

# Routability-driven Placement for Mixed-size Designs using Design-hierarchy and Pin Information

Prasun Datta

*Dept. Computer Science and Engineering*  
*National Institute of Technology Silchar*  
Assam, India  
datta.prasun@gmail.com

Shyamapada Mukherjee

*Dept. Computer Science and Engineering*  
*National Institute of Technology Silchar*  
Assam, India  
shyamamukherji@gmail.com

**Abstract**—Routability-driven placement is a better idea than wirelength-driven placement because the first one considers global routing constraints along with placement constraints which help in satisfying and finding detailed routing in integrated circuit design. In this paper, we have proposed a routability-driven placement for large mixed-size standard cell designs. A new circuit block clustering technique has been introduced with the help of design hierarchy, pin offset and pin direction information. A modified quadratic programming based global placement method has been devised. A new routability-aware cell spreading scheme has been proposed to implement legalization considering pin information. Finally, a routing congestion dependent detailed placement has been formulated using cell swapping and movement. The implemented algorithms outperforms the other recent routability-driven placers on ICCAD 2012 benchmarks.

**Index Terms**—VLSI, Placement, Mixed-size designs, Routability, Analytical methods.

## I. INTRODUCTION

The unprecedented growth area of the semiconductor industry for the last two decades or so by far, is the Very Large Scale Integrated (VLSI) technology that provides a new and more complex range of ‘of the self’ circuits containing more than billions of transistors in a single chip. Due to the enrichment in process technology and thus increased architectural complexity of circuits, automated cell placement for VLSI circuits becomes an integral part in Very Large Scale Integrated (VLSI) physical design cycle for achieving a design with optimized routability and minimized total estimated wirelength along with reduced timing usage. In the past few years, several routability-driven placement contests ISPD 2011 [1], DAC 2012 [2], ICCAD 2012 [3], ISPD 2014 [4] were organized to encourage research in this field. Congestion is one of the important optimization objectives in Very Large Scale Integrated (VLSI) placement stage. Numerous probabilistic congestion estimation approaches [5], [6], [7], [8], [9] have been proposed in the past years for capturing and examining methodically the routing congestion. A number of previous probabilistic global routing algorithms [10], [11], [5], [12] can estimate congestion of a region during placement. Some quality global routers FastRoute [13], BFG-R [14], and NCTUgr [15] in run time may be combined with other placers like (IPR [13], SimPLR [16], Ripple [17], and POLAR 2.0 [18], Ripple 2.0 [19]) to obtain quality results during estimating congestion.

After estimating congestion, the placement steps for routability optimization are performed through global placement, detailed placement and post-placement process. There have been several studies on incorporating the congestion metric during the global-placement stage of physical synthesis flow. The works [20], [21] have devised congestion estimation technique within partition-driven quadratic placement strategy.

In the work [22] develops a white-space allocation (WSA) technique based on [23] for congestion estimation during min-cut global placement. In the last couple of years several global placement approaches [24], [25], [26], [27], [9] attempted to modify the objective function under analytic placement framework using additional optimization forces or costs to move cells away from the highly congested areas. Most of these analytic placers use nonlinear function [25], [26], [27] for modeling routing congestion. These placement techniques based on multilevel framework to achieve high-quality placement results but sometimes it is quite difficult to estimate congestion correctly of those intermediate level designs. Some existing placement algorithms, such as [28], [19], [17], [29], [23], [30], [31], [11] employ white space allocation or cell bloating techniques to lessen the cell densities in the congested areas and try to improve routability and quality of layout in view of routing demand reduction strategy. To cope with the routing overflow occurring along the boundaries of the preplaced macros, NTUplacer4 [32] dynamically changes the base potential around narrow channels between macros. In detailed placement, the most familiar technique is to change the objective of cell swapping to alleviate congestion [33], [16], [34], [35], [36]. In the post-placement process, there are several previous approaches addressing routability, such as Crop [36], CRISP [31], and Ropt [37]. After achieving the congestion map, Crop [36] adjusts the boundary of each G-Cell and distributes cells within the G-Cell. CRISP [31] adopts cell inflation and spreading to allocate cells more sparsely in the congested regions. [37] refines routability by using the routing information produced through a global router. However, it is difficult for congestion estimators to adequately predict the routing congestion because routing solution quality varies with routers comprise dissimilar routing congestion estimation. So, in order to achieve correct interconnection information, it is needed to incorporate routing into placement process. In

recent years, modern circuit designs have frequently employed hierarchical design methodologies for faster turnaround time. Consequently, how to achieve expected routability and circuit performance during placement preserving the design hierarchy of a circuit incurs greater challenges to modern circuit designs. Therefore, for better scalability and solution quality it is desirable to devise an effective routability-driven placer for modern hierarchical mixed-size designs.

## II. PRELIMINARIES

Since circuit placement has been an important and known problem, it has been defined very clearly by various researchers. In this section, we discuss the routability-driven placement problem and its mathematical form.

