

Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning

Probir Sarkar and Cheng-Kok Koh, *Member, IEEE*

Abstract—In this paper, we present a repeater block planning algorithm for interconnect-centric floorplanning. We introduce the concept of *independent feasible regions* for repeaters and derive an analytical formula for their computation. We develop a routability-driven repeater clustering algorithm to perform repeater block planning based on iterative deletion. The goal is to obtain a high-quality solution for the repeater block locations so that performance-driven interconnect synthesis at the routing stage can be carried out with ease while minimizing the chip area. Experimental results show that our method increases the percentage of all global nets that meet their target delays from 67.5% to 85%. Moreover, our approach minimizes the expected routing congestion, making it easier for performance-driven routers to synthesize global nets that require the insertion of repeaters to meet timing constraints.

Index Terms—Buffer insertion, deep submicrometer, floorplanning, physical design, routing.

I. INTRODUCTION

DUE TO the continued scaling of very large scale integration (VLSI) technologies, interconnects play a dominant role in determining system performance, power, reliability, and cost. To ensure timing closure of designs, impacts of interconnects must be considered as early as possible in the design flow. Several interconnect synthesis techniques—topology construction, repeater insertion, device sizing, and wire sizing and spacing—have been studied in the literature. A comprehensive survey of these techniques can be found in [2].

Studies in [3]–[6] show repeater insertion being among the most effective methods to optimize signal delay. Without repeaters, the delay of a long resistance–capacitance RC wire is quadratic in terms of the wire length. With judicious insertion of repeaters, the delay becomes linear [7]–[9]. However, most interconnect synthesis techniques were designed for post-placement interconnect optimization. In [10], it was projected that over 700 K repeaters will be inserted in a single chip for the 70-nm technology. The insertion of that many repeaters will significantly change the floorplan and placement of a design, rendering the original floorplan and placement invalid and leading to a slow design convergence.

Manuscript received July 28, 2000. This work was supported by National Science Foundation CAREER Award CCR-9984553. This paper was presented in part at the International Symposium on Physical Design, San Diego, CA, April 2000. This paper was recommended by Associate Editor D. Hill.

P. Sarkar was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Conexant Systems, Newport Beach, CA 92660 USA.

C.-K. Koh is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

Publisher Item Identifier S 0278-0070(01)03082-2.

Floorplanning [11], [12], the first stage of the physical design process, has significant effects on overall system power, performance, and reliability. However, very few existing timing-driven floorplanning techniques consider the option of repeater insertion. An advantage of considering repeater optimization during floorplanning is that the problem size at this stage is smaller—compared to the problem size faced by place-and-route—and, hence, permits a more effective search of the design space. In [13], the floorplanner assumed that repeaters could be inserted arbitrarily in an existing floorplan. However, repeaters consume silicon resources and certain circuit blocks such as the cache may not allow the insertion of repeaters. To overcome this problem, the authors in [1] considered the insertion of blocks of repeaters in the channel regions between circuit blocks. However, the greedy clustering of repeaters into a repeater block may result in routing congestion. As pointed out by the authors in [14], it is important to perform interconnect planning for global routing during the floorplanning stage. Many designs are routing-limited; it may not be feasible to get signals to and out of repeaters due to the limitation in routing resources. Therefore, it is important to consider routing feasibility (in subsequent steps) during the global distribution of repeaters in the floorplanning step.

In this paper, we propose a routability-driven repeater block planning algorithm for interconnect-centric floorplanning. We introduce the concept of independent feasible region (IFR) for repeater insertion under delay constraint and derive an analytical formula for the computation of IFRs. All repeaters are “freely” movable within their respective IFRs without violating the timing constraints. This has the advantage that each repeater of a net has equal flexibility of position, which facilitates global optimization.

