Dragon2005: Large-Scale Mixed-size Placement Tool

Taraneh Taghavi
CS Dept.,
UCLA
LA, CA
taghavi@cs.ucla.edu

Xiaojian Yang
Synplicity,Inc.
Sunnyvale, CA
xiaojian@gmail.com

Bo-Kyung choi
Magma Design
Automation,Inc.
Santa Clara, CA
bkchoi@gmail.com

ABSTRACT

In this paper, we develop a mixed-size placement tool, Dragon2005, to solve large scale placement problems effectively. A top-down hierarchical approach based on min-cut partitioning and simulated annealing is used to place very large SoC-style designs containing thousands of macro blocks of various sizes and millions of standard cells. Macro aware partitioning and techniques to properly handle different bin sizes are required, because of the existence of large macro blocks. Our tool is also a congestion and timing aware placement tool.

Categories and Subject Descriptors:
B.7.2 Integrated Circuits:Design Aids

General Terms: Algorithms

Keywords: Physical Design, placement

1. INTRODUCTION

With the increase in complexity of IC designs, the traditionally flat approaches to physical design problems will become ineffective [9]. Hierarchical design approachs are used in most state-of-the-art design flows, especially with the extensive reuse of predesigned IP blocks. IP blocks can be treated as macro blocks during the process of physical design. In addition, connections between macro cells and standard cells are becoming tighter with the increasing use of IP blocks. It is becoming harder to separate IP blocks and standard cells and place them into different partitions. Therefore, the ability to handle macro blocks together with standard cells is becoming indispensable for physical design tools.

In this paper, we propose a methodology to handle the problem of mixed-size placement, where standard cells and macro blocks of various sizes have to be placed simultaneously, optimizing a certain objective such as total wirelength or routability. A Min-cut based top-down approach is taken to handle the large complexity of industrial designs, and simulated annealing is used to optimize the total wirelength. At each step of the placement flow, techniques to resolve the problems caused by the presence of macro cells are discussed. Some other functionalities like delay-budgeting assignment for timing-driven placement and also white space allocation for routability-driven placement are also embedded in our placement tool.

2. OVERVIEW OF DRAGON2005

A typical top-down hierarchical placement approach can be generalized as follows: at a given hierarchical level, the layout area is partitioned into several global bins. All the cells of the circuit are distributed into these global bins to minimize a certain placement objective. This cell distribution problem is called a hierarchical placement problem. If a cell is distributed into a particular global bin, it will be placed within the area of this bin in the final layout. As we proceed to more refined levels, the number of global bins increases and the physical size of global bins decreases. Thus we can get more and more detailed information about physical locations of cells as we proceed. The top-down approach terminates when there are only a few cells in each global bin. Dragon2005 based on Dragon2000 developed and presented in [2], is divided into two phases, global placement (GP) and detailed placement (DP). A top-down hierarchical approach is used in the GP phase. We recursively solve the hierarchical placement problem and quadrisect each global bin bins at each level. Overlap between cells are allowed in the GP phase. The DP phase takes the output from GP and produces an overlap free layout. Then it iteratively improves the legal layout using a greedy heuristic. Due to the computational complexity, the DP heuristic is only capable of performing optimization locally. Thus it is expected that the top-down hierarchical GP phase should finish the majority of work in placement. Wirelength and net-cut are two popularly used objectives in different hierarchical placement algorithms.

In this work, we propose hierarchical techniques to place large-scale mixed size designs that may contain thousand of macro blocks and millions of standard cells. Min-cut based top-down approach is taken to handle the large complexity of designs and simulated annealing is used to minimize the total wirelength. Min-cut partitioning should be aware of large macro cells and may result in different sized bins. During simulated annealing, different bin sizes have to be considered. The techniques discussed in this paper can be easily incorporated into any hierarchical placement flow and effectively produce legal final layouts with a short runtime.