### A. Problem Formulation

The placement problem is represented as a given hypergraph  $H = (V, E)$  where  $V = \{v_1, v_2, \dots, v_n\}$  is a set of nodes or circuit blocks and  $E = \{e_1, e_2, \dots, e_m\}$  is the set of hyperedges connecting different blocks in  $V$ . Among the blocks in  $V$ , some blocks ( $F$ ) have pre-placed (fixed) locations in the chip area whereas rest of the blocks are movable ( $M$ ) and can be placed anywhere in the placement area. Now, the placement problem can be defined as finding the suitable location of the blocks in  $M$  such the total wirelength is minimized. This is called wirelength-driven placement. Hence, in the placement problem when along with the wirelength, the global routing constraints or routing resource constraints are also considered to be satisfied, the placement problem is called routability-driven placement. Therefore, in routability-driven placement, there is always a trade off between wirelength and routing congestion. Half perimeter wirelength (HPWL) is the main objective in any traditional wirelength-driven placers. The objective function is expressed as follows:

$$\min : W(X, Y) \quad (1)$$

where  $W(X, Y)$  is the sum of all the HPWL for the all the blocks in various nets and  $X$  and  $Y$  are the coordinate vectors of the blocks contain lower left value of  $x$  and  $y$  coordinate of each block.

$$W(X, Y) = \sum_{e \in E} \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \quad (2)$$

Since,  $W(X, Y)$  is not differentiable, hence minimizing the value is not very easy. Various methods have been applied for smoothening this function. Quadratic form is very popular in this case such as:

$$\hat{W}(X, Y) = \sum_{v_i, v_j} w_{i,j} ((x_i - x_j)^2 + (y_i - y_j)^2) \quad (3)$$

where  $w_{i,j}$  is connection weight between blocks  $v_i$  and  $v_j$ .

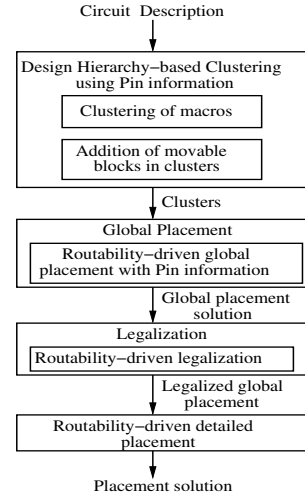


Fig. 1: Flowchart

### B. Design Hierarchy of Circuit Netlist

For modern circuits, a hierarchical design methodology is pervasive mainly due to its faster turnaround time. In this methodology, a design is partitioned according to logical functions, and each individual block is synthesized separately. The designers who understand the logical functions of a chip naturally come up with a hierarchical partition to form a set of hierarchy groups, and intuitively it makes sense to place circuit elements with the same logical function in the same vicinity for better routability. According to the nature of hierarchy groups, we can use a design hierarchy tree structure to represent the relations among hierarchy groups, as shown in Fig. 2b. Each internal node denotes the hierarchy classification, and each leaf node is a single circuit block in the netlist. In a design-hierarchy tree, since each hierarchy has different tasks or features, not all branches have the same depth. Besides, each node has its parent, which implies the hierarchy group to which it belongs.

## III. PROPOSED PLACEMENT ALGORITHM

Our objective is to determine block (macro/cell) locations in the global placement stage so that potential routing overflows are minimized. We perform the following tasks to achieve this objective: (1) balanced hierarchy grouping, (2) hierarchy aware clustering, (3) hierarchy aware analytical placement, and (4) net-topology-based block spreading. The flowchart in Fig. 1 shows the workflow of the proposed technique. We discuss these issues in the following subsections.

### A. Design Hierarchy-based Clustering using Pin Information

Placement for different circuits blocks (macros/cells) directed by their design hierarchy gives a better placement result in terms of wirelength and congestion [3], [27]. Though, sometimes inappropriate hierarchy consideration may lead to bad results also. Few papers [27], [32] have already considered hierarchy of the blocks to achieve better placement results but they have not considered *pin offset* and *direction* which

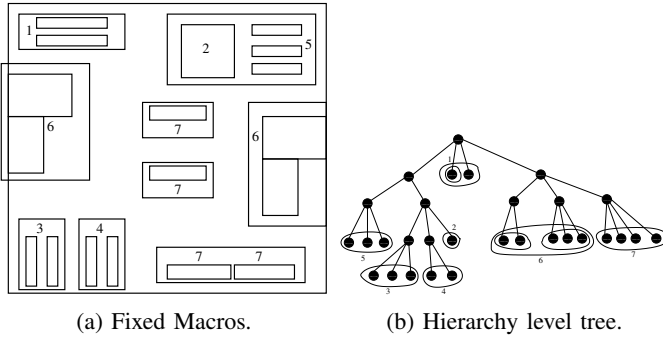


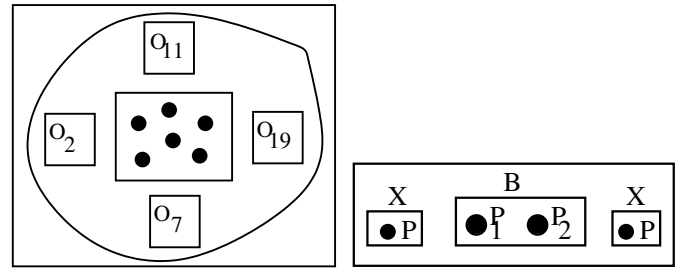
Fig. 2: Clustering of macros and the hierarchical level tree of modules.

we have considered in our proposed technique for making clusters of circuit blocks and placement along with hierarchy information. Pin offset and direction can help in minimizing routing wirelength in a better and complete way.

