System Aware Floorplanning for Chip-Package Co-design

Tse-Han Pan and Paul D. Franzon
North Carolina State University
Raleigh, North Carolina
{tpan, paulf}@ncsu.edu

Vaishnav Srinivas, Mahalingam Nagarajan, and Darko Popovic
Qualcomm Technologies, Inc.
San Diego, California
{vaishnav, mahaling, dpopovic}@qti.qualcomm.com

***Abstract*—SoC floorplanning is crucial as it bridges the system design and the physical design of the chip. In this paper, we present a new floorplanning solution based on a novel floorplan model that more closely depicts the design challenges imposed by modern SoC system constraints. Our experimental results demonstrated that this solution is able to create floorplans with hard peripheral instances and soft rectilinear instances with zero white space while supporting system constraints on multiple design instances.**

***Index Terms*—Floorplanning, Chip-package co-design**

I. Introduction

Floorplanning has been a crucial process in modern SoC design. Not only does the resulting floorplan have significant impact on the complexity of subsequent physical design flow, but it also serves as a common blueprint between the system designer and physical design engineers. The floorplan, as a common blueprint, is crucial in design exploration for system designers to determine the design requirements.

The design requirements will result in various design constraints on the floorplan. The optimized floorplan will ideally satisfy on-die constraints such as timing, area, power, and thermal constraints, while also supporting integration for hardened macros. In addition, the floorplan needs to accommodate the system's power delivery network, package routing, and mechanical requirements. In order to optimize floorplan across various objectives, an automated floorplanning tool is desirable for system designers to better assess the impact of system constraints on the floorplan and to adjust the system accordingly in the design planning process.

In this paper, we present a floorplanning flow for system designers to evaluate the system impact on SoC floorplans. The review in prior floorplanning algorithms will be presented in Section II, the formulation and data representation of the problem will be discussed in Section III, an overview of the implementation in Section IV, and the preliminary results in Section V. We will discuss the usage and limitations of this flow, as well as future works in Section VI.

II. Background

In classical floorplanning algorithms, the problem of floorplanning is abstracted into a non-overlapping shape placement problem [1]. The input consists of a set of fixed rectangular design instances along with a connectivity matrix or connectivity graph that captures the number of interconnections between the instances. The typical objective is to minimize the bounding area of all instances and the estimated total wirelength of the interconnects. The abstraction of the problem was later extended by research efforts to allow *soft* rectangles that have constant area but flexible aspect ratio to support fixed rectilinear shapes and incorporate other on-die criteria such as critical-path wire length. Existing successful research on floorplanning includes discovering effective representation of the problem in the context of metaheuristic algorithms such as simulated annealing (SA) [1], and recently, machine learning [2].

However, there are limitations on this problem abstraction in classical floorplanning, as stated by Kahng [3]. The abstraction of classical floorplanning overconstrains the instance shape into rectangles or slicing rectangles; this abstraction also leads to obsession with packing algorithms [1] to minimize the bounding box area taken by the fixed shapes. Although some research includes flexible rectilinear shapes to improve the floorplan [4], this model still deviates from the design requirements of modern SoCs as they are rarely built from hard rectangular design instances. Instead, most of the floorplan area in modern SoCs are taken up by *soft* mixed block and cell design instances that have flexibility in their floorplan shapes. The *hard* design instances are mostly limited to peripheral interfaces and custom designed circuits in the design exploration phase.

To address this problem, Mehta and Sherwani proposed methods to generate rectilinear floorplans on a uniform grid [5], while Kahng proposed the *perfect rectilinear floorplanning* (PRFP) problem formulation to better model modern SoC floorplans. A follow up on this formulation was later published by Feng, Mehta, and Yang [6]. These works better model the on-die constraints of modern SoC floorplanning, but require an initial placement on some [5] or all [6] of its instances.

This work proposes a problem formulation that extends the concept of PRFP to incorporate system constraints into the scope of automated floorplanning algorithms that better model modern SoC floorplanning requirements, along with a novel floorplanning solution to generate system aware floorplans for SoCs. The proposed solution can generate a floorplan

Prof. Franzon would like to acknowledge Qualcomm Incorporated for a Faculty Award.

979-8-3503-1798-5/23/$31.00 © 2023 IEEE

without initial placement, while also effectively preventing the floorplan from violating the geometric constraints derived from package routing requirements and signal integrity (SI) constraints.

III. The Constrained Floorplanning Problem

A. Data Representation

The proposed method in this paper supports rectilinear instance shapes. The rectilinear instance floorplan shape is represented similar to the rectilinear shape in the design exchange format (DEF). Each rectilinear shape is represented as the lower left and upper right coordinates of the sub-rectangles that comprise the shape as shown in Fig. 1. This representation permits arbitrary rectilinear hard macros to be accurately represented with no loss compared to the uniform grid approach.

