Estimating Routing Congestion using Probabilistic Analysis

Jinan Lou
Synopsys Inc.
700 E. Middlefield Road
Mountain View, CA 94043-4033
1-650-584-1239
jlou@synopsys.com

Shankar Krishnamoorthy
Synopsys Inc.
700 E. Middlefield Road
Mountain View, CA 94043-4033
1-650-584-4406
shankar@synopsys.com

Henry S. Sheng
Synopsys Inc.
700 E. Middlefield Road
Mountain View, CA 94043-4033
1-650-584-4773
hsheng@synopsys.com

ABSTRACT

Design routability is a major concern in the ASIC design flow, particularly with today's increasingly aggressive process technology nodes. Increased die areas, cell densities, routing layers, and net count all contribute to complex interconnect requirements, which can significantly deteriorate performance, and sometimes lead to unroutable solutions. Congestion analysis and optimization must be performed early in the design cycle to improve routability. This paper presents a congestion estimation algorithm for a placed netlist. We propose a net-based stochastic model for computing expected horizontal and vertical track usage, which considers routing blockages. The main advantages of this algorithm are accuracy and fast runtime. We show that the congestion estimated by this algorithm correlates well with post-route congestion, and show experimental results of subsequent congestion optimization based this algorithm.

I. INTRODUCTION AND BACKGROUND

Congestion is a supply and demand problem for routing resources. During placement, cells are placed within the design boundary to minimize cost functions, typically based on total wire length. Wires between cells are then connected during the routing step. Design densities in deep submicron technologies can be severe, which result in major escalations in routing demands. Unfortunately, minimizing total wire length has no direct impact on routability. Consequently, in the absence of strong congestion analysis and removal techniques, routing demand in industrial designs can quickly exhaust supply. Congestion deteriorates design performance because of detoured nets, and can lead to unroutable solutions. Congestion also deteriorates the ability to make eco changes, and constrains the flexibility of routing to optimize secondary objectives (e.g. via count, crosstalk, antenna rules). When facing routability issues, designers typically must enlarge their floorplans, implying added expense and schedule delay. Worse yet, there is no guarantee that a new floorplan will yield a satisfactory solution. Substantial time and resources can be saved by measuring congestion earlier in the design cycle. This congestion analysis needs to be accurate and yet fast enough to include in the inner loop of synthesis/placement optimizations.

The supply of routing resources can be computed from the technology parameters of the design, such as number of available routing layers, minimum pitch and routing direction of each layer. The demand for routing resources depends on the floorplan, synthesis, placement and routing solutions. However, fully accurate routing demand is not available until after detailed routing, which is too late to make significant changes. Therefore, estimation algorithms are required for congestion analysis during earlier phases of the design. Previous work in this area use either empirical models [1][9] or global routers to model congestion [4][5][8][10]. Most empirical models need design-specific tuning, and do not correlate well to the real congestion. There are two problems with global router based solutions. First, the runtime penalty is high, especially if multiple estimates are required during optimization. Second, the accuracy of a router-based congestion estimate cannot be guaranteed if a different router is used as the final routing tool. In addition, most congestion estimation algorithms fail to handle routing blockages.

This paper presents a stochastic congestion estimation algorithm for a placed netlist that is blockage-aware. We show that this congestion model correlates well to post-route results. The runtime of this algorithm is fast, thus making it an ideal candidate to be used within the inner loop of synthesis and placement optimizations. Its stochastic nature promotes robustness against the implementation details of downstream routing algorithms.

This paper is organized as follows: we present the details of our probabilistic congestion model in section II; the presence of routing blockages is discussed in section III; finally, experimental results and concluding remarks are given in section IV and V.

II. CONGESTION ESTIMATION

II.1. Premises

Given a placed netlist, we discretize the core area with a homogeneous rectangular grid (c.f. Figure 1). We then analyze the congestion for every grid in the mesh. The number of grids in the mesh can be either a fixed number, or a variable that depends on the technology parameters.

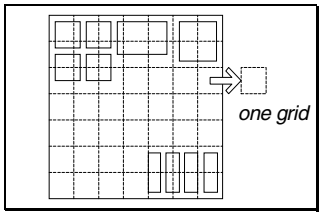

Figure 1. The grid mesh of a design

Definition 1: The *capacity* of a grid is defined as the number of available routing tracks within the grid. The *horizontal capacity* of a grid is defined as the number of available horizontal routing tracks, and the *vertical capacity* of a grid is defined as the number of available vertical routing tracks.