In this section, we present a design-hierarchy based clustering of circuit blocks with the help of pin information. Clustering is implemented in two steps: (1) clustering of macros and (2) addition of movable blocks in the clusters created in the step 1. In both the steps, hierarchy of the blocks and pin information are considered for clustering.

1) *Clustering of Macros*: In hierarchical designs, circuit blocks which perform the same logical function are classified and kept in the same hierarchy level or group. Therefore, keeping the macros of the same logical function is a good idea. But, according to the designers perspective, the macros which are far apart from each other should not be included in the same group even if they do the same logical function. Otherwise, it will increase the wirelength. Similarly, position of the pins (pin offset) w.r.t the center of a circuit block has also a great impact on the total wirelength. Hence, the pin offset and direction during clustering extend the placement in the right direction to reduce the wirelength and congestion. After placement based on the relative position of the blocks, the required wirelength may increase or decrease. Therefore, in this paper, we have considered all the important point during clustering from design-hierarchy to pin information to reduce the wirelength and congestion as much as possible. In Fig. 2, clustering technique has been elaborately depicted.

2) *Addition of Movable Blocks in Clusters*: In large mixed-size designs, macros have different shape and size. They also create blockages and occupy a reasonable amount of routing channel area due to less porosity issue. Hence, placing movable blocks very close to macros create routing congestion problem very prominent. Hence, including movable standard cells in a macro cluster is a very important concern to be handled very carefully. In this phase, pin information and blockages created by the macros are taken into consideration to include the standard cells in the clustered created in the previous phase. The effect of consideration of pin information (pin offset and direction) has been shown in Fig. 3a. Based on the pin offset, we tentatively decide the relative position



(a) Pin offset for relative position. (b) Pin direction for placement.

Fig. 3: A block diagram for the relative position of blocks according to pin offset and pin direction.

of movable blocks with respect to the macros with which the movable blocks are making a cluster. Left, right, above, and below are the relative positions of the blocks w.r.t. the macros based on the pin offsets of a block and the macro. This helps to reduce the wirelength effectively. Similarly, pin direction (input or output) is also an important factor to find the relative position of the blocks which too help in reducing wirelength. In addition to this, the inclusion of blocks to a cluster is decided based on the routing blockage create by a macro with the help pin information. During clustering at this stage, if the information is kept along with the cluster information, the formulation of the objective function can be trickily devised which indirectly help in reducing total wirelength.

The Fig. 3b describes the effect of pin offset in the wirelength.  $P_1$ ,  $P_2$ , and  $P$  belong to the same net. The corresponding block  $x$  of  $P$  is placed right to the block  $B$ .  $P$  is close to  $P_2$  but both are input pins.  $P_1$  is an output pin. Therefore,  $P$  should be connected to  $P_1$ . So,  $P$  should be closer to  $P_1$  to reduce the wirelength. Hence, left position is the appropriate one. Based on the cluster created for macros, all the macros in a cluster is considered as a single macro. The pin offset in that macro is identified to find out the movable blocks which are connected to that. Now, based on the pin offset(s) in a movable block and the offset of the pin(s) connected, the relative position of the block and the macro is decided. This will help in formulating the objective function.

The Fig. 3b shows the importance of consideration of the input/output pin directions in deciding the relative position of the macro and standard cells. A net normally connects one output and one or more input pins. To make the placement and routing problem easy, a multi-pin net is divided into 2-pin nets where each 2-pin net has one input and one output pin. Therefore, to formulate the objective function for reducing the wirelength, pin types and their directions are important.

### B. Global Placement

A routability-aware global placement algorithm has been discussed in this section. Since global placement has a crucial role in the further stages of physical design, it is very important to consider most of the constraints that can affect the routability-driven placement. An analytical approach has been

adopted to formulate the routing problem considering various constraints one by one.

1) *Wirelength-Driven global Placement*: A simple wirelength-driven placement formulation is very naive and can be represented by the similar expression as in 16. Since, the expression is not smooth, hence do not have continuous derivatives. We have used the weighted-average wirelength model [38] to calculate the approximated value of HPWL in the 16 using quadratic programming without any constraints.

$$\text{minimize} : W(X, Y) = (x_i - x_j)^2 + (y_i - y_j)^2 \quad (4)$$

2) *Wirelength-driven Placement with bin area constraints*: Only the wirelength-driven placements are not sufficient for recent modern circuit because they always tend to fail in satisfying routability constraints. On the other hand congested area with different circuit blocks may create hindrance for local routing inside a bin as well as global routing in different routing channels. Hence, it is a necessity to consider block density during the global placement.

$$\epsilon(\text{area}(b_{ij}) - \sum(\text{area}(B_f))) > \sum(\text{area}(B_m)) \quad (5)$$