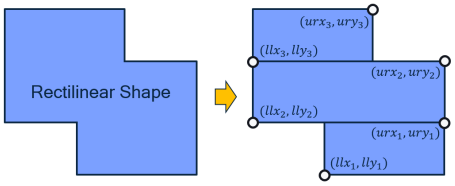


Fig. 1: Data representation of rectilinear shapes.

B. Problem Formulation

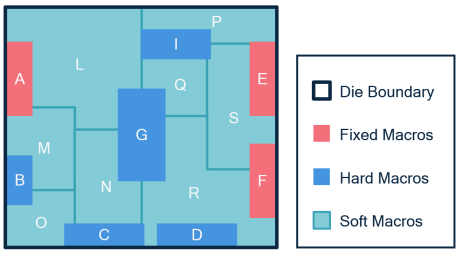


Fig. 2: An SoC example with three instance types.

Given the top level design netlist, the netlist can be represented as a graph $G = (V, E)$ where vertices $V = \{v_i\}$ represent the design instances and edges $E = \{e_j\}$ represent the connection between design instances. Each edge e_j is weighted to reflect the bus width and criticality of the net. The wirelength optimization is dependent on the weight of the edge.

Each design instance v_i is uniquely indexable. The instances are categorized into three types: fixed instances, hard instances, and soft instances as outlined in Fig. 2. Fixed instances have defined rectilinear outlines and fixed locations. Hard instances have a defined rectilinear outlines, but the locations of the instances are not defined and can be moved during the floorplan operation. Soft instances do not have a defined outline or location. During the floorplanning operation, a seed location is used to represent the centroid location of each instance. The rectilinear shape of each soft instance is generated during the floorplan operation.

In addition to the instance types, geometrical constraints can be applied to vertices. The constraints consist of instance attributes (single instance constraints) such as edge affinity, and inter-instance constraints such as minimum placement distance between two instances. The edge affinity constraint assigns instances to one edge of the die, which is useful in constraining the floor placement of peripheral instances. Minimum placement distance ensures the clearance between hard instances for various packaging requirements.

IV. Implementation

The floorplan operation is carried out in two phases as shown in Fig. 3. In phase 1, the centroid location (seed location of design instances) is generated with the force-directed algorithm as an initial state for phase 2. The centroid location is used as the initial instance location to the floorplan engine in phase 2 to generate and refine the constrained floorplan with simulated annealing. Various objective functions are selected and weighted to use as the objective function for SA. The objective of phase 1 is to generate a good initial state for phase 2 where the relative spatial locations of the instances result in lower total wire length. The force-directed solver simulates the interactions between the forces and solves for the minimum potential energy state of the system. In the force-directed solver, the interconnections are modeled as ideal springs which exert attractive spring forces between the centroids of the design instances. In order to spread the centroid location, electric charges are added to instances according to instance area. These charges result in repelling electrostatic forces that spread the design instances.

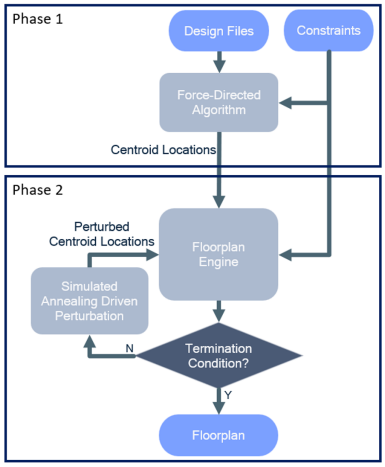


Fig. 3: Overview of the floorplanning process.

Once the initial centroid location is generated, the floorplan engine in phase 2 uses the acquired centroid locations to generate PRFP. In order to generate PRFP, hard instances are treated as blockages while the soft instance shapes are generated with the orthogonal Voronoi shapes, detailed in the work by Wang et. al. [7]. The floorplan is then iteratively refined using simulated annealing. The objective functions for the simulated annealing are composed of a weighted sum

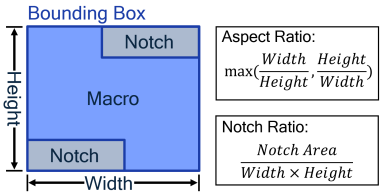


Fig. 4: Quality metrics for rectilinear shapes.

of the total estimated wire length, accuracy of soft instance areas, and soft instance shape quality. The shape quality of soft instances consists of the bounding box aspect ratio and notch ratio as shown in Fig. 4. During the simulated annealing process, one randomly selected instance will be perturbed; a normalized random value will be added to its centroid location and the shape area weight. The objective function of the perturbed solution will be evaluated, and the perturbed solution will either be discarded or used for the next iteration according to the acceptance probability under the current annealing temperature in SA.