ISPD'01, April 1-4, 2001, Sonoma, California, USA.

Definition 2: The *usage* of a grid is defined as the number of used routing tracks within the grid. The *horizontal usage* of a grid is defined as the number of used horizontal routing tracks, and the *vertical usage* of a grid is defined as the number of used vertical routing tracks.

Definition 3: The *horizontal congestion cost* of a grid is defined as the ratio between the horizontal usage and the horizontal capacity of the grid. The *vertical congestion cost* of a grid is defined as the ratio between the vertical usage and the vertical capacity of the grid.

Obviously, if a congestion cost is larger than 100%, the grid is congested for that direction. The larger the cost is, the worse the congestion will be.

The capacities of the grids, which are the supply part of the congestion analysis, can be computed easily from the dimensions of the grid and the technology parameters. Assume the number of horizontal routing layers is N^h, the number of vertical routing layers is N^v, and the minimum pitches for the i^{th} horizontal and vertical layer are L^h_i and L^v_i, respectively. Also assume the width and height of each grid are W and H. The following equations compute the horizontal and vertical capacities of a grid:

$$horizontal\_capacity = H \times \sum_{i=1}^{N^h} \left(1 / L^h_i \right) \quad (1)$$

$$vertical\_capacity = W \times \sum_{i=1}^{N^v} \left(\frac{1}{L^v_i} \right) \quad (2)$$

Our probabilistic congestion model will be used to compute the usages of the grids, which represent routing resource demand. We make the following general assumptions for this model:

- *All nets are optimally routed with shortest length*
- *All nets make at most one change of direction per grid*
- *No change of direction is allowed in the grid with pins unless there are more than one pins in the same row or column of the grid*

The first assumption restricts the routes to be within the net bounding box, unless there are routing blockages that prohibit this. The second and third assumptions simplify our model. For any optimally routed net, the amount of consumed routing resources is independent of the number of turns the net makes. By restricting how the change of direction is accomplished, we are able to reduce the solution space while maintaining the accuracy of our model. If a net is allowed to change its direction at most once per grid, probabilistically, it will change the direction in the middle of the grid. For example, there are five vertical tracks in the grid shown in Figure 2A. When the net needs to change the direction from horizontal to vertical, it can use either one of them with the same probability. The expected horizontal and vertical usages for this net within the grid are the same as if the net always changes its direction in the middle of the grid. If the grid is on the same row or column of one pin, assumption three requires that the change of direction shall occur in the place where a straight line can be drawn to that pin (c.f. Figure 2B).

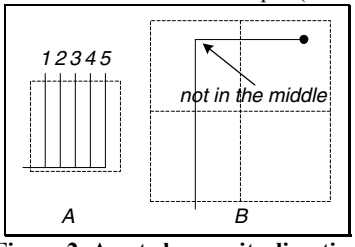

Figure 2. A net changes its direction

II.2. The basic model for two-pin nets

We first introduce our basic model for any two-pin net with the following restrictions:

- *The pins are located at the lower left and upper right corners*
- *The bounding box of the net covers at least two rows and two columns*
- *There is no routing blockages within the bounding box*

The extensions to multi-pin nets will be discussed in section II.5. We will relax the first restriction by allowing pins to be located anywhere within the grid in section II.3. In section II.4, we present the solutions to special cases where the bounding box of the net covers at most one row or one column. Congestion analysis with the presence of routing blockages are explained in section III.