3) *Wirelength-driven Placement with Block and Pin Density*: Pin density is an another important parameter which has a great effect on the routability. A congested pin region or bin may increase the routing demand in the surrounding areas. Normally, during global placement, local routing scenarios are deferred to the detailed routing phase. But, that might create unsolved routing solution. Therefore, considering pin density during global placement helps to reduced local routing demands in G-cells and routing channels.

$$\sum \text{count\_pin}(B_i) < \gamma P_{\text{threshold}} \quad (6)$$

$$|B_m|_{b_{ij}} < \varphi B_n \quad (7)$$

4) *Routability-driven Global Placement*: Wirelength-driven global placement with area, block and pin density constraints may produce better results w.r.t routability than the simple wirelength-driven placement. But there is a chance of increase in total HPWL. In spite of that the first approach does not satisfy all the routing constraints and it is not a complete routability-driven placement algorithm. Hence, for the complete routability-driven placement, channel routing capacity and demand must be taken into consideration during global placement. Therefore, along with the simple wirelength-driven objective function, the following routability constraints is added to formulated a complete routability-driven global placement:

$$D_e < \lambda C_e, e \in Gr_E \quad (8)$$

where  $D_e$  is the routing demand in routing edge  $e$  in the routing grid graph  $Gr$ .  $C_e$  denotes the routing capacity of the edge. The set of routing edges is represented by  $Gr_E$ .  $\lambda$  is real constant to control usages of the routing edges and narrow channels. In the routing graph, routing capacity of some edges reduced because of the blockages created by large macros or

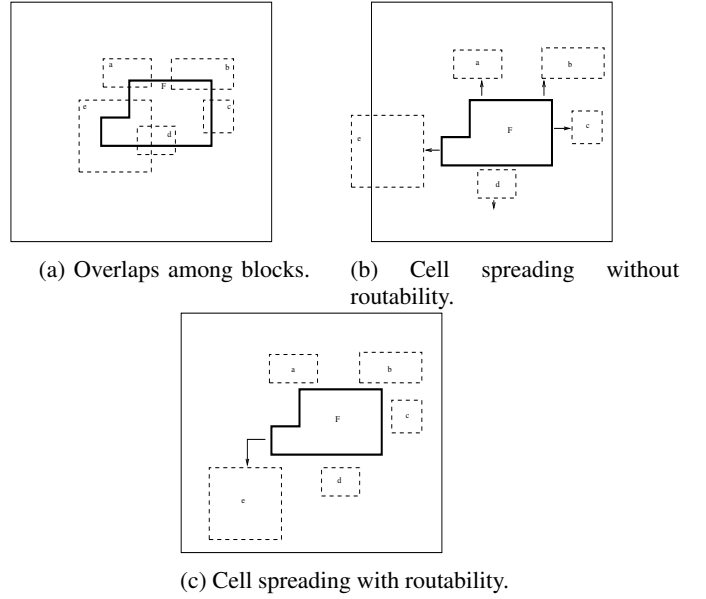


Fig. 4: Cell spreading for legalization.

fixed blocks. In other words, all the edges do not have the same routing capacity.

5) *Routability-driven global placement with pin information*: Though, the routability-driven placement formulation is more or less complete but helping the solver by assisting it by giving information of the relative pin position or offset may give more better result in terms of wirelength. So adding some constraints with the objective of the formulation for routability-driven placement for giving pin direction and offset will provide extra benefit to the solution. The following constraints are added to add pin offset or direction information for giving better direction for the relative position of the blocks in a net:

$$\text{left}(b_i, b_j) : x_i - x_j > 0 \quad (9)$$

$$\text{right}(b_i, b_j) : x_j - x_i > 0 \quad (10)$$

$$\text{above}(b_i, b_j) : y_i - y_j > 0 \quad (11)$$

$$\text{below}(b_i, b_j) : y_j - y_i > 0 \quad (12)$$

where the expressions (9) to (12) represent the relative positions right to, left to, above, and below of block  $b_i$  w.r.t block  $b_j$ .

Finally, a complete routability-driven global placement problem can be expressed as a constraint quadratic programming problem.

### C. Routability-driven legalization

After global placement, legalization is performed to remove overlaps among circuit blocks. During global placement, no constraints have been considered to restrict overlapping. Hence, a dedicated phase is implemented to apply legalization. Legalization can be of two types: normal legalization and routability-driven legalization. In normal legalization, the main objective is to remove overlaps among blocks without taking

routability into account. But, in routability-driven legalization, overlap removal is performed keeping in view routability of different routing edges.

