

Optimization of Placement for Modern VLSI Design

R. Prabhakar
Assoc. Professor
HITS, Bogaram
R.R. District, A.P, India

Dr. K.E. Srinivasa
Murthy
Principal, SVIT
Anantapur, A.P, India

Dr. K. Soundara Rajan
Professor, ECE Dept.
JNTU, Anantapur
A.P, India

ABSTRACT

The VLSI placement problem is to place objects into a fixed die such that there are no overlaps among objects and some cost metrics (wire length, routability) are optimized. The nature of multiple objects and incremental design process for modern VLSI design demands Advanced Incremental Placement Techniques. Incremental placement changes either the wire length or the Placement Density of an existing placement to optimize one or more design objectives. The difficulty of Incremental Placement lies in that it is hard to optimize one design objective while maintaining the existing optimized design objectives. In this dissertation, three main approaches are used to overcome this problem. The first approach is to perform sensitivity analysis and make smaller changes one step at a time on the most sensitive direction. This approach always changes placement in the direction where a change can result in the highest improvement in design objective. The second approach is to maintain the relative order during Incremental Placement. This is called a 'Correct-by-construction' approach. When we move cells while maintaining their relative order, it is implicitly preserve the existing design characteristics. The third approach is to specify maintain other design constraints while optimizing one design objective. This is more direct approach. It needs to formulate design constraints that can be honored by incremental placer. For the first approach, two techniques are available. First technique is 'Sensitivity based Netweighting'. The objective is to maintain both Worst Negative Slack (WNS) and Figure of Merit (FOM), defined as the Total Slack Difference, compared to a certain Threshold for all timing end points. It performs Incremental global placements with the netweights based on comprehensive analysis of the wirelength, Slack on FOM sensitivities to the netweight. The second technique is noise map driven two step incremental placements. The novel noise map is used to estimate the placement impact on coupling noise, which takes into account of Coupling Capacitance, Driver Resistance and Wire resistance. It performs a two step incremental placement i.e., cell inflation and Local refinement, to expand regions with high noise impact in order to reduce Total Noise.

The technique for second approach is Diffusion based Placement Migration, which is the smooth movement of the cells in an existing placement to address a variety of post placement design issues. This method simulates a diffusion process where cells move from high concentration area to low concentration area. The application on Placement Legalization shows significant improvements in wirelength and timing as compared to the other commonly used legalization techniques. For the third approach, a technique called First-do-no-harm detailed placement is used. It uses set of pin-based timing and electrical constraints to prevent detailed placement techniques from degrading timing or violating electrical constraints while

reducing wirelength. This technique will provide better result for detailed placement not only reduces Total Wirelength (TWL), but also significantly improves timing.

General Terms

Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords

Keywords are your own designated keywords which can be used for easy location of the manuscript using any search engines.

1. INTRODUCTION

The VLSI placement problem is to place objects into a fixed die such that there are no overlaps among objects and some cost metric (e.g., wirelength, routability) is optimized. It is a major step in physical design that has been studied for decades. Yet, it has attracted much attention recently because recent studies show that existing placers still cannot produce near optimal solutions. As a result, many new academic placers were invented in the recent years. Further, modern VLSI design challenges have reshaped the placement problem. As the feature size keeps shrinking, billions of transistors (or millions of standard cells) can be integrated in a single chip. Meanwhile, the intellectual property (IP) modules and pre-designed macro blocks (such as embedded memories, analog blocks, pre-designed datapaths, etc.) are often reused. As a result, advanced VLSI designs often contain a large number (hundreds) of macros of very different sizes from each other and the standard cells, and some of the macros may be preplaced in the chip. The dramatically increasing interconnect complexity further imposes routing difficulty. In addition to wirelength, therefore, modern placement shall also consider the density constraint. To solve such the modern large-scale mixed-size placement problem, many academic placers were invented in recent years. Those placers can be classified into three major categories: (1) the analytical approach, (2) the min-cut partitioning based approach, and (3) the hybrid approach. Among those approaches, the analytical placers have shown their superior efficiency and quality. To handle preplaced blocks, NTUplace3 applies a two-stage smoothing technique, Gaussian smoothing followed by level smoothing, to facilitate block spreading during global placement. The density is controlled mainly by cell spreading during global placement and cell sliding during detailed placement. We further use the conjugate gradient¹ The log-sum-exp wirelength model is a patented technology and use requires a license from Synopsys. method with dynamic step-size control to speed up the global placement and apply macro shifting to find better macro positions. During legalization, NTUplace3