II.2.1. A simple example

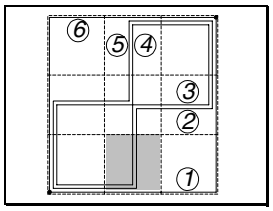

Figure 3. A two-pin net covering a 3×3 mesh

We introduce the probabilistic congestion model with a simple example. Figure 3 illustrates a two-pin net that covers a 3×3 mesh. The two pins are at the lower left and upper right corners of the mesh, respectively. There are six possible ways to route these pins as shown in this figure. Assuming that each route would happen with the same probability, we compute the probabilistic track usages for the vertical and horizontal directions of each grid in this mesh. Let us use the lower middle grid as an example. There are three routes going through this grid (labeled as routes 1, 2 and 4). Route 1 is a complete horizontal route. Therefore, it consumes one entire horizontal track, but no vertical track. Routes 2 and 4 change the direction from horizontal to vertical in this grid. As explained in section II.1, the knees of the routes will be in the middle of the grid. Therefore, both of them consume one entire vertical track, but half of a horizontal track. Therefore, the total usages for this grid are two horizontal tracks and two vertical tracks. Because the total number of possible routes is six, the probabilistic usages for this net in this grid are one third of a track for both the horizontal and vertical directions. The following matrix summarizes the probabilistic track usages of this net for the horizontal and vertical directions:

$$\frac{1}{6} \times \begin{bmatrix} (1\ \ 1) & (2\ \ 2) & (3\ \ 3) \\ (2\ \ 2) & (2\ \ 2) & (2\ \ 2) \\ (3\ \ 3) & (2\ \ 2) & (1\ \ 1) \end{bmatrix}$$

II.2.2. The total number of possible routes

Definition 4: We define $F(m, n)$ as the total number of possible ways to optimally route a two-pin net covering an $m \times n$ mesh.

Lemma 1: $F(m,1) = F(1,n) = 1$ (3)

Proof: If there is only one row or column, assumption three in section II.1 is no longer applicable. Therefore, we follow the same assumption of other nets such that the route will change direction in the middle of the grid. In this case, we have only one possible route when there is only one row or column (c.f. Figure 4). Therefore, $F(m,1)$ and $F(1,n)$ are always 1. ■

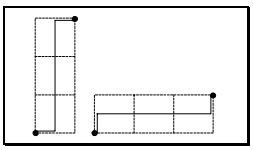

Figure 4. $F(m, n)$ with one row or one column

Theorem 1: Assume $m,n \geq 2$, $F(m, n)$ can be computed using following equation:

$$F(m,n) = F(m-1,n) + F(m,n-1) \quad \textbf{(4)}$$

Proof: Consider the lower left grid where one of the two pins is located (c.f. Figure 5). In order to optimally route these two pins, all routes must exit this grid from either the top or the right hand side of the grid. These two sets of routes are mutually exclusive. In addition, the routes that exit from the top of the grid cover a $(m\text{-}1)\times n$ mesh, and the routes that exit from the right cover an $m\times(n\text{-}1)$ mesh. The total number of routes is the summation of these two values; therefore, we proved the above equation. ■

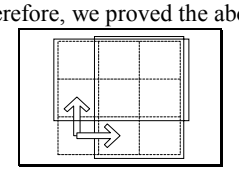

Figure 5. Computing $F(m, n)$

Corollary 1: $F(m,n) = F(n,m)$ (5)

Proof: The proof is trivial from (3) and (4). ■

Theorem 2: If $m \geq n$, $F(m, n)$ can be computed as:

$$F(m,n) = \begin{cases} 1 & n = 1 \\ m & n = 2 \\ \sum_{i_{n-2}=1}^{m} \sum_{i_{n-3}=1}^{i_{n-2}} \cdots \sum_{i_1=1}^{i_2} i_1 & n \geq 3 \end{cases} \quad (6)$$

Proof: The proof can be obtained by induction. ■

II.2.3. The probabilistic usage matrix

Definition 5: Assume the pins are located at the lower left and upper right corners of the bounding box, we define the *probabilistic usage matrix P* for a two-pin net that covers a $m\times n$ mesh as follows:

$$P(m,n) = \begin{bmatrix} (P_x(m,1) \;\; P_y(m,1)) & \cdots & (P_x(m,n) \;\; P_y(m,n)) \\ \vdots & \cdots & \vdots \\ (P_x(1,1) \;\; P_y(1,1)) & \cdots & (P_x(1,n) \;\; P_y(1,n)) \end{bmatrix}$$

where $P_x(i,j)$ and $P_y(i,j)$ represent the probabilistic horizontal and vertical usages for this net in grid (i, j).