Legalization is performed by a simple cell spreading method in this paper. Based on the pin information, cell size, type, and the number of pins in a block (cell), cells are moved left, right, above or below with respect to other cells. A fixed macro does not move, hence other movable blocks overlapping it, must move to the nearest empty spaces which causes less effect in the wirelength. If the block which is moving, crosses the bin boundary, then a gain value is calculated to find out the overall gain or loss in terms of routing congestion in nearby routing channels. The gain is calculated in the following way:

$$gain = rc_t(e_i) - rc_{t+1}(e_i) \quad (13)$$

where  $rc_t(e_i)$  and  $rc_{t+1}(e_i)$  are the routing congestion of routing edge  $e_i$  before and after the cell spreading. If the gain is positive, then crossing the bin boundary is accepted, otherwise an empty place is searched in the same bin for the block/cell. In Fig. 4, a clear explanation of the cell spreading is depicted. Fig. 4a shows an overlap among a fixed block F with five movable blocks a, b, c, d, and e. A cell spreading is applied to remove overlaps in Fig. 4b where blocks are spread according to the relative position identified during clustering. Due to normal cell spreading, block e crosses the bin boundary, then a gain value is calculated. Suppose, the value is negative, a new empty place is found in the bin and placed (Fig. 4c).

#### D. Rotability-driven detailed placement

Most of the existing techniques have used cell swapping or moving to obtain routability-driven detailed placement. These techniques are indeed helpful and required to remove routing congestion in different routing channels. The selection of cells for swapping or moving is an important factor in detailed placement. Wrong selection may increase wirelength and/or routing congestion in other routing channels.

In this paper, we have adopted cell swapping and moving approach to remove routing congestion in routing channels. We have considered all such factors which may lead to longer HPWL, congestion, pin offset etc. to produce an optimal routability-driven placement. The swapping of cells is decided based on the identification of the victimized routing channels. The congestion of a channel is calculated based on the following expression:

$$rc(e_i) = W(e_i) - \frac{CW}{WW + WS} \quad (14)$$

where  $rc(e_i)$  denotes the routing congestion or overflow of the routing channel  $e_i$ .  $W(e_i)$  is the number of connecting wires passing through  $e_i$  after global placement. CW, WW, and WS signify the channel width of  $e_i$ , wire width and wire spacing respectively. After calculating this for each channel, they are sorted in decreasing order based on the value  $rc(e_i)$  for all  $e_i$ . Now, the nets are identified in the most congested channel. Based on the physical structure of any net and the position of the corresponding cells/blocks, a cell swapping or

moving decision is taken place. If the endpoints of the net (2-pin net) are far apart and more than 90% of the blocks of the corresponding net are in closer area of an endpoint, then other block is shifted to the nearby areas of that block. This movement not only reduces the congestion, but also improves wirelength. After that, global route of the net is updated. If the wirelength is increases little and has very trivial effect, movement is accepted. If the wirelength increases high, shifting/movement does not take place. Then, any other net is taken for calculation overflow using similar process. If two endpoints reside on the opposite sides of  $e_i$ , cell swapping is used to reduce congestion.

#### IV. EXPERIMENTAL RESULTS

In this section, the experimental results of our proposed technique have been reported. We have compared the obtained results with three recent placers for mixed-size designs. All the experiments have been conducted on a Linux workstation with the 3.04GHz Intel processor and 32 GB RAM. The algorithms are implemented in the C++ language and compiled by g++ 4.8.2 compiler. The experiments have been organized on the benchmarks of ICCAD 2012 Contest [3]. As per the ICCAD Contest 2012, scaled wirelength (Scaled\_WL) and routing congestion (RC) metrics have been applied to perform routability evaluation. To estimate Routing Congestion (RC), Average Congestion of the top x% congested g-Edges (ACE(x)) and Peak Weighted Congestion (PWC) are considered from the routing result by NCTU-GR 2.0 [39] as suggested in ICCAD 2012. Following are the equations to calculate Scaled\_WL and RC:

$$Scaled\_WL = HPWL \times (1 + PF \times (RC - 100)) \quad (15)$$

$$RC = max(100, PWC) \quad (16)$$

$$PWC = \frac{\sum_x k_x \times ACE(x)}{\sum_x k_x} \quad (17)$$

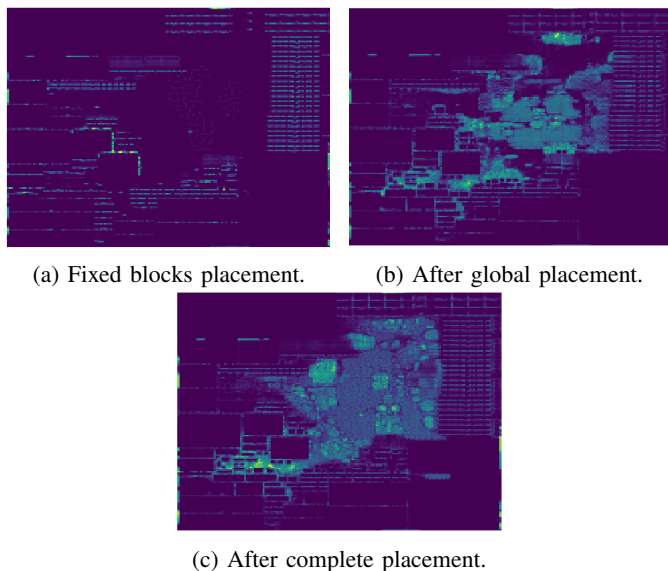
where PF is the Penalty Factor and it is set to 0.3 as per the ICCAD contest to calculate scaled\_WL from HPWL. The routing congestion RC is measured by the average of  $ACE(x)$  for different values of  $x$  i.e. 0.5, 1, 2, 5 and  $k_x = 1$  for all  $x$ .