3. FRAMEWORK OF OUR PLACEMENT TOOL

Figure 1 shows our placement flow given in [3]. The circuit is recursively partitioned alternatively along horizontal and vertical cut lines. The subcircuits after partitioning are assigned to rectangular bins. At some points a bin-based simulated annealing where the objects that are moved are the subcircuits in the bins, is performed to improve the current placement. Such a procedure terminates when a certain stop criteria (e.g. average number of cells per bin is less than a given number) is met. An adjustment step is then executed to fit the current bin-based placement into row structures. The next step is a cell-based simulated annealing. The bin structure still exists and the cells are moved between the centers of bins. The locations of these centers can be changed during the annealing pro-

ISPD'05, April 3–3,2006,San Francisco,California, USA.
ACM 1-59593-021-3/05/0004.

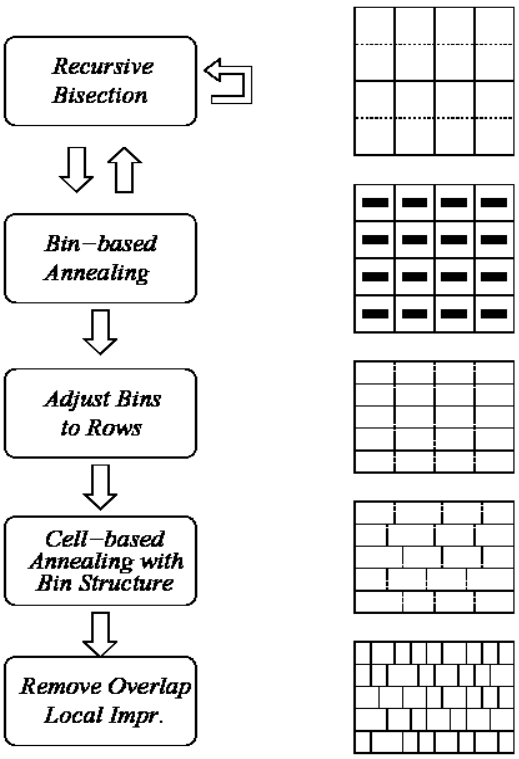

Figure 1: Overall flow of placement tool

cedure. The final step simply spreads overlapped cells and makes local improvements to obtain the detailed placement.

3.1 Partitioning

To handle the high complexity of the problem, the input netlist is recursively divided into two partitions using a state-of-the-art min-cut partitioner, hMetis [7]. Two things have to be considered during partitioning. One is the number of cuts across the partitions and the other is the balance in the sizes of two partitioned sets. hMetis is shown to be able to get very good solutions in terms of both the cutsize and the balance [6].

3.2 Simulated Annealing

A weakness of pure min-cut type placement is its irreversibility. Once a cell is assigned to one side of the cut line, it will never move to the other side to improve the placement. Combining simulated annealing in this flow helps placements move out of the local minima. We use multilevel simulated annealing in this placement flow. The key idea is to reduce the number of movable objectives in annealing. The difference between our flow and hierarchical annealing is instead of using a single cooling schedule through three hierarchical levels, we use low temperature annealing at each level and do not fix the number of levels. Moreover, we avoid using simulated annealing at the final placement stage and use a fast greedy improvement instead. Both bin annealing and cell annealing use total wirelength as the cost function. Also they adopt the same cooling schedule. Swapping is the main move in both types of annealing, and shifting is used a little bit in cell annealing. The disadvantage of simulated annealing is its expensive runtime cost. Although our flow tries to reduce this cost by bin-based approach, annealing is still the most time consuming part.

4. MIXED-SIZE PLACEMENT

To build a mixed-size placement tool that can handle macro cells as well as standard cells, we follow the basic flow a min-cut and simulated annealing based placer [2], that is believed to be very successful [8].

4.1 Macro-Aware Partitioning

Min-cut based hierarchical approaches run into trouble in mixed-size placement, when there is a large macro cell that is bigger than the bin size at a certain hierarchical level. Figure 2 illustrates this problem. In Figure 2 (a), we are trying to vertically cut the bin. The size of both sub-bins have to be equal to have a regular bin structure. However, the macro is too large to fit into any of the sub-bins, even though the actual area of the macro is equal to the half of the bin being cut. Since each cell has to be assigned to only one bin, we have to put the macro either into the left or into the right sub-bin. If we put the macro into one sub-bin and the rest standard cells into the other sub-bin, the resulting layout will be extremely illegal. Figure 2 (b) shows a possible placement solution, which can never be obtained by traditional min-cut based approaches.