The P matrix has the following properties:

$$P_{x|y}(i,j) = P_{x|y}(m-i+1,\ n-j+1) \quad (7)$$

$$\sum_{j=1}^{n} P_y(i,j) = 1 \ \ \forall i \quad (8)$$

$$\sum_{i=1}^{m} P_x(i,j) = 1 \ \ \forall j \quad (9)$$

(7) indicates that the entries in the P matrix are images of each other with respect to the center of the matrix. This symmetry is because those two pins in this mesh are equivalent. Therefore, we only need to compute the entries in the lower triangle of the matrix, and the entries in the upper triangle of the matrix can be copied from the corresponding entry in the lower triangle. Moreover, all optimal routes for this net consume exactly m vertical tracks and n horizontal tracks. (8) and (9) indicate that the routes will consume exactly one vertical track per row, and one horizontal track per column.

Theorem 3: The (lower triangle) of P matrix can be computed as following:

$$P_x(i,j) = \frac{1}{F(m,n)} \times \begin{cases} F(m,n-1) & \text{case } \textit{a}\text{: } i=1, j=1 \\ 1 & \text{case } \textit{b}\text{: } i=1,\ j=n \\ F(m-i+1,n-1) & \text{case } \textit{c}\text{: } 1<i<m,\ j=1 \\ \dfrac{F(m,n-j+1)+F(m,n-j)}{2} & \text{case } \textit{d}\text{: } i=1,\ 1<j<n \\ \dfrac{F(i,j)F(m-i+1,n-j)+F(i,j-1)F(m-i+1,n-j+1)}{2} & \end{cases} \quad (10)$$

$$P_y(i,j) = \frac{1}{F(m,n)} \times \begin{cases} F(m-1,n) & \text{case } \textit{a}\text{: } i=1,\ j=1 \\ 1 & \text{case } \textit{b}\text{: } i=1,\ j=n \\ \dfrac{F(m-i+1,n)+F(m-i,n)}{2} & \text{case } \textit{c}\text{: } 1<i<m,\ j=1 \\ F(m-1,n-j+1) & \text{case } \textit{d}\text{: } i=1,\ 1<j<n \\ \dfrac{F(i,j)F(m-i,n-j+1)+F(i-1,j)F(m-i+1,n-j+1)}{2} & \end{cases} \quad (11)$$

Proof: The total number of possible routes is $F(m,n)$. Therefore, this term becomes the common denominator for all expressions. For simplicity, we assume that we will route from the lower left pin to the upper right pin. We divide the grids into five different categories depending on the location of the grid within the mesh:

Case *a*: At the lower left corner (where i=1 and j=1), all routes must leave the grid from either the top or the right hand side of the grid. The number of routes leaving from the top is $F(m\text{-}1,n)$, and each of them consumes one vertical track and no horizontal track. Similarly, the number of routes leaving on the right is $F(m,n\text{-}1)$, and each of them consumes one horizontal track and no vertical track. Therefore, the horizontal and vertical usages for this grid are $F(m,n\text{-}1)$ and $F(m\text{-}1,n)$, respectively.

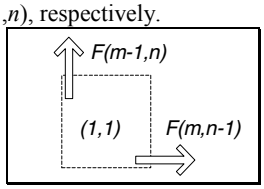

Figure 6. The lower left grid

Case *b*: At the lower right corner (where i=1 and j=n), there is only one route going through it (c.f. Figure 7). This route uses one horizontal and one vertical track. Therefore, the horizontal and vertical usages for this grid are both 1.

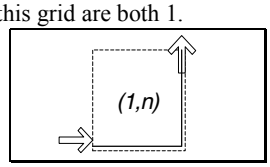

Figure 7. The lower right grid

Case *c*: For the rest of the grids in the first row (where i=1 and $1<j<n$), the routes enter from the left side of the grid, and leaves from either the top or the right hand side of the grid. The numbers of routes that enter the grid from left, leave the grid from top, and leave the grid from right are $F(m,n\text{-}j+1)$, $F(m\text{-}1,n\text{-}j+1)$ and $F(m,n\text{-}j)$, respectively. All routes going from left to right consume one horizontal track, but no vertical track. The routes going from left to top will make the turn in the middle of the grid, and each of them consumes one vertical track and half of a horizontal track. Therefore, the horizontal track usage for all routes passing through this grid is $F(m,n\text{-}j)+F(m\text{-}1,n\text{-}j+1)/2 = (F(m,n\text{-}j+1)+F(m,n\text{-}j))/2$, and the total vertical track usage is $F(m\text{-}1,n\text{-}j+1)$.