removes the overlaps and places all standard cells into rows using a priority-based scheme based on block sizes and locations. A look-ahead legalization scheme is also incorporated into global placement to facilitate the legalization process. During detailed placement, NTUplace3 adopts cell matching and cell swapping to minimize the wirelength, and cell sliding to optimize the density. Although the recent academic placers have made significant progress in the large-scale mixed-size placement problem, there are still many emerging challenges for this problem. As the number of macros increases dramatically, the single-stage methodology of integrated macro and standard-cell designs incurs significant difficulties in legality and complexity. Other design methodologies would be needed to tackle the increasing design complexity. In addition to wirelength, other cost metrics such as routability, timing, power, and thermal should also be addressed to handle the increasing integration complexity and operation frequency. We shall discuss these placement challenges and related research directions. The remainder of this paper is organized as follows. Section II gives the analytical model used in NTUplace3. Section III explains the placement techniques employed in NTUplace3. Finally, conclusions are given in Section VI.

2. ANALYTICAL PLACEMENT MODEL

Circuit placement can be formulated as a hypergraph $H = (V, E)$ placement problem. Let vertices $V = \{v_1, v_2, \dots, v_n\}$ represent blocks and hyperedges $E = \{e_1, e_2, \dots, e_m\}$ represent nets. Let x_i and y_i be the respective x and y coordinates of the center of the block v_i , and a_i be the area of the block v_i . The circuit may contain some preplaced blocks which have fixed x and

y coordinates and are not movable. We intend to determine the optimal positions of movable blocks so that the total wirelength is minimized and there is no overlap among blocks. x_i, y_i center coordinate of block v_i , w_i width and height of block v_i w_b, h_b width and height of bin b M_b maximum area of movable blocks in bin b D_b potential (area of movable blocks) in bin b P_b base potential (preplaced block area) in bin b $tdensity$ target placement density Fig. 1. Notation used in this paper. To evenly distribute the blocks, we divide the placement region into uniform non-overlapping bin grids. Then, the global placement problem can be formulated as a constrained minimization problem as follows: $\min W(\mathbf{x}, \mathbf{y})$ s.t. $D_b(\mathbf{x}, \mathbf{y}) \leq M_b$, for each bin b ,

where $W(\mathbf{x}, \mathbf{y})$ is the wirelength function, $D_b(\mathbf{x}, \mathbf{y})$ is the potential function that is the total area of movable blocks in bin b , and M_b is the maximum area of movable blocks in bin b . M_b can be computed by

$M_b = tdensity(w_b h_b - P_b)$, where $tdensity$ is

a user-specified target density value for each bin, w_b (h_b) is the width (height) of bin b , and P_b is the base potential that equals the preplaced block area in bin b . Note that M_b is a fixed value as long as all preplaced block positions are given and the bin size is determined. The wirelength $W(\mathbf{x}, \mathbf{y})$ is defined as the total half-perimeter

wirelength (HPWL) given by

$$W(\mathbf{x}, \mathbf{y}) = \text{net } e(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j|). \quad (2)$$

Since $W(\mathbf{x}, \mathbf{y})$ is non-convex, it is hard to minimize it directly. Thus, several smooth wirelength approximation functions are proposed in the literature. In NTUplace3, we apply the log-sum-exp wirelength model,

$$\gamma_e \in E(\log_{\sum_{v_i, v_j \in e} \exp(x_i/k\gamma)} + \log_{\sum_{v_i, v_j \in e} \exp(-x_i/k\gamma)} + \log_{\sum_{v_i, v_j \in e} \exp(y_j/k\gamma)} + \log_{\sum_{v_i, v_j \in e} \exp(-y_j/k\gamma)}). \quad (3)$$

As $\gamma \approx 0$, log-sum-exp wire length gives a good approximation to the HPWL.