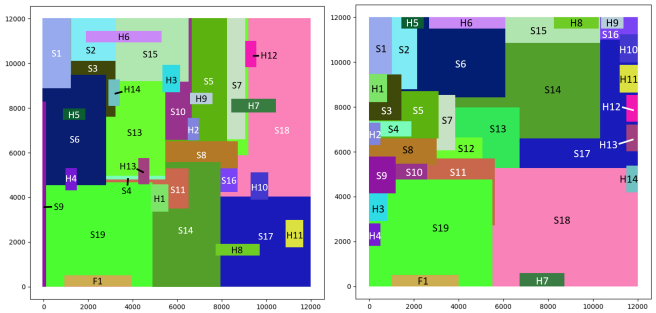
V. Results

The floorplanning flow was evaluated on a mock SoC netlist with 1 fixed instance (F1), 14 hard instances (H1-14), and 19 soft instances (S1-19) as shown in Fig. 5. In the experiment setup for constrained floorplan, edge affinity constraints are applied to all peripheral hard macros, as well as distance constraints derived from SI and packaging constraints. The distance constraints keep all the peripheral macros 500 units away from the corner of the die and prohibit peripheral blocks to be placed within 100 units. Additional placement constraints on H7, H3, and H1 limit the placement of H7 to the right half of the edge, with at least a 4000-unit clearance between H1 and H3. The result is shown in Fig. 5b. An unconstrained experiment is setup for comparison, the unconstrained floorplan is shown in Fig. 5a.

As shown in Fig. 5b the experiment successfully generated a mixed instance type PRFP and the resulting floorplan meets all of the edge affinity and clearance constraints. The resulting rectilinear soft instance shapes have a weighted average bounding box aspect ratio of 1.43 and an average notched area ratio of 0.49. The wirelength of the floorplan averages to be 18% of the die height. The constraints reduces the average package trace length by 4353 units, which is 36% of the die height, compared to the unconstrained floorplan.

VI. Conclusions and Discussions

The system aware floorplanning flow incorporates the system design constraints into the floorplanning problem. This problem statement, as an extension of PRFP proposed by Kahng [3], is a step forward to better capture the practical aspects of modern SoC floorplanning. This flow overcomes the need for manual pre-placement for hard macros, which is a requirement in prior works [5] [6], and leverages work in computer graphics [7] in conjunction with simulated annealing to generate quality rectilinear shapes with a low number



(a) Unconstrained Floorplan. (b) Constrained Floorplan.

Fig. 5: Floorplan Results.

of sides, for soft floorplan instances. This flow successfully generated mixed instance type PRFP with zero dead space that satisfies system constraints on peripheral instance placements. Furthermore, each generated rectilinear floorplan shape maintains a low bounding box aspect ratio and a low notched area ratio. However, the average wire length of the design can be further improved. One potential solution to this problem is to better tune the parameters in phase 1 and to adapt the simulated annealing algorithm in phase 2 of the flow with multi-objective optimization techniques to search for the set of floorplans that are close to the Pareto front instead of one floorplan that best matches the weight combination. In addition, there are many design constraints that should be further investigated such as additional instance adjacency requirements, and limitations on power density. These additional design constraints may further improve the thermal design and the power delivery network.

Acknowledgment

Special thanks to Rajeev Jain, Ananth Ramaswamy, Kathir Ramakrishnan, Guoping Xu, and Karen Sahakyan for the guidence and inputs to the project, and Tianda Huang for the assistance in the writing of this paper.

References

[1] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 12, pp. 1518-1524, Dec. 1996, doi: 10.1109/43.552084.

[2] Q. Xu et al., "GoodFloorplan: Graph Convolutional Network and Reinforcement Learning-Based Floorplanning," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 10, pp. 3492-3502, Oct. 2022, doi: 10.1109/TCAD.2021.3131550.

[3] Kahng, Andrew B. "Classical floorplanning harmful?." In Proceedings of the 2000 international symposium on Physical design, pp. 207-213. 2000.

[4] C. C. N. Chu and E. F. Y. Young, "Nonrectangular shaping and sizing of soft modules for floorplan-design improvement," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 1, pp. 71-79, Jan. 2004, doi: 10.1109/TCAD.2003.819896.

[5] Mehta, Dinesh P., and Naveed Sherwani. "On the use of flexible, rectilinear blocks to obtain minimum-area floorplans in mixed block and cell designs." ACM Transactions on Design Automation of Electronic Systems (TODAES) 5, no. 1, pp. 82-97, Jan. 2000.

[6] Feng, Yan, Dinesh P. Mehta, and Hannah Yang. "Constrained" Modern" Floorplanning." In Proceedings of the 2003 international symposium on Physical design, pp. 128-135. 2003.

[7] Wang, Yan-Chao, Feng Lin, and Hock-Soon Seah. "Orthogonal voronoi diagram and treemap." arXiv preprint, 2019. arXiv:1904.02348.