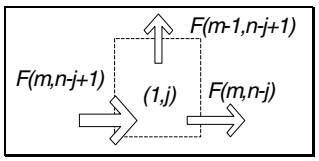

Figure 8. Grids in the first row

Case *d*: Similar to case *c*, for the rest of the grids in the first column (where $1<i<m$ and $j=1$), the routes enter from the bottom of the grid, and leaves from either the top or the right hand side of the grid. Every route going from bottom to top consumes one vertical track, but no horizontal track. Every route going from bottom to right consumes one horizontal track and half of a vertical track. Therefore, the horizontal and vertical track usages for this grid are $F(m-i+1,n-1)$ and $(F(m-i+1,n)+F(m-i,n))/2$, respectively.

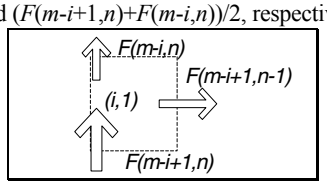

Figure 9. Grids in the first column

Case *e*: For the rest of the grids (in the middle), routes enter the grid from either left or bottom, and leave the grid from either top or right hand side of the grid. There are four different types of routes depending on their entry and departure sides. The following table summarizes these four types of routes for the number of routes and their track usages on the horizontal and vertical directions:

Type	Entry	Depart	# of routes	Hor	Ver
1	Left	Right	$F(i,j-1)\times F(m-i+1,n-j)$	1	0
2	Left	Top	$F(i,j-1)\times F(m-i,n-j+1)$	0.5	0.5
3	Bottom	Right	$F(i-1,j)\times F(m-i+1,n-j)$	0.5	0.5
4	Bottom	Top	$F(i-1,j)\times F(m-i,n-j+1)$	0	1

Table 1. Types of routes for a grid in the middle

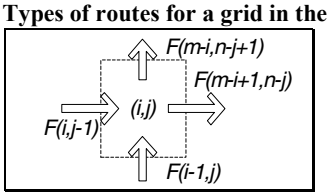

Figure 10. Grids in the middle

Types 1, 2 and 3 consume horizontal tracks. The total horizontal track usage for this grid is:

$$F(i,j-1)F(m-i+1,n-j)+\frac{F(i,j-1)F(m-i,n-j+1)}{2}+\frac{F(i-1,j)F(m-i+1,n-j)}{2}$$
$$=\frac{F(i,j)F(m-i+1,n-j)+F(i,j-1)F(m-i+1,n-j+1)}{2}$$

Types 2, 3 and 4 consume vertical tracks. The total vertical track usage for this grid is:

$$F(i-1,j)F(m-i,n-j+1)+\frac{F(i-1,j)F(m-i+1,n-j)}{2}+\frac{F(i,j-1)F(m-i,n-j+1)}{2}$$
$$=\frac{F(i,j)F(m-i,n-j+1)+F(i-1,j)F(m-i+1,n-j+1)}{2}$$

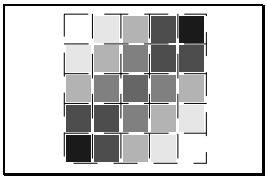

Figure 11. Track usages in a 5×5 mesh

Figure 11 shows the *P* matrix for a 5×5 grid in a color-coded manner. Darker color represents higher value of usages. The corner grids where pins are located are the darkest, because any route must go through these two grids.

II.3. Off-grid pins

(10) and (11) assume that pins are located at the lower left and upper right corners of the mesh. In this section, we relax this restriction by allowing pins to be located anywhere within the grid. We call these kinds of pins *off-grid pins*. Let us assume that the width and height of each grid are *W* and *H*, and the distances from the pins to the grid edges are d_{x1}, d_{y1}, d_{x2}, d_{y2}, respectively, as shown in Figure 12. Off-grid pins only affect the usages on the outside ring of the mesh. The usages for the grids in the middle will remain the same. Because of the asymmetry of the pin locations, the usages for the last row and last column must be computed instead of copied from their corresponding entries in the first row and first column. The horizontal usages for the first and last row are scaled down by d_{x1}/W and d_{x2}/W, respectively, and the vertical usages are scaled down by d_{y1}/H and dy_2/H, respectively.

