchical CMPs. Their objective is to accurately estimate the chip area and wire length, subject to the physical constraints. The methods proposed in this work are applied at the second phase of the described exploration flow.

4. FLOORPLANNING STRATEGY

Floorplanning is the task of defining tentative locations for the blocks of system under certain geometric constraints. The blocks represent pre-designed CMP components such as cores, memories and routers. The blocks can either have a fixed size or accept a set of different *aspect ratios*. The traditional floorplanning problem only considers the minimization of the total area occupied by the components. More advanced floorplanning strategies can also consider the minimization of other metrics such as the estimated wire length.

Because of the complexity of the problem, it is essential to select efficient data structures to represent floorplans. *Slicing trees* [17] are a very popular and compact representation. When combined with compaction, slicing trees are a complete floorplan representation for all non-slicing floorplans [14]. As blocks with multiple aspect ratios are common (e.g., memories), slicing floorplans are very appropriate for CMP floorplanning [29].

In this work, we use Simulated Annealing for the exploration of slicing floorplans similarly as proposed in [25], where the cost function is defined as a linear combination of area and wire length approximated with half-perimeter wire length. In this work we extend this cost function with other components that aim at generating floorplans with some properties and constraints for tiled hierarchical CMPs.

4.1 CMP floorplanning constraints

In Section 1.3 we have discussed some of the requirements for the physical planning of tiled hierarchical CMPs. Next we address them in more detail.

Over-the-cell routing. One important aspect of our approach for floorplanning is that routing is to be done en-

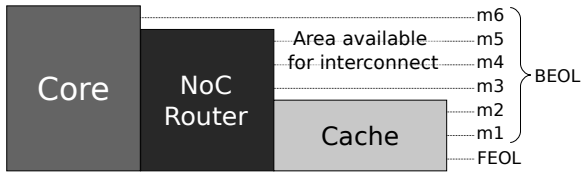


Figure 7: Over-the-cell routing

tirely inside the bounds of the floorplan, using free metal layers on top of placed components (Fig. 7). Because of the prevalence of cache memories in the tiles, we can assume that every configuration can be routed using the available metal layers on top of the components without requiring any extra whitespace. During floorplanning, and as part of the wire length estimation that will be described further in this section, unroutable configurations are discarded.

Abutability. Because only a single tile of a chip is floorplanned, some nets that connect different clusters will have floating terminals that must be placed on one of the boundaries of the tile. However, the placement of this terminal must lie adjacent to the placement of a corresponding terminal on the next cluster. Thus, a special symmetry constraint is created between pairs of nets. All the global interconnect nets have this property.

Wire length constraints. Due to performance reasons, certain critical nets must have a wire length constraint. In case these constraints are violated the floorplan is rejected. This maximum length will depend on the desired interconnect operating frequency, wire sizing and other parameters [1].

Equidistantly-spaced nets. For most interconnect network, the communication delay is determined by the maximum length of a set of links. For example, in a ring, the cycle period must be long enough to allow packets to propagate across the longest of the ring hops. In these cases, it is desirable not to strictly minimize the total wire length, but to balance the individual lengths of the respective links. For this reason, nets that must satisfy this requirements are evaluated differently in the cost function, minimizing the sum of the squares of the lengths instead:

$$WL_{Eq}(\text{FP}) = \sum_{\forall \text{net} \in \text{Ring}} WL(\text{net})^2$$

4.2 Cost function

The final cost function used in the search is as follows:

$$\text{COST}(\text{FP}) = \alpha \text{Area}(\text{FP}) + \beta \text{WL}(\text{FP}) + \gamma \text{WL}_{Eq}(\text{FP}) + P(\text{FP})$$

In this expression, FP is the floorplan being evaluated, *Area* is defined as the effective area of the floorplan, *WL* is the sum of the wire length estimation for each net, and *WL_{Eq}* is the sum of the squares of the estimated wire lengths for nets in the ring interconnect, if any. The goal of *WL_{Eq}* is to penalize floorplans where equidistantly-spaced nets have excessively diverging lengths, as mentioned in the previous section.