Table I reports the ICCAD 2012 benchmark circuits. Total eight circuits are used, where each circuit is of different complexity. The column heads ‘‘Cir. Name’’, ‘‘Total Nodes’’, ‘‘#Mov. Blocks’’, ‘‘#Terminal Nodes’’, ‘‘#Terminal\_NI Nodes’’, ‘‘#Nets’’, ‘‘#Pins’’ and ‘‘#Util(%)’’ specify the circuit name, number of circuit blocks, movable blocks, fixed blocks, fixed blocks with overlapping, circuit nets, I/O pins, and utilization of space.

In Table II, comparison based results have been presented where placement results of our proposed placer MiHiPlacer (Placer for Mixed-size Hierarchy design circuits) are compared with Ripple [32] and NTUplacer4h [19] w.r.t HPWL, scaled wirelength, routing congestion, and CPU time. We have used NCTU-GR 2.0 [39] router for evaluating our placement results. The global routing results obtained from NCTU-GR 2.0 for the placement results produced by MiHiPlacer give the routing

TABLE I: Details of ICCAD 2012 Benchmark Circuits for Placement

| Cir. Name   | Total Nodes | #Mov. Blocks | #Terminal Nodes | #Terminal_NI Nodes | #Net    | #Pins   | Util.(%) |
|-------------|-------------|--------------|-----------------|--------------------|---------|---------|----------|
| superblue1  | 847441      | 765102       | 52627           | 29712              | 822744  | 2861188 | 69       |
| superblue3  | 919911      | 833370       | 55033           | 31508              | 898001  | 3110509 | 73       |
| superblue4  | 600220      | 521466       | 40550           | 38204              | 567607  | 1884008 | 70       |
| superblue5  | 772457      | 677416       | 74365           | 20676              | 786999  | 2500306 | 77       |
| superblue7  | 1364958     | 1271887      | 66995           | 26076              | 1340418 | 4935083 | 76       |
| superblue10 | 1202665     | 1045874      | 96251           | 60540              | 1158784 | 3894138 | 70       |
| superblue16 | 698741      | 680450       | 419             | 17872              | 697458  | 2280931 | 69       |
| superblue18 | 483452      | 442405       | 25963           | 15984              | 468918  | 1864306 | 67       |

Fig. 5: Heatmap of the placement results of *superblue5* benchmark circuit.

congestion information. The HPWL for the placement of each benchmark circuit is calculated after the global routing produced by NCTU-GR 2.0. Scaled wire length is approximated using the expression 15. CPU time shows the total time to find out the placement result by our proposed technique. In Fig. 5, heat maps for the benchmark *superblue5* have been shown. 5b is the placement after global placement of 5a and 5c is the heat map corresponding to placement after detailed placement. The proposed technique is better than other existing placers in most of the cases w.r.t HPWL, Scaled\_Wirelength, RC, and CPU time. Most of the existing placers use very complex steps and expensive framework to achieve placement results. The proposed technique uses very simple but smart steps to provide good results.

## V. CONCLUSION

The presented analytical based placer for mixed-size designs provides better results with respect to HPWL, scaled wirelength, routing congestion and CPU time. The technique uses pin information for global placement. A routability-driven legalization based on cell spreading technique efficiently considers routing congestion for spreading the cells. Finally, a cell swapping/movement based routability-driven detailed

placement is devised to provide a complete placement. This technique has used very simple steps to achieve better results. The results reported can be refined with iterative steps. In the next version of the paper, we will address all such issues.