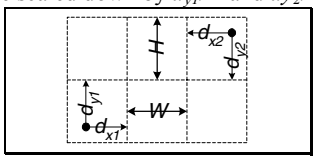

Figure 12. Off-grid pins

II.4. Short nets and flat nets

If both pins are located in the same grid, we call the net a *short net*. Similarly, if both pins are located in the same row or column, we call the net a *flat net*.

For a short net, the horizontal and vertical probabilistic usages are $|d_{x1}+d_{x2}-W|/W$ and $|d_{y1}+d_{y2}-H|/H$, respectively.

For a flat net whose pins are in the same column, the horizontal usage is $|d_{x1}+d_{x2}-W|/2W$ for the first and last row, and 0 otherwise. The vertical usage is d_{y1}/H for the first row, d_{y2}/H for the last row, and 1 otherwise.

For a flat net whose pins are in the same row, the horizontal usage is d_{x1}/W for the first column, d_{x2}/W for the last column, and 1 otherwise. The vertical usage is $|d_{y1}+d_{y2}-H|/2H$ for the first and last column, and 0 otherwise.

II.5. Multi-pin nets

The two-pin net model can be extended to model multi-pin nets. We construct either the minimum spanning tree (MST) or the rectilinear steiner tree (RST) of the multi-pin net, and then use the two-pin net model for each pair of connected pins to compute the probabilistic usages. MST algorithms run faster, but might double-count the overlapping net segments. This presents a tradeoff between runtime and accuracy. Therefore, MST based congestion estimation can be used during early optimization steps, such as location based logic synthesis and coarse placement, where a faster turnaround time is appreciated. RST based congestion estimation can be used in later stages such as placement legalization.

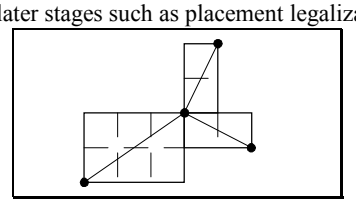

Figure 13. Multi-pin nets

III. BLOCKAGES

If we regard the routes as water flows in a river, routing blockages are like stones. They define regions with reduced routing resources. There are two types of routing blockage: partial or complete. Partial blockages block a certain number of layers, but there are still limited routing resources available. Complete blockages block all the layers, and require all routes must de-tour around them. Partial blockages can be easily modeled by eliminating the blocked layers during the capacity computation for the grids. In this section, we discuss techniques to work with complete routing blockages.

III.1. Simple blockages

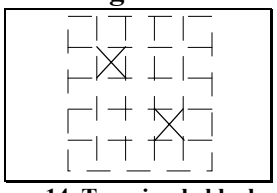

Figure 14. Two simple blockages

We define isolated routing blockages in the grid as *simple blockages*. Figure 14 shows a 5×5 mesh with two simple blockages. These routing blockages have two effects. First, the usages in the blocked grids are always zero. Second, these usages are distributed to the neighboring grids. However, they are not distributed evenly to them. The grids that are, 1) closer to the blocked grid, and 2) have more unblocked neighboring grids, have the higher probability to receive the distributed usages. Assume the distance between a candidate unblocked grid and the blocked grid is d, and the number of unblocked neighboring grids for the candidate grid is n. We compute the weight for the grid as $w=2^{-d}\times n$, and use this weight to proportionally distribute the usage of the blocked grid to all candidate grids. In practice, we only consider unblocked grids whose distance to the blocked grid is within a pre-defined number D.

Figure 15 shows the probabilistic usages of the grid shown in Figure 14. D is defined as 1 in this example. Comparing with Figure 11, the usage of the grid in the middle of the mesh has increased significantly because both blockages distribute their usages to this grid.

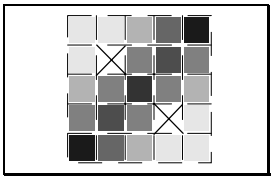

Figure 15. Track usages with two simple blockages

III.2. Line blockages

If one complete row or column of the mesh is blocked by a routing blockage, we call it a *line blockage*. In Figure 16A, we show a routing blockage that blocks four grids. For the two nets shown in this figure, their original bounding boxes have a line blockage because the middle row is completely blocked. In this case, no route can be completed within the bounding box. We need to extend the bounding box to include at least one unblocked grid in that row to bypass the blockage. For the net on the left hand side, the unblocked grid is found by extending the bounding box one grid to the left. Similarly, for the net on the right hand side, the unblocked grid is found by extending the bounding box one grid to the right. The computation of the track usages will then be based on the assumption that there is one possible route.