The last term, *P*(FP), aggregates all penalties that are applied when a floorplan does not satisfy one of the constraints detailed in this section. The α , β and γ parameters are weights that a designer can use to guide the search

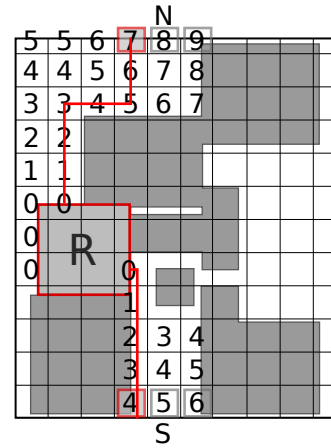


Figure 8: Maze routing a pair of nets with abutability constraint (floorplan from Fig. 4, blockages marked in black).

towards floorplans with smaller area or towards floorplans with smaller wire lengths. An example of this trade-off will be seen in Section 6.3.

4.3 Wire length estimation

A good wire length estimator is important for the evaluation of the cost function. Wire length estimations are used in the *WL*(FP) and *WL_{Eq}*(FP) terms of the cost function. Additionally, it is used to check satisfiability of some of the constraints, such as abutability and wire length limits.

In over-the-cell routing, the only space considered for routing is the free space over the components that have the top metal layers available. Since cores and routers typically implement a complex internal wiring and thus utilize the highest number of layers, memories are the only components in the entire design that leave some metal layers unused. In fact, the relative area of memories in a cluster is defined by the configuration, but it usually ranges between 50%-60% for the best configurations as seen in our tests.

Thus, the lowest metal layers will typically have no space for routing, while the upper layers will have up to 60% of space available, thereby making over-the-cell routing possible. An example can be seen in Fig. 8, which represents a middle metal layer from the floorplan in Fig. 4, with the area occupied by components marked in a dark color.

The work upon this floorplanning algorithm has been based on, [25], proposes the use of the half-perimeter wire length as an estimator. In this work, we propose the use of Lee's algorithm [18], often known as Maze routing. The algorithm is simplified by a) routing links, not individual wires, b) routing each net independently, and c) working over one metal layer only. Thus, routes might be generated that may be found unfeasible during wire planning. However, for the case of nets with two terminals, we can guarantee that a route found using this method is a valid lower bound. Thus, this information can be used to verify wire length and routability constraints. Because of simplification (a), the size of the routing grid is determined by the minimum link width.

The use of Lee's algorithm also enables checking for violations of the abutability requirement. When planning pairs of nets with such requirement, the algorithm will only accept

a path if a matching path has been found on the opposite side for the paired net. The algorithm also will not stop at the first path, but rather collect all paths and select the one where the route is shortest to both opposing extremes of the tile. In Fig. 8, this algorithm is applied to estimate the length of the two vertical mesh links (from the Router to the north side and from the Router to the south). The shortest route for the north net is discarded because at the opposing side of the tile (same column, last row) there has been no path found for the south net.

5. WIRE PLANNING

In order to fully realize the floorplanning estimated in the previous section, we need to establish a wire planning that connects all the required nets between the components and that allows the tiling of the cells. This wire planning must use over-the-cell routing and minimize its wire lengths, while balancing the nets.

This problem corresponds to a routing problem and we solve it in two steps. In the first step, we formulate the routing problem as a Boolean satisfiability problem for which we obtain a feasible solution with a SAT solver. Then, in the second step, we iteratively reduce the wire length of several nets by converting the satisfiability problem to an integer linear programming problem that we solve with an ILP solver. In the following, we describe their essential elements.

Problem formulation. We formulate the routing problem as a Boolean satisfiability problem in the lines of [11], which we extend with some insights that are needed in our context.