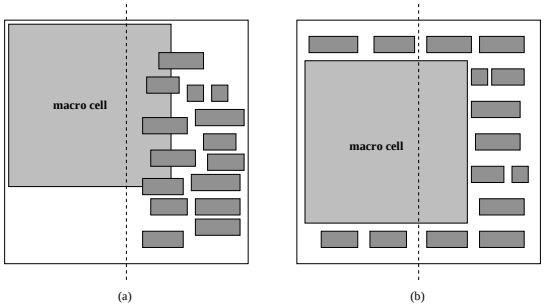

Figure 2: (a) a macro cell is too large to fit into a sub-bin. (b) possible placement solution for the bin, which can never be obtained by traditional approaches.

In order to deal with macro cells [1], we must give up the regularity of bin structure, so bins can have different sizes. For example, we vertically partition the bin in Figure 3.

If there are more than one macros in the bin being partitioned, we pre-assign macros so that they can fit in the sub-bins they will belong to, and perform partitioning for the rest of standard cells. If a macro cannot fit into any of the sub-bins, it is preassigned to a sub-bin that minimizes the violation. When a bin contains only one cell/macro, the bin is no longer partitioned but still can move around during simulated annealing to minimize wirelength.

4.2 Bin-Based Simulated Annealing

After each bipartition, bin based simulated annealing is done to find a good location for each partition to be placed in, minimizing the total wirelength.

Because the bin structure is irregular due to unbalance partitioning, we have to take care of different bin sizes during simulated annealing [1]. There are three types of moves in bin-based simulated annealing: horizontal switch, vertical switch, and diagonal switch. These moves switch two adjacent bins. If the bin structure is regular, we can freely choose any type of move. However, here we have constraints on these moves. Diagonal switches are allowed only when the two bins have the same size (both width and height), vertical switches are allowed only when the widths of both bins are the same, and horizontal switches are allowed only when the heights are the same.

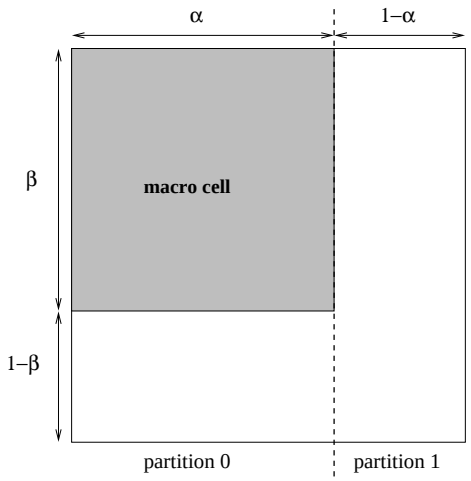

Figure 3: Partitioning when a bin contains a large macro

4.3 Legalization

Once average number of cells in a bin is less than a certain number, recursive partitioning and simulated annealing is halted and we proceed to the detailed placement step.

First, overlaps between cells have to be resolved to get a legal placement. This step is called legalization. Without macro cells, this stage is very simple. All that needs to be done is to place cells next to each other in a row from left to right. However, with the presence of macro cells that spans through multiple rows, the problem is no longer straightforward. Since we have attempted to put macro cells inside bin boundaries during the previous steps, we expect that macro cell will not cause too many problems, but simply placing cells next to each other result in cells outside of the chip boundary. When placing cells to remove overlaps, we have to consider two conflicting factors: the degradation in placement quality and the legality of result. To address this conflict, we use a cost function for placing each cell that combines both factors. For macro cells, the cost function of putting cell c into row r is

$$cost(c,r) = \alpha \cdot (position\ offset) + (1-\alpha) \cdot x_{final}$$

and for standard cells

$$cost(c,r) = \alpha \cdot (wirelength\ change) + (1-\alpha) \cdot x_{final}$$

where $position\ offset$ is the distance from the original position to the final position. $wirelength\ change$ is the change in wirelength caused by moving the cell which can be negative when wirelength decreases. x_{final} is the x-coordinate of the cell after legalization. α is a coefficient to control the importance of each term.