3. CORE TECHNIQUES OF NTUPLACE3

Like many modern placers, NTUplace3 consists of three major steps: global placement, legalization, and detailed placement. Global placement evenly distributes the blocks and finds the better position for each block to minimize the target cost (e.g., wirelength). Then, legalization removes all overlaps among blocks and places standard cells into rows. Finally, detailed placement further refines the solution quality. In the following sections, we describe the underlying techniques used in the global placement, legalization, and detailed placement of NTUplace3.

Algorithm: Multilevel Global Placement

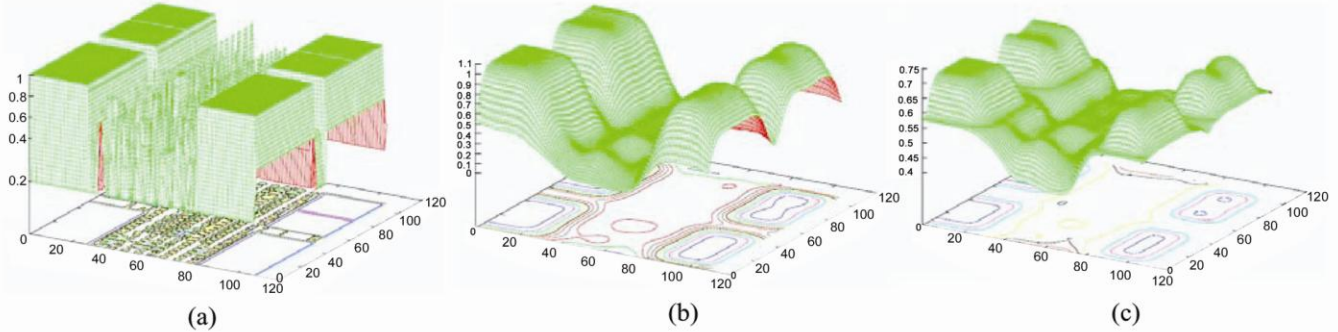
Input:

hypergraph H_0 : mixed-size circuit n_{max} : the maximum block number in the coarsest level

Output:

(x^*, y^*) : optimal block positions

01. $level = 0$;
02. **while** ($BlockNumber(H_{level}) > n_{max}$)
03. $level++$;
04. $H_{level} = FirstChoiceClustering(H_{level-1})$;
05. initialize block positions by $SolveQP(H_{level})$;
06. **for** $currentLevel = level$ **to** 0
07. initialize bin grid size $n_{bin} \propto \sqrt{n_x}$;
08. initialize base potential for each bin;
09. initialize $\lambda_0 = -\partial W(\mathbf{x}, \mathbf{y}) / \partial D_b(\mathbf{x}, \mathbf{y})$; $m = 0$;
10. **do**
11. solve $\min W(\mathbf{x}, \mathbf{y}) + \lambda_m (D_b(\mathbf{x}, \mathbf{y}) - M_b) / 2$;
12. $m++$;
13. $\lambda_m = 2\lambda_{m-1}$;
14. **if** ($currentLevel == 0$ & $overflow\ ratio < 10\%$)
15. call $LookAheadLegalization()$ and save the best result;
16. compute $overflow\ ratio$;
17. **until** (spreading enough or no further reduction in $overflow\ ratio$)
18. **if** ($currentLevel == 0$)
19. restore the best look-ahead result;
20. **else**
21. call $MacroShifting()$;
22. decluster and update block positions.



a) The density Profile of newblue2. b)The base potential after Gaussian Smoothing result in a better smoothing potential.c) The base potential after level Smoothing.