The main variables of the SAT problem correspond to the presence (or absence) of a wire segment between two adjacent nodes of the underlying 3D grid. Another set of variables encodes the assignment of wire segments to specific nets. The SAT problem includes several types of constraints:

- Consistency constraints enforce the expected behavior of the variables we have introduced, e.g., if an edge is assigned to a net, then the edge must be occupied by a wire.
- Routability constraints define a legal routing between the components. Basically, these constraints establish that a set of wire segments guarantee the connectivity of all pins of a net. The formulation is similar to the one presented in [11] but extended to handle floating terminals. Our solution is based on the idea that routing must be performed among regions of points that define the endpoints of the nets. These regions are characterized by a set of (not necessarily adjacent nor disjoint) points that may describe the location of a component or the set of all possible locations for a pin. The correctness of our routability constraints is based on Euler’s graph theory.
- Abutability constraints ensure the symmetry between the wires that are used to interconnect tiled cells. These constraints assert that if a wire in the North boundary provides a signal for a net that interconnects adjacent cells, another wire for the same cell must be placed in the same position in the South boundary. Similar relations must also occur in the other direction and for East/West boundaries.

- Optionally, constraints for design rules can be requested in order to fulfill fabric requirements or to reduce running time. One of the typical design rules is to assign one direction to each metal layer.

Generation of a feasible solution. As said, solving the previous satisfiability problem provides a first feasible solution for our wire planning problem (or shows the absence of such a solution!). The results presented in this paper use PicoSAT [4] to solve the SAT model.

Reduction of wire length. Once we have a feasible solution for the wire planning problem, we improve it by reducing its wire length while maintaining its feasibility. Our strategy is iterative, where each iteration consists in ripping out a small set of nets from the feasible solution and reroute them, subject to the previously specified constraints and minimizing the total wire length.

To do so, we convert our Boolean satisfiability problem into an integer linear problem: Boolean variables are transformed in 0/1 variables, Boolean constraints are easily converted to linear inequalities and, the linear function that counts the amount of wire is used as the objective function of the ILP.

Since the above process is applied for a small set of nets at each iteration, the resulting problem is tractable and can be solved with efficient solvers in a moderate amount of time. Note that solving the original problem with all the nets and seeking for the absolute minimum is too slow for the sizes of the problems we are faced to.

The currently implemented iterative process proceeds by just ripping out and rerouting one net at a time, with the exception of the set of nets that interconnect tiled cells, which are ripped out and rerouted in one step. This process is repeated while reductions in the wire length are obtained, favoring the reduction of long nets before the reduction of shorter nets.

The results presented in this paper use Gurobi [10] to solve the wire length optimization models.

6. EXPERIMENTAL RESULTS

In this section we demonstrate the impact of using physical planning during system-level exploration, and also show the need for the use of CMP-specific constraints during physical planning for a proper configuration evaluation.

6.1 Exploration setup

All of the experiments from this section use configurations that were obtained using automated system-level exploration. The parameters of this exploration are described in Table 1. We limit the search to tiled hierarchical CMPs using a mesh as the global interconnect, with the second level interconnect being either a bus or a ring (bi-directional or uni-directional). The number of tiles, the number of cores and the distribution of cores among the tiles are exploration variables. We assume that three different models of cores are available (C1, C2 and C3), with different performance and area characteristics obtained by scaling publicly available data of the Intel Core 2 Duo E6400 processor. We also assume that, while cores and interconnect routers occupy all metal layers, cache memories only use two of them. Therefore, routing can be performed over the cache memories. The operating frequency of the interconnect networks has

Parameter	Value		
Maximum chip area	350 mm ²		
Maximum chip power	350 W		
Interconnect frequency	1.6 GHz		
Global interconnect types	Mesh		
Global mesh dimensions	2×2 to 16×16		
Local interconnect types	Bus, Ring		
Local interconnect sizes	Limited by chip area only		
Memory density	1 mm ² /MB		
Cache latency (per size)	5.0 · <i>CacheSize</i> ^{0.5} cycles		
Off-chip memory latency	100 cycles		
Interconnect link width	10 μm (10 ³ wires×10 nm)		
Available metal layers	m1, m2, m3, m4		
Used by cores	All		
Used by NoC routers	All		
Used by cache memories	m1, m2		
Core types	<i>C1</i>	<i>C2</i>	<i>C3</i>
Core performance (IPC)	1.75	2	2.5
Core area	1 mm ²	1.25 mm ²	2 mm ²
L1 size	64, 96, 128 KB per core		
L2 size	64 KB to 1 MB per core		
L3 size	Up to 100 MB per chip		

Table 1: Parameters for system-level exploration.

been used to define the constraints on the maximum wire length for the links.