After all overlaps are removed, greedy local improvement is performed. First, we try to switch adjacent standard cells to see if wirelength can be improved. This step will cure the wirelength loss caused by blind cell spreading step of legalization.

In order to handle fixed blockage, set up an array of integer is set up in each row. Each array element represents a site. First, the site array is initialized to all empty. For those sites taken by fixed macros, the site array element is marked as blocked. Cells will not be placed on the blocked sites. For each row, all the bins from left to right are traversed and each cell is attempted to be placed in the bin onto the site array. If there are blocked sites we jump over it and go to the right side of the chunk of blocked sites.

5. WHITE SPACE ALLOCATION FOR CONGESTION HANDLING

The use of white space in fixed-die placement is an effective way to alleviate congested areas in order to improve routability [5]. Our placement tool includes a white space allocation approach that dynamically assigns white space according to the congestion distribution of the placement. In the topdown placement flow, white space is assigned to congested regions using a smooth allocating function. A post allocation optimization step is taken to further improve placement quality. Experimental results show that the proposed allocation approach, combined with a multilevel placement flow, significantly improves placement routability and layout quality.

6. TIMING-DRIVEN PLACEMENT

The other functionality our tool can support is timing-driven placement [4]. A slack assignment approach is used for delay budgeting to make the tool capable of timing-driven placement. Compared to Cadence QPlace, the proposed placement flow generates placement with shorter clock cycle and better routability. Our experimental resutls show that considering design hierarchy is a promising way to handle timing oprimization problem.

7. CONCLUSION

A hierarchical method for placement is proposed in this paper. The proposed methods is based on min-cut partitioning and simulated annealing. We extend our placement tool to handle macro cells so that it can be used to place very large SoC-style designs that may contain thousands of IP blocks of various sizes as well as millions of standard cells. Our tool is also capable of doing timing-driven as well as routability-drive placement for standard-cells.

8. ADDITIONAL AUTHORS

Additional authors: Maogang Wang (Blaze-DFM,Inc., email: `mwa009@yahoo.com`) and Majid Sarrafzadeh (CS Dept.,UCLA email: `majid@cs.ucla.edu`).

9. REFERENCES

[1] B. K. Choi, T. Taghavi and M. Sarrafzadeh. Hierarchical placement for Fixed-Die Mixed-Size Designs Submitted to *Design Automation Conference 2005*.

[2] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Fast standard-cell placement for large circuits. In *International Conference on Computer-Aided Design*, pages 260–263, 2000.

[3] X. Yang, B. K Choi, and M. Sarrafzadeh. A Standard-Cell Placement Tool for Designs with High Row Utilization. In *International Conference on Computer Design*, September 2002.

[4] X. Yang, B. K Choi, and M. Sarrafzadeh. Timing-Driven Placement using Design Hierarchy Guided Constraint Generation. In *International Conference on Computer-Aided Design*, November 2002.

[5] X. Yang, B. K Choi, and M. Sarrafzadeh. Routability-Driven White Space Allocation for Fixed-Die Standard-Cell Placement. In *International Symposium on Physical Design*, pages 42–47, ACM,April 2002.

[6] J. Cong, M. Romesis, and M. Xie. Optimality, scalability and stability study of partitioning and placement algorithms. In *International Symposium on Physical Design*, pages 88–94, April 2003.

[7] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. In *Design Automation Conference*, pages 526–529. IEEE/ACM, 1997.

[8] P. H. Madden. Reporting of standard cell placement results. In *International Symposium on Physical Design*, pages 30–35. ACM, April 2001.

[9] M. Sarrafzadeh, M. Wang, and X. Yang. *Modern Placement Techniques*. Kluwer Academic Publishers, 2002.