A. Global Placement

As mentioned earlier, the global placement is based on the multilevel framework and the log-sum-exp wirelength model. A two-stage smoothing technique is used to handle preplaced blocks. We further use the conjugate gradient method with dynamic stepsize control to speed up the global placement and apply macro shifting to find better macro positions. Now we detail those techniques.

1) Multilevel Framework:

We use the multilevel framework for global placement to improve the scalability. Our algorithm is summarized in Figure 2. The coarsening stage (lines 1–4) iteratively clusters the blocks based on connectivity/size to reduce the problem size until a given threshold is reached. Then, we find an initial placement (line 5). In the uncoarsening stage (lines 6–22), it iteratively declusters the blocks and refines the block positions to reduce the objective function. The declustering process continues until all blocks are evenly distributed. In NTUplace3, the evenness of block distribution is measured by the *overflow ratio*, which is defined as follows:

$$\text{overflow ratio} = \frac{\sum_b \max(Db(\mathbf{x}, \mathbf{y}) - Mb, 0)}{\text{total movable area}} \quad (7)$$
 The global placement stage stops when the overflow ratio is less than a user-specified target value, which is 0 by default.

2) Base Potential Smoothing:

Preplaced blocks pre-define the *base potential*, which significantly affects block spreading. Since the base potential P_b

is not smooth, it incurs mountains that prevent movable blocks from passing through these regions.

Therefore, we shall smooth the base potential to facilitate block spreading. We first use the Gaussian function to smooth the base potential change, remove the rugged regions in the base potential, and then smooth the base potential level so that blocks can spread to the whole placement region. The base potential of each block can be calculated by the bellshaped function. However, we observe that the potential generated by the bell-shaped function has “valleys” among the adjacent regions of blocks, and these regions do not have any free space but their potentials are so low that a large number of blocks may spread to these regions. To avoid this problem, we use the Gaussian function to smooth the base potential. The twodimensional Gaussian is given by

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (8)$$

where σ is the standard deviation of the distribution. Applying convolution to the Gaussian function G with the base potential P , $P_{-}(x, y) = G(x, y) * P(x, y)$, we can obtain a smoother base potential P_{-} . Gaussian smoothing works as a low-pass filter, which can smooth the local density change. After the Gaussian smoothing, we apply another landscape smoothing function to reduce the potential levels. The smoothing function $P_{--}(x, y)$ is defined as follows:

$$P_{--}(x, y) = \begin{cases} P_{-} + (P_{-}(x, y) - P_{-})\delta & \text{if } P_{-}(x, y) \geq P_{-} \\ P_{-} - (P_{-}(x, y) - P_{-})\delta & \text{if } P_{-}(x, y) \leq P_{-} \end{cases} \quad (9)$$

where $\delta \geq 1$. Level smoothing reduces “mountain” (high potential regions) heights so that blocks can spread to the whole placement area smoothly. Figure 3 shows the smoothing process of the circuit newblue2.

3) Conjugate Gradient Search with Dynamic Step Size:

We use the conjugate gradient (CG) method with dynamic step size instead of line search to minimize Equation (6). After computing the conjugate gradient direction dk , the step size ak is computed by $ak = s/||dk||_2$, where s is a user-specified scaling factor. By doing so, we can limit the step size of block spreading since the total quadratic Euclidean movement is fixed, v

$$i \in V(\Delta x 2i + \Delta y 2i) = ||akdk||_2 = s^2, (10)$$

where Δxi and Δyi are the respective amounts of the movement along the x and y directions for the block vi in each iteration. To show the effectiveness of the dynamic step-size control, we performed experiments on adaptec1 with different step sizes. In, the CPU times and HPWLs are plotted as functions of the step sizes. As shown in , the CPU time decreases as the step size s becomes larger. In contrast, the HPWL decreases as the step size s gets smaller. The results show that the step size significantly affects the running time and the solution quality. Fig. 4. The CPU times and HPWLs resulting from different step sizes based on the circuit adaptec1.