## REFERENCES

- [1] N. Viswanathan, C. J. Alpert, C. Sze, Z. Li, G.-J. Nam, and J. A. Roy, "The ispd-2011 routability-driven placement contest and benchmark suite," in *Proceedings of the 2011 International Symposium on Physical Design*, ser. ISPD '11. New York, NY, USA: ACM, 2011, pp. 141–146.
- [2] N. Viswanathan, C. Alpert, C. Sze, Z. Li, and Y. Wei, "The dac 2012 routability-driven placement contest and benchmark suite," in *DAC Design Automation Conference 2012*, June 2012, pp. 774–782.
- [3] —, "Iccad-2012 cad contest in design hierarchy aware routability-driven placement and benchmark suite," in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2012, pp. 345–348.
- [4] V. Yutsis, I. S. Bustany, D. Chinnery, J. R. Shinnerl, and W.-H. Liu, "Ispd 2014 benchmarks with sub-45nm technology rules for detailed-routing-driven placement," in *Proceedings of the 2014 International Symposium on Physical Design*, ser. ISPD '14. ACM, 2014, pp. 161–168.
- [5] J. Lou, S. Krishnamoorthy, and H. S. Sheng, "Estimating routing congestion using probabilistic analysis," in *Proceedings of the 2001 International Symposium on Physical Design*, ser. ISPD '01. New York, NY, USA: ACM, 2001, pp. 112–117.
- [6] A. B. Kahng and X. Xu, "Accurate pseudo-constructive wirelength and congestion estimation," in *Proceedings of the 2003 International Workshop on System-level Interconnect Prediction*, ser. SLIP '03. New York, NY, USA: ACM, 2003, pp. 61–68.
- [7] J. Westra, C. Bartels, and P. Groeneveld, "Probabilistic congestion prediction," in *Proceedings of the 2004 International Symposium on Physical Design*, ser. ISPD '04. New York, NY, USA: ACM, 2004, pp. 204–209.
- [8] C.-w. Sham and E. F. Y. Young, "Congestion prediction in early stages," in *Proceedings of the 2005 International Workshop on System Level Interconnect Prediction*, ser. SLIP '05. New York, NY, USA: ACM, 2005, pp. 91–98.
- [9] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," in *2007 Design, Automation Test in Europe Conference Exhibition*, April 2007, pp. 1–6.
- [10] C.-L. E. Cheng, "Risa: Accurate and efficient placement routability modeling," in *Proceedings of the 1994 IEEE/ACM International Conference on Computer-aided Design*, ser. ICCAD '94. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994, pp. 690–695.
- [11] W. Hou, H. Yu, X. Hong, Y. Cai, W. Wu, J. Gu, and W. H. Kao, "A new congestion-driven placement algorithm based on cell inflation," in *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '01. New York, NY, USA: ACM, 2001, pp. 605–608.
- [12] X. Yang, R. Kastner, M. Sarrafzadeh, and M. Sarrafzadeh, "Congestion reduction during placement based on integer programming," in *Proceedings of the 2001 IEEE/ACM International Conference on Computer-aided Design*, ser. ICCAD '01. Piscataway, NJ, USA: IEEE Press, 2001, pp. 573–576.
- [13] M. Pan and C. Chu, "Fastroute: A step to integrate global routing into placement," in *2006 IEEE/ACM International Conference on Computer Aided Design*, Nov 2006, pp. 464–471.

TABLE II: Comparison based results of the placement solution of our proposed algorithm for HPWL and CPU. Global routing based congestion evaluation for Routing congestion and Scaled\_wirelength estimation evaluated by the NCTU-GR 2.0 [39] of proposed algorithm and the top placers on the ICCAD 2012 placement benchmarks [3]. The actual value for each parameter is achieved by multiplying by  $10^7$ .

| Cir. Name   | Ripple [32] |           |        |        | NTUplace4h [19] |           |        |        | MiHiPlacer |           |        |        |
|-------------|-------------|-----------|--------|--------|-----------------|-----------|--------|--------|------------|-----------|--------|--------|
|             | HPWL        | SCaled_WL | RC     | CPU    | HPWL            | SCaled_WL | RC     | CPU    | HPWL       | SCaled_WL | RC     | CPU    |
| superblue1  | 27.86       | 28.89     | 101.23 | 170.22 | 26.86           | 28.13     | 101.58 | 84.37  | 26.12      | 26.97     | 101.15 | 81.23  |
| superblue3  | 33.31       | 36.04     | 102.73 | 251.90 | 31.76           | 34.59     | 102.98 | 92.22  | 29.89      | 32.53     | 101.45 | 89.94  |
| superblue4  | 21.83       | 22.69     | 101.32 | 142.92 | 21.81           | 23.05     | 101.90 | 64.71  | 21.32      | 21.97     | 100.26 | 51.97  |
| superblue5  | 34.47       | 34.86     | 100.38 | 180.55 | 34.92           | 35.56     | 100.61 | 86.37  | 32.88      | 33.26     | 100.12 | 79.54  |
| superblue7  | 41.79       | 42.91     | 100.39 | 383.62 | 39.51           | 39.82     | 100.26 | 165.71 | 38.36      | 38.87     | 100.12 | 142.13 |
| superblue10 | 57.80       | 61.11     | 101.91 | 438.53 | 58.55           | 61.67     | 101.77 | 153.14 | 58.85      | 60.23     | 100.23 | 139.86 |
| superblue16 | 26.70       | 28.40     | 102.13 | 158.23 | 26.43           | 27.94     | 101.91 | 63.12  | 26.43      | 27.21     | 100.34 | 60.25  |
| superblue18 | 16.37       | 17.91     | 103.15 | 183.15 | 15.08           | 16.36     | 102.84 | 54.75  | 15.12      | 15.67     | 101.32 | 4687   |
| Normalized  | 1.03        | 1.03      | 1.00   | 2.51   | 1.00            | 1.00      | 1.00   | 1.00   | 1.00       | 1.00      | 1.00   | 1.00   |