As in [1], we cluster individual repeaters into repeater blocks to form a regular layout structure for a higher density implementation. We develop an effective routability-driven repeater block planning algorithm. A tile-based congestion model is used to drive an *iterative deletion* heuristic for repeater clustering. Experimental results show that our method can boost the completion rate, i.e., the percentage of all global nets that meet their target delays, to 85% from the 67.5% completion rate reported in [1]. Furthermore, the expected congestion due to our repeater block planning algorithm is lower than that produced by a planner without considering congestion.

The remainder of the paper is organized as follows. In Section II, we present the problem formulation. Section III defines the IFR for repeater insertion. Section IV presents the routability-driven repeater block planning algorithm. Experimental results are shown in Section V, followed by the conclusion in Section VI.

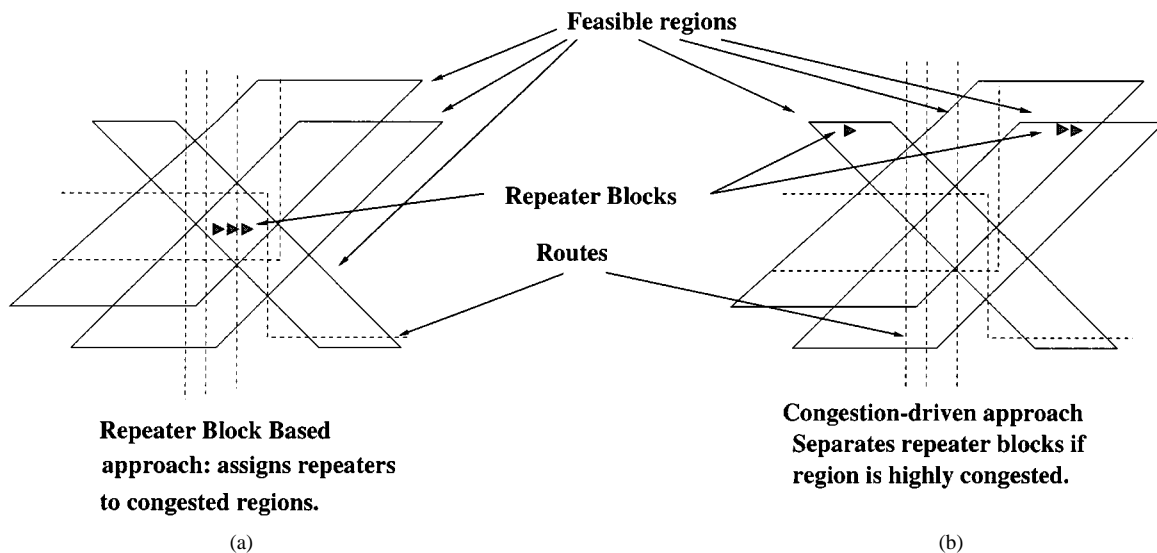


Fig. 1. Impact of routing congestion on repeater block planning.

II. PROBLEM FORMULATION

In this paper, we study the following *repeater block planning problem*: given an initial floorplan and timing constraints on each net, find the number, locations, and sizes of the repeater blocks to be inserted in order to meet the timing constraints. As in [1], we consider the insertion of repeater blocks in the channel regions between circuit blocks in which repeater insertion is not allowed.

Although the primary objective of the *repeater block planner* is to meet the timing constraints for all nets, it is equally important to avoid routing “hot spots” and keep the number of repeater blocks and the increase in chip area within tolerable limits. The two objectives of reducing the total number of repeater blocks added and avoiding regions with high routing congestion are contradictory in nature; by clustering large number of repeaters into a repeater block, we create a highly congested routing region around that repeater block.

Fig. 1 compares a routability-driven repeater block planner with a repeater planner that tries to minimize repeater blocks. Feasible regions (FRs) of three repeaters from different nets are shown and the dashed lines represent the routes passing through the region. If minimizing repeater blocks is the sole objective, the repeater block would be chosen in the high congestion region as shown in Fig. 1(a). A routability-aware repeater planner, however, would move the repeater blocks away from the high congestion zone. As in Fig. 1(b), a routability-driven repeater block planner would strike a balance between the two contradictory criteria: routing congestion and repeater block count.