If an unblocked grid can be found by extending the bounding box to either direction, we choose the direction with the minimum expansion. However, if both directions have the same expansion factor (c.f. Figure 16B), we extend the bounding box in both directions. In this case, the computation of the track usages will be based on the assumption that there are two possible routes.

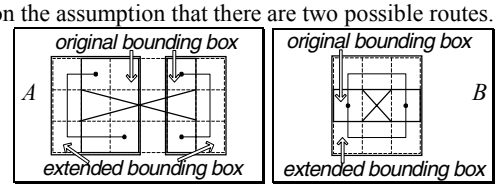

Figure 16. A complete blockages

III.3. Adjacent blockages

If blockages are adjacent to each other, we first find the minimum bounding box that covers all the adjacent blockages. If the new bounding box is isolated, we treat all the blocked grids as simple blockages, and distribute the usages to the free grids as described in section III.1. If the new bounding box is a line blockage, we will use the same techniques in section III.2 to estimate the routings.

III.4. Complex blockages

If a routing blockage cannot be processed using any of the above techniques, we have to use a maze routing algorithm [7] to find a route between the source and the destination. Fortunately, in real designs, the number of this kind of blockages are relatively small. This is because that almost all complete routing blockages are found on top of macro cells, and most macro cells are placed along the edges of the core area. The blocked grids along the edges of the core area can be handled efficiently using techniques in section III.1 and III.2. This makes the runtime overhead of the maze router a negligible factor for the overall runtime.

IV. EXPERIMENTAL RESULTS

IV.1. Correlation

We present the correlation between the predicted congestion versus the actual congestion seen by routers as congestion maps. A congestion map visually plots the congestion in the design by assigning different colors to different congestion costs. A lighter color means higher congestion cost. Most commercial routing tools are able to produce a congestion map.

We include two industrial designs to report our experimental results. Table 2 summarizes the basic statistics of these designs.

	# of instances	# of nets	CPU time
Design1	316K	332K	70s
Design2	347K	374K	110s

Table 2. Two industrial designs

In Figure 17, we present the congestion maps for two industrial designs. The congestion map on the left hand side is the congestion predicted by our probabilistic model, and the one on the right is created by a third-party commercial router. The comparison between these two congestion maps shows that we are able to predict the congestion hot spots in a fraction of the runtime of invoking a global router.

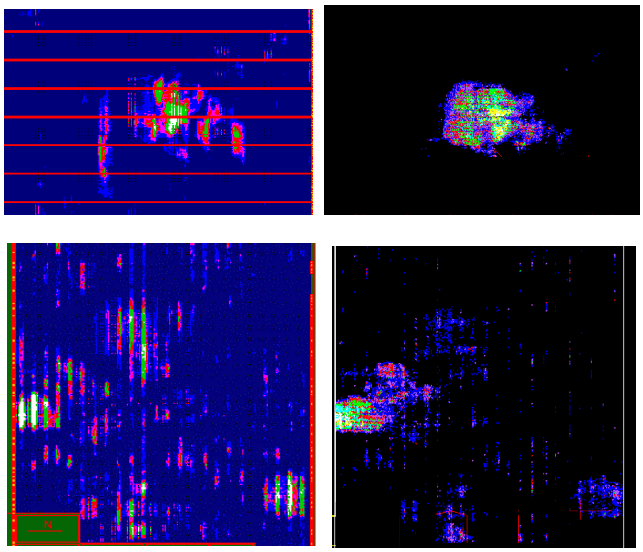

Figure 17. Congestion Correlation

IV.2. Congestion Removal

Many techniques can be used to remove congestion based on congestion estimation. Congestion cost can be included as a part of the cost function of the location based logic optimizer and the placer [2][5][6][8]; synthesis based congestion removal techniques can be performed during placement; and finally, incremental congestion removal algorithms [10] can be used to remove local hot spots in a post-placement design. The details of these techniques are beyond the scope of this paper.