- [14] J. Hu, J. A. Roy, and I. L. Markov, "Completing high-quality global routes," in *Proceedings of the 19th International Symposium on Physical Design*, ser. ISPD '10. New York, NY, USA: ACM, 2010, pp. 35–41.
- [15] W.-C. K. Y.-L. L. Wen-Hao Liu, Ke-Ren Dai, *Trnsitor*, 2012. [Online]. Available: <https://people.cs.nctu.edu.tw/~whliu/NCTU-GR.htm>
- [16] M.-C. Kim, J. Hu, D.-J. Lee, and I. L. Markov, "A simplr method for routability-driven placement," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11. Piscataway, NJ, USA: IEEE Press, 2011, pp. 67–73.
- [17] X. He, T. Huang, L. Xiao, H. Tian, and E. F. Y. Young, "Ripple: A robust and effective routability-driven placer," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1546–1556, Oct 2013.
- [18] T. Lin and C. Chu, "Polar 2.0: An effective routability-driven placer," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2014, pp. 1–6.
- [19] X. He, W.-K. Chow, J. Kuang, K.-C. Lam, W. Cai, and E. F. Young, "Ripple 2.0: High quality routability-driven placement via global router integration," in *Proceedings of the 2013 ACM/EDAC/IEEE Design Automation Conference (DAC)*, ser. DAC '13. New York, NY, USA: ACM, 2013, pp. 1–6.
- [20] P. N. Parakh, R. B. Brown, and K. A. Sakallah, "Congestion driven quadratic placement," in *Proceedings of the 35th Annual Design Automation Conference*, ser. DAC '98. New York, NY, USA: ACM, 1998, pp. 275–278.
- [21] U. Brenner and A. Rohe, "An effective congestion driven placement framework," in *Proceedings of the 2002 International Symposium on Physical Design*, ser. ISPD '02. New York, NY, USA: ACM, 2002, pp. 6–11.
- [22] J. A. Roy and I. L. Markov, "Seeing the forest and the trees: Steiner wirelength optimization in placement," *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 26, no. 4, pp. 632–644, Apr. 2007.
- [23] C. Li, M. Xie, C.-K. Koh, J. Cong, and P. H. Madden, "Routability-driven placement and white space allocation," in *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, Nov 2004, pp. 394–401.
- [24] K. Tsota, C.-K. Koh, and V. Balakrishnan, "Guiding global placement with wire density," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, Nov 2008, pp. 212–217.
- [25] A. B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 5, pp. 734–747, May 2005.
- [26] Z.-W. Jiang, B.-Y. Su, and Y.-W. Chang, "Routability-driven analytical placement by net overlapping removal for large-scale mixed-size designs," in *Proceedings of the 45th Annual Design Automation Conference*, ser. DAC '08. New York, NY, USA: ACM, 2008, pp. 167–172.
- [27] Y. L. Chuang, G. J. Nam, C. J. Alpert, Y. W. Chang, J. Roy, and N. Viswanathan, "Design-hierarchy aware mixed-size placement for routability optimization," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 663–668.
- [28] J. Hu, M. C. Kim, and I. L. Markov, "Taming the complexity of coordinated place and route," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, May 2013, pp. 1–7.
- [29] X. Yang, B.-K. Choi, and M. Sarrafzadeh, "Routability-driven white space allocation for fixed-die standard-cell placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 4, pp. 410–419, Apr 2003.
- [30] U. Brenner and A. Rohe, "An effective congestion-driven placement framework," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 4, pp. 387–394, April 2003.
- [31] J. A. Roy, N. Viswanathan, G. J. Nam, C. J. Alpert, and I. L. Markov, "Crisp: Congestion reduction by iterated spreading during placement," in *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, Nov 2009, pp. 357–362.
- [32] M. K. Hsu, Y. F. Chen, C. C. Huang, S. Chou, T. H. Lin, T. C. Chen, and Y. W. Chang, "Ntuplace4h: A novel routability-driven placement algorithm for hierarchical mixed-size circuit designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1914–1927, Dec 2014.
- [33] X. He, T. Huang, L. Xiao, H. Tian, G. Cui, and E. F. Y. Young, "Ripple: An effective routability-driven placer by iterative cell movement," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11. Piscataway, NJ, USA: IEEE Press, 2011, pp. 74–79.
- [34] M.-K. Hsu, S. Chou, T.-H. Lin, and Y.-W. Chang, "Routability-driven analytical placement for mixed-size circuit designs," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11. Piscataway, NJ, USA: IEEE Press, 2011, pp. 80–84.
- [35] M. Pan and C. Chu, "Ipr: An integrated placement and routing algorithm," in *2007 44th ACM/IEEE Design Automation Conference*, June 2007, pp. 59–62.
- [36] Y. Zhang and C. Chu, "Crop: Fast and effective congestion refinement of placement," in *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, Nov 2009, pp. 344–350.
- [37] W.-H. Liu, C.-K. Koh, and Y.-L. Li, "Optimization of placement solutions for routability," in *Proceedings of the 50th Annual Design Automation Conference*, ser. DAC '13. New York, NY, USA: ACM, 2013, pp. 153:1–153:9.
- [38] M.-K. Hsu, V. Balabanov, and Y.-W. Chang, "Tsv-aware analytical placement for 3-d ic designs based on a novel weighted-average wirelength model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 4, pp. 497–509, Apr 2013.
- [39] W. H. Liu, W. C. Kao, Y. L. Li, and K. Y. Chao, "Nctu-gr 2.0: Multithreaded collision-aware global routing with bounded-length maze routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 709–722, May 2013.