4) Macro Shifting:

In the global placement stage, it is important to preserve legal macro positions since macros are much bigger than standard cells and illegal macro positions typically make legalization much more difficult. To avoid this, we apply macro shifting at each declustering level of the global placement stage. Macro shifting moves macros to the closest legal positions. Integrated within the multilevel framework, only macros with sizes larger than the average cluster size of the current level are processed. Then, the legal macro positions provide a better initial solution for the next declustering level, and those macros are still allowed to spread at subsequent declustering levels.

B. Legalization

After global placement, legalization removes all overlaps and places standard cells into rows. We extend the standard-cell legalization method in to solve the mixed-size legalization problem. The legalization order of macros and cells are determined by their x coordinates and sizes (widths and heights). Larger blocks get the priority for legalization. Therefore, we legalize macros earlier than standard cells. After the legalization order is determined, macros are placed to their nearest available positions and cells are packed into rows with the smallest wirelength. Despite its simplicity, this macro/cell legalization strategy works well on all benchmarks. Recall that we performed block spreading during global placement. It is important to determine when to terminate the block spreading. If blocks do not spread enough, the wirelength may significantly be increased after legalization since blocks are over congested. If blocks spread too much, the wirelength before legalization may not be good even the legalization step only increases wirelength a little. This situation becomes even worse when the density is also considered, since the placement objective is more complicated. To improve the legalization quality, we use a look-

ahead legalization technique during global placement to make the subsequent legalization process easier. At the finest level of the multilevel placement, we apply legalization right after placement objective optimization in each iteration and record the best result with the minimum cost (wirelength and density penalty). Although the look-ahead legalization may take longer running time due to more iterations of legalization, we can ensure that blocks do not over spread and thus obtain a better legal placement. As a result, the look-ahead legalization significantly alleviates the difficulty in removing the macro and standard-cell overlaps during the later legalization stage, and eventually leads to a more robust placement result.

C. Detailed Placement

The detailed placement stage consists of two stages: the wirelength minimization stage and the density optimization stage. In the wirelength minimization stage, we apply *cell matching* and *cell swapping* to reduce the total wirelength. In the density optimization stage, we apply the *cell sliding* technique to reduce the density overflow in congested regions. In the following, we explain the cell-matching, cell-swapping, and cell-sliding algorithms.

1) Cell Matching:

We extend the window-based detailed placement (WDP) algorithm and name our approach *cell matching* here. The WDP algorithm finds a group of exchangeable cells inside a given window, and formulates a bipartite matching problem by matching the cells to the empty slots in the window. The cost is given by the HPWL difference of a cell in each empty slot. The bipartite matching problem can be solved optimally in polynomial time, but the optimal assignment cannot guarantee the optimal HPWL result because the HPWL cost of a cell to each empty slot depends on the positions of the other connected cells. Our cell matching algorithm remedies this drawback by selecting *independent cells* at one time to perform bipartite matching. Here by independent cells, we mean that there is no common net between any pairs of the selected cells.

2) Cell Swapping:

The cell swapping technique selects k adjacent cells each time to find the best ordering by enumerating all possible orderings using the branch-and-bound method. Here, k is a user-specified parameter. In our implementation, we set $k = 3$ for a good tradeoff between the running time and solution quality. This process repeats until all standard cells are processed.