In Figure 18, we present the congestion maps with and without the congestion removal techniques for the two industrial designs shown in Figure 17. All the congestion optimization techniques require multiple congestion estimations during various stages of the optimization. The congestion map on the left hand side is the result without congestion removal techniques, and the one on the right is with congestion removal techniques using the probabilistic congestion estimation model. With the accurate congestion prediction, we are able to remove most of the congestion in the design. Note that for the second design, the placement without congestion removal is not routable.

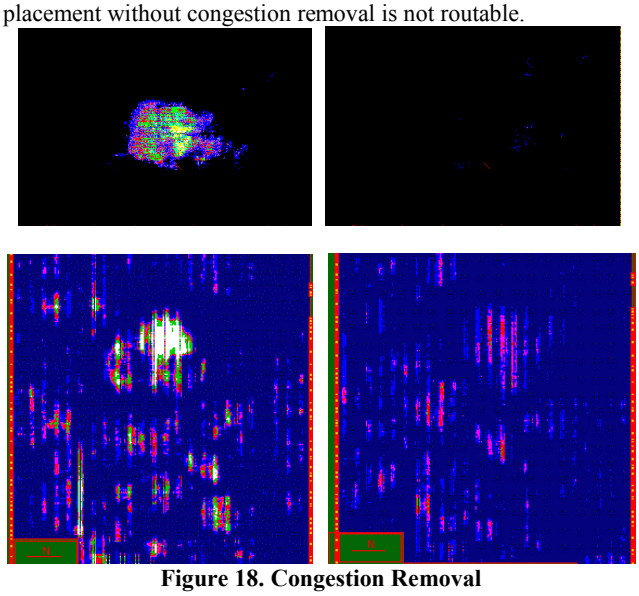

Figure 18. Congestion Removal

V. CONCLUSION

In this paper, we present a fast and accurate congestion estimation algorithm for a placed netlist. This algorithm is based on the supply and demand analysis of routing resources, where the supply is determined by technology parameters, and the demand is computed by a probabilistic congestion model. For every net in the design, the probabilistic model considers all possible ways that a router can route the net, and then computes the probabilistic track usages for this net. With extreme fast runtime and good post-route correlation, this algorithm can be used as the core congestion estimation for many congestion optimization techniques.

VI. REFERENCES

[1] C-L. E. Cheng, "RISA: Accurate and Efficient Placement Routability Modeling," *Proceedings of International Conference on Computer Aided Design,* pages 690-695, 1994

[2] J. M. Kleinhans, S. Sigl, F. M. Johannes and K. J. Antreich, "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization," *IEEE Transactions on Computer Aided Design,* Vol. 10, No. 3, pages 356-365, 1991

[3] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, Massachusetts, 1992

[4] S. Mayrhofer and U. Lauther, "Congestion-Driven Placement Using a New Multi-partitioning Heuristic," *Proceedings of International Conference on Computer Aided Design,* pages 332-335, 1990

[5] P. N. Parakh, R. B. Brown and K. A. Sakallah, "Congestion Driven Quadratic Placement," *Proceedings of 35th Design Automation Conference,* pages 275-278, 1998

[6] C. Sechen, and A. Sangiovanni-Vincentelli, "TimberWolf 3.2: A new standard cell placement and global routing package," *Proceedings of 23rd Design Automation Conference*, pages 432-439, 1986

[7] N. Sherwani, *Algorithms for VLSI Physical Design Automation,* 3rd edition, Kluwer Academic Publishers, Boston, Dordrecht, London, 2000

[8] R. S. Tsay, S. C. Chang and J. Thorvaldson, "Early Wireability Checking and 2-D Congestion-Driven Circuit Placement," *Proceedings of Fifth Annual IEEE International ASIC Conference and Exhibit,* pages 50-53, 1992

[9] M. Wang and M. Sarrafzadeh, "On The Behavior of Congestion Minimization During Placement," *International Symposium on Physical Desig*n, pages 145-150, 1999

[10] M. Wang and M. Sarrafzadeh, "Modeling and Minimization of Routing Congestion," *Proceedings of Asian-Pacific Design Automation Conference,* pages 185-190, 2000