Placement Using a Localization Probability Model (LPM)

Markus Olbrich, Erich Barke
Institute of Microelectronic Systems, EDA Group
Hannover, Germany
{olbrich | barke}@ims.uni-hannover.de

Abstract

We propose a new placement model for global placement. This model uses probabilities to localize the cells. It enables arbitrary levels of placement abstraction. Wirelength estimations at any level can be derived from the model. We present a new placer, that uses a special variant of the proposed model. Examples show that the model properties improve placement quality.

1 Introduction

In the area of VLSI design, placement plays an important role due to the fact that it influences performance properties substantially. Three main approaches are used in today's placers: Simulated Annealing [7, 8, 9], analytical methods as force directed placement [5] and partitioning based methods [3, 4] mostly using min-cut heuristics [2, 6].

Algorithms for large circuits benefit from two concepts for shifting from rough to detailed placement. The first one is topological coarseness which is achieved by introducing hierarchy into the circuit. Hierarchy was generated automatically by clustering already in [8]. Today, this approach is known as "multilevel placement" [10].

The second concept is topographical coarseness. State-of-the-art placement algorithms benefit from separating global and detailed placement phases. Global placement means that cells do not obtain exact positions or their positions are only "hints", respectively. Final non-overlapping cell positions are assigned during a following detailed placement phase. We extend this idea in a more consistent way. Our goal is to represent cell positions of continuous and assignable strictness enhancement. This allows an arbitrary number of strictness steps during placement. As a result, we propose our localization probability model (LPM). The concept is to enable the definition of blurred positions. Instead of determining the cell positions precisely during placement we specify localization probabilities for the cells.

2 Localization Probability Model

The position of each cell i is determined by the function $p_i(x,y)$ which defines the probability, that position (x,y) is inside cell i. If the area of a cell i is w_i, the probability function has the following property, where A is the infinite area:

$$\int_{\mathrm{A}} p_i(x,y)\,dA = w_i \tag{1}$$

We use the wirelength expectancy values as wirelength estimations by applying the localization probabilities.

3 Special LPM

Our standard-cell placement algorithm uses a special version of the LPM called SLPM. For runtime and memory efficiency reasons, we limit the types of probability functions. We use rectangular regions with a constant probability inside the rectangle and zero outside. Additionally, the rectangle positions are restricted in the following way: If the layout area lies inside a square of area A_s, then the probability functions are of the following form:

$$p_i(x,y) = \tilde{p}_i R\left(\frac{2^{s_i}x}{\sqrt{A_s}} - \frac{x_i}{2}\right) R\left(\frac{2^{s_i}y}{\sqrt{A_s}} - \frac{y_i}{2}\right) \tag{2}$$

$$\tilde{p}_i = 4^{s_i}\frac{w_i}{A_s} \tag{3}$$

$$0 \le x_i < 2^{s_i+1} - 1 \wedge 0 \le y_i < 2^{s_i+1} - 1 \tag{4}$$

$$x_i, y_i, s_i \in \mathbf{N} \tag{5}$$

with

$$R(t) = \begin{cases} 1 & : \quad 0 < t < 1 \\ 0 & : \quad \text{otherwise} \end{cases} \tag{6}$$

The probability function of each cell i is fully determined by the parameters w_i, x_i, y_i and s_i. Since w_i is constant, a placement algorithm has to assign only the latter three. They determine the position and the size of a rectangular

1530-1591/04 $20.00 (c) 2004 IEEE

region. The s_i of the cells can differ during the placement progression. That means, the regions of different cells can have different sizes at the same time. Additionally, the regions can overlap.

4 Placement Algorithm

We implemented a global placer based on the SLPM called SLPMP. Global placement is a combinatorial optimization problem. In this first approach, we use the total wirelength as a cost function.

Our global placer controls the following parameters: the region position (x_i, y_i) of the cells and the region size s_i that changes from a low to a high value.

In the R-phase, the following is executed for each cell i: First s_i is assigned $s_i + 1$. Then, the position (x_i, y_i) is chosen to be the one that achieves the lowest f_c.

The V-phase varies the positions (x_i, y_i) of randomly selected cells while keeping s_i constant. In our current implementation the V-phase works greedily. A variation is not accepted if f_c increases.

For detailed placement, we use the following simple heuristic: We put the cells to a randomly chosen position according to its p_i. Then, we choose the next standard-cell row. In each row we sort the cells and distribute them equidistantly. Figure 1 shows a resulting legalized placement.

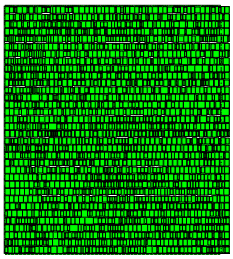

Figure 1. Legalized placement

5 Results

The efficiency of our placer was compared to results received from the state-of-the-art placer Dragon2000 [10]. Dragon2000 outperforms many other academic and commercial standard-cell placers. Especially the runtimes of Dragon2000 are short compared to other standard-cell placers. Therefore, a comparison with Dragon2000 is very meaningful. We took examples from the MCNC standard-cell placement benchmark suite and applied them to both Dragon2000 and SLPMP. Since we used the binary of Dragon [1], we were able to run both placers on the same machine. Additionally, we used the same input files for both placers. A utilization of 95% and no row space was assumed.

Table 1. Wirelength and runtime changes of SLPMP compared to Dragon2000

Circuit	Wirelength	Time
fract	-18.3%	0.0%
primary1	-19.7%	-15.8%
primary2	12.6%	-23.4%
industry2	-8.6%	118.4%
industry3	23.9%	-24.2%

We calculated the average of ten runs with default parameter settings. Table 1 shows the relative results. The results of SLPMP are competitive to Dragon2000 in both wirelength and runtime.

6 Conclusion

We presented a general placement model for representing global placements with arbitrary precision. Although we used very simple strategies for varying cell positions, we obtained placement results of similar quality and runtime as state-of-the-art placers. We expect future improvements by enhancing the cost function and the detailed placement procedure.

References

[1] http://www.cs.ucla.edu/∼xjyang/dragon/download.html.

[2] C. J. Alpert, D. J.-H. Huang, and A. B. Kahng. Multilevel circuit partitioning. In *DAC 97*, pages 530–533, 1997.

[3] M. Breuer. A class of min-cut placement algorithms. In *DAC 77*, pages 284–289, 1977.

[4] A. E. Caldwell, A. B. Kahng, A. A. Kennings, and I. L. Markov. Hypergraph partitioning for VLSI CAD. In *DAC 99*, pages 349–354, 1999.

[5] H. Eisenmann and F. M. Johannes. Generic global placement and floorplanning. In *DAC 98*, pages 269–274, 1998.

[6] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *DAC 82*, pages 175–181, 1982.

[7] C. Sechen. *VLSI Placement and Global Routing Using Simulated Annealing*. Kluwer Academic Publishers, Boston, 1988.

[8] W.-J. Sun and C. Sechen. Efficient and effective placement for very large circuits. In *ICCAD 93*, pages 170–177, 1993.

[9] J. D. Vicente, J. Lanchares, and R. Hermida. FPGA placement by thermodynamic combinatorial optimization. In *DATE 02*, pages 54–60, 2002.

[10] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. In *ICCAD 00*, pages 260–263, 2000.