To characterize the memory accesses, a model extracted from the SPEC2006 *soplex* benchmark is used. The exploration generates 200 configurations in around 20 minutes. Each configuration is described as a block diagram of components, such as the one shown in Fig. 2. For example, the best configuration from this exploration has 25 clusters connected with a 5 × 5 mesh. Each cluster has a bus as local interconnect, two C2 cores and two C3 cores, along with 1 MB of L2 cache per core. The CMP has a total of 50 MB L3 cache distributed across the 25 clusters. It has an estimated throughput of 107.77 IPC.

6.2 Impact of physical planning

In order to prove how the use of physical planning can significantly alter the results of system-level exploration, we applied our physical planning tool to the 200 configurations found by the exploration. This floorplanning process, if run sequentially, takes 5 hours (an average of 90 seconds per configuration). However, on a machine with multiple cores each of the 200 configurations can be run separately.

The results are shown in Fig. 9. For each configuration, *block area* indicates the sum of the areas from all components. The exploration tool, before physical planning, uses this value as estimator for the expected chip area in order to satisfy the maximum area constraint. In this example, no configuration has a block area larger than 350 mm². *Conventional floorplan* shows a minimal area floorplan obtained without using any of the constraints described in this work (abutability, link length optimization, etc.). On the other hand, *NoC-aware floorplan* depicts the floorplan with minimal area that satisfies these constraints. A dashed line connects the block area data point with the minimal NoC-aware floorplan area for the same configuration.

Despite the fact that all configurations have a block area lower than the limit, a large number exceeds the area limit once physical planning is taken into account. As an example,

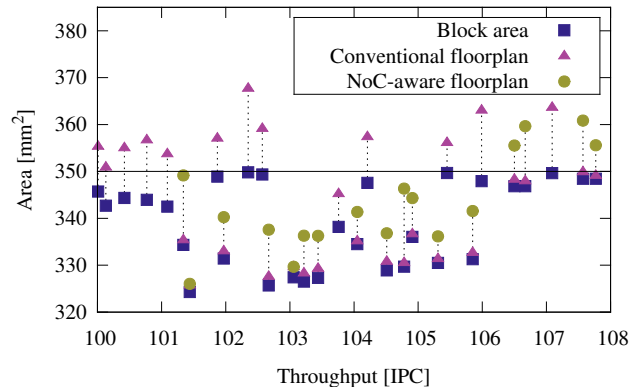


Figure 9: Area for different floorplanning strategies.

the best configuration found by the exploration (rightmost in Fig. 9) has a block area of 348.45 mm², which is below the area constraint. A conventional, minimal area floorplan exists with an area of 349.17 mm², also below the constraint. However, using the tool presented in this work, we find that the smallest floorplan satisfying all floorplanning constraints has an area of 355.59 mm². This violates the area constraint and, therefore, is not actually a valid configuration.

The first viable configuration with area below the limit has a significantly lower performance at 105.85 IPC. Out of the 200 configurations selected during the exploration, 39% of configurations had no floorplan satisfying all the constraints. Even for the configurations for which such a floorplan was found, only 23% satisfy the 350 mm² area limit. Configurations using rings as local interconnect, despite their excellent performance characteristics, have much stricter physical constraints and thus often violate design constraints. Without physical planning, those configurations would have been tagged as “promising” and would have been analyzed with more accurate simulation tools.