3) Cell Sliding:

The objective of cell sliding is to reduce the density overflow in the congested area. We divide the placement region into uniform non-overlapping bins, and then iteratively reduce the densities of overflowed bins by sliding the cells horizontally from denser bins to sparser bins, with the cell order being preserved. Each iteration consists of two stages: left sliding and right sliding. In each stage, we calculate the density of each bin and then

compute the area flow $fb_{b_}$ between bin b and its left or right neighboring bin $b_$. Here, $fb_{b_}$ denotes the desired amount of cell area to move from bin b to $b_$. Recall that we define Db as the total movable cell area in bin b and Mb as the maximum allowable block area in bin b . If bin b has no area overflow or the area overflow ratio of b is smaller than $b_$, that is $Db \leq Mb$ or $Db/Mb \leq Db_/Mb_$, we set $fb_{b_} = 0$. Otherwise we calculate $fb_{b_}$ according to the capacity of $b_$. If bin $b_$ has enough free space, we move the overflow area of bin b to $b_$. Otherwise, we evenly distribute the overflow area between b and $b_$. Therefore, $fb_{b_}$ is defined by

$$fb_{b_} = Db - Mb, \text{ if } (Mb_ - Db_) \geq (Db - Mb)DbMb_-Db_MbMb+Mb_, \text{ otherwise, (11)}$$

where the second condition of Equation (11) is derived from $Db - Mb + (Db - Mb + Db_ - Mb_)MbMb + Mb_ = DbMb - Db_MbMb + Mb_ \setminus$, (12)

After the area flow $fb_{b_}$ is computed, we sequentially slide the cells across the boundary between b and $b_$ until the amount of sliding area reaches $fb_{b_}$ or there is no more area for cell sliding. Then we update Db and $Db_$. In the right sliding stage, we start from the left-most bin of the placement region, and $b_$ is right to b . In the left sliding stage, we start from the right-most bin, and $b_$ is left to b , accordingly. We iterative slide the cells from the area overflow regions to sparser regions until no significant improvement can be obtained.

4. CONCLUSIONS

Modern VLSI design challenges have reshaped the placement problem. In this paper, we have presented example techniques to tackle the challenges arising from large-scale mixed-size circuit designs with the wirelength optimization. Although significant progress has been made in placement research, modern circuit

designs have induced many more challenges and opportunities for future research on macro placement and routability-, timing-, power-, and/or thermal-driven optimization of the placement problem.

5. REFERENCES

- [1]. Tung_chieh, Zhe-Wei Jiang, Tien-chang Hsu, Hsin-Chen Chen, Ntu Place 3: An Analytical Placer for Large-Scale Mixed-Size Designs With Preplaced Blocks and Density constraints, IEEE transactions on Computer Aided Design of Integrated Circuits and Systems, Vol 27, No.7, July-2008, Page No:1228-1240..
- [2]. Tung-Chieh Chen, Ping-Hung Yuh, Yao-Wen Chang, Fwu-Juh Huang and Tien-Yueh Liu, MP-Trees: A Packing Based Macro placement Algorithm for Modern Mixed Size Designs, IEEE transactions on Computer Aided Design of Integrated Circuits and Systems, Vol 27, No.9, September-2008, Page No:1621-1634.
- [3]. Jason Cong, and Mix xie: A Robust Mixed Size Legalization and Detailed Placement Algorithm, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol 27, No.8, August-2008, Page No:1349-1362..
- [4]. Zhe-Wei Jiang, Hsin-Chen Chen, Tung Chieh Chen and Yao-Wen Chang: Challenges and Solutions in Modern VLSI Placement.
- [5]. Saurabh N. Adya, Igor L. Markov and Paul G. Villarrubia: Improving Min-Cut Placement for VLSI using Analytical Techniques.
- [6]. Jurgen M. Kleinhan, Georg Sigl, Frank M. Johannes, and Kurt J. Antreich, Gordian: VLSI Placement by Quadratic Programming and Slicing Optimization, IEEE transactions on Computer Aided Design, Vol 10, No.3, March-1991, Page No:356-365..