6.3 Physical planning search space

A single CMP configuration can have a large number of alternative floorplans. Nevertheless, it is desirable to select one or few candidate floorplans. At the same time, we are considering two metrics by which feasible floorplans can be evaluated: area and wire length. Thus, there is a trade-off.

In Section 2 we showed two candidate floorplans where one had much shorter total wire length at the cost of doubling the chip area. Since this trade-off might be inconvenient for some designs, the weights in the cost function (described in Section 4) can be modified to guide the search towards floorplans with better area or towards shorter wire length.

Figure 10 is an example of the available floorplans for a given CMP configuration. In the chart, each point represents a valid floorplan and its position depends on the area and wire length for that floorplan. By changing the weights in the cost function, a designer can decide which points of the Pareto frontier (solid line) are most desirable.

To illustrate, we selected two representative floorplans from the Pareto frontier that we show in Fig. 11. These are, respectively, the floorplan with the minimal area (but satisfying all constraints) and the overall best floorplan assuming we give the same weights to both area and wire length minimization.

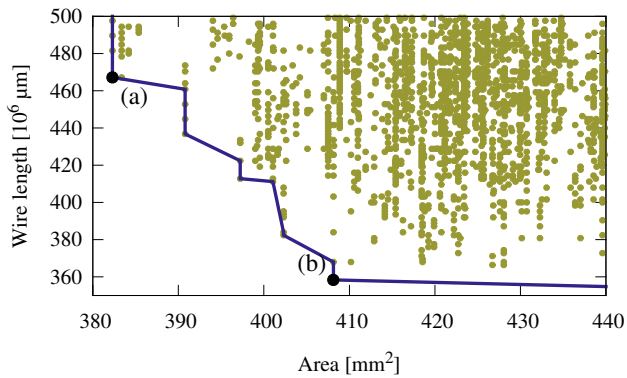


Figure 10: Example of physical planning search space for a single CMP configuration.

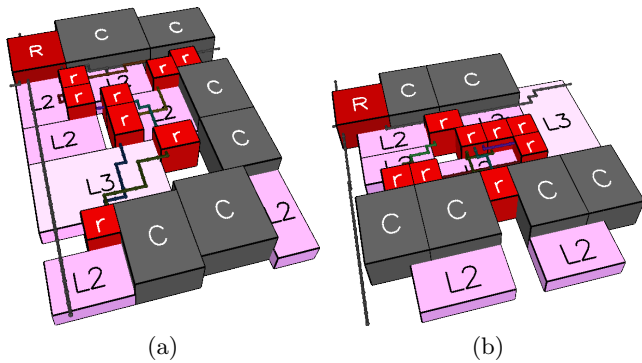


Figure 11: Two design points from the exploration space in Fig. 10.

7. CONCLUSIONS

This work has shown the importance of floorplanning and wire planning during the exploration of CMP architectures. Classical approaches for VLSI physical planning have been extended to support constraints specific for tiled CMPs with hierarchical interconnects. The presence of physical constraints has an important impact in deciding the parameters for the design of CMPs and contributes to guide the exploration towards physically-viable architectures. An interesting and important extension of this work would be to incorporate wire sizing as an additional parameter for exploring area/performance trade-offs.

8. REFERENCES

- [1] International Technology Roadmap for Semiconductors. <http://www.itrs.net/reports.html>.
- [2] S. Adya and I. Markov. Fixed-outline floorplanning: enabling hierarchical design. *IEEE Transactions on VLSI Systems*, 11(6):1120–1135, Dec. 2003.
- [3] J. Balfour and W. J. Dally. Design tradeoffs for tiled CMP on-chip networks. In *Proc. Intl. Conf. on Supercomputing*, pages 187–198, 2006.
- [4] A. Biere. PicoSAT. <http://fmv.jku.at/picosat>.
- [5] S. Boettcher and A. G. Percus. Extremal optimization: Methods derived from co-evolution. In *Proc. of the Genetic and Evolutionary Computation Conf.*, pages 825–832, 1999.
- [6] H. H. Chan and I. L. Markov. Practical slicing and non-slicing block-packing without simulated annealing. In *Proc. of the Great Lakes Symposium on VLSI*, pages 282–287, 2004.
- [7] W. J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *Proc. ACM/IEEE Design Automation Conference*, pages 684–689, 2001.
- [8] R. Das, S. Eacheapati, A. Mishra, V. Narayanan, and C. Das. Design and evaluation of a hierarchical on-chip interconnect for next-generation CMPs. In *High Performance Comp. Arch.*, pages 175–186, Feb. 2009.
- [9] F. Gilibert, F. Silla, M. E. Gomez, M. Lodde, A. Roca, J. Flich, J. Duato, C. Hernández, and S. Rodrigo. *Designing Network On-Chip Architectures in the Nanoscale Era*. CRC Press, 2010.
- [10] Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual. <http://www.gurobi.com>.
- [11] W. Hung, X. Song, T. Kam, L. Cheng, and G. Yang. Routability checking for three-dimensional architectures. *IEEE Transactions on VLSI Systems*, 12(12):1371–1374, Dec. 2004.
- [12] J. Howard et al. A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. *J. Solid-State Circuits*, 46(1):173–183, 2011.
- [13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [14] M. Lai and D. F. Wong. Slicing tree is a complete floorplan representation. In *Proc. Design, Automation and Test in Europe (DATE)*, pages 228–232, 2001.
- [15] M. Monchiero, R. Canal, and A. Gonzalez. Power/performance/thermal design-space exploration for multicore architectures. *Parallel and Distributed Systems*, 19(5):666–681, May 2008.
- [16] N. Nikitin, J. de San Pedro, J. Carmona, and J. Cortadella. Analytical performance modeling of hierarchical interconnect fabrics. In *Proc. ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pages 107–114, May 2012.
- [17] R. H. Otten. Automatic floorplan design. In *Proc. ACM/IEEE Design Automation Conference*, pages 261–267, 1982.
- [18] F. Rubin. The Lee path connection algorithm. *IEEE Trans. Comput.*, 23(9):907–914, Sept. 1974.
- [19] S. Bell et al. TILE64 - processor: A 64-core SoC with mesh interconnect. In *Solid-State Circuits*, pages 88–98, Feb. 2008.
- [20] K. Sankaranarayanan, S. Velusamy, M. Stan, C. L, and K. Skadron. A case for thermal-aware floorplanning at the microarchitectural level. *Journal of ILP*, 7, 2005.
- [21] N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, 1993.
- [22] K. Srinivasan and K. S. Chatha. A low complexity heuristic for design of custom network-on-chip architectures. In *Proc. Design, Automation and Test in Europe (DATE)*, pages 130–135, 2006.
- [23] X. Tang, R. Tian, and M. D. Wong. Minimizing wire length in floorplanning. *IEEE Transactions on Computer-Aided Design*, 25(9):1744–1753, Sept. 2006.
- [24] X. Tang and D. F. Wong. Floorplanning with alignment and performance constraints. In *Proc. ACM/IEEE Design Automation Conference*, pages 848–853, 2002.
- [25] D. F. Wong and C. L. Liu. A new algorithm for floorplan design. In *Proc. ACM/IEEE Design Automation Conference*, pages 101–107, 1986.
- [26] B.-S. Wu and T.-Y. Ho. Bus-pin-aware bus-driven floorplanning. In *Proc. of the Great Lakes Symposium on VLSI*, pages 27–32, 2010.
- [27] T. T. Ye and G. D. Micheli. Physical planning for on-chip multiprocessor networks and switch fabrics. In *Int. Conf. Application-Specific Systems, Architectures, and Processors*, pages 97–107, 2003.
- [28] E. F. Y. Young, C. C. N. Chu, and M. L. Ho. Placement constraints in floorplan design. *IEEE Trans. Very Large Scale Integr. Syst.*, 12(7):735–745, July 2004.
- [29] F. Y. Young and D. F. Wong. How good are slicing floorplans? In *Proc. Int. Symposium on Physical Design*, pages 144–149, 1997.