We measure the *overall cost* of a repeater block plan by a weighted composition of two cost functions: one cost function guides the solution toward minimum number of repeater blocks; the other one minimizes the routing congestion of the solution. We shall present in detail the cost functions in Sections IV-A and IV-B. The weights of the two cost functions may be suitably adjusted to reflect the relative importance of each criterion. Our repeater block planner tries to obtain a repeater block plan that minimizes the overall cost of the solution.

III. FEASIBLE REGION COMPUTATION

In this section, we introduce the concept of IFR for repeater placement and obtain an expression for computing its width. We define the IFR for a repeater to be the region in which it can be placed such that the timing constraint of the net is satisfied, assuming that the other repeaters of that net are also located within their respective IFRs. The effectiveness of its use in repeater block planning is demonstrated in Section V by the significantly higher completion rates that we obtain as compared to [1].

A. Preliminaries

First, we present the definitions and expressions that will be used in stating the main result for IFR computation. Each driver/repeater is modeled as a switch-level *RC* circuit [16] and the Elmore delay formula [17] is used for delay computations. The notation for the physical parameters of the interconnect and repeater we use in this paper is as follows:

| | |
|-------|-----------------------------------|
| r | wire resistance per unit length; |
| c | wire capacitance per unit length; |
| T_b | intrinsic repeater delay; |
| C_b | repeater input capacitance; |
| R_b | repeater output resistance. |

Given a wire segment of length l with driver output resistance R and sink capacitance C , the Elmore delay of this segment is defined as

$$D(R, C, l) = \left(\frac{rC}{2}\right)l^2 + (Rc + rC)l + RC. \quad (1)$$

Using the above expression, the Elmore delay of a single-source single-sink net \mathbf{N} (two-pin net) of length L with n repeaters can be expressed as

$$\begin{aligned} D^{\mathbf{N}}(x_1, x_2, \dots, x_n, L) &= D(R_d, C_b, x_1) + D(R_b, C_s, L - x_n) \\ &\quad + \sum_{i=1}^{n-1} D(R_b, C_b, x_{i+1} - x_i) + nT_b \end{aligned}$$

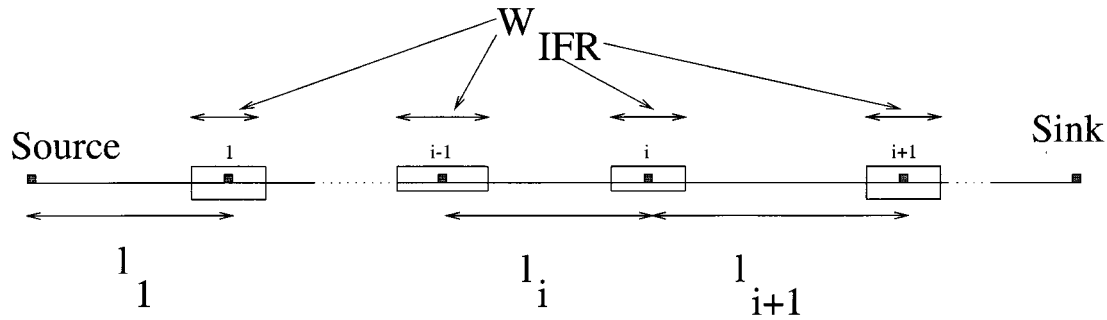


Fig. 2. IFRs.

where

- R_d driver resistance;
- C_s sink capacitance;
- x_i location of the i th repeater.

The optimal locations of the n repeaters for delay minimization of the net as shown in [18] are

$$x_i^* = (i-1)y_L^* + x_L^* \quad i \in \{1, 2, \dots, n\} \quad (2)$$

where

$$x_L^* = \frac{1}{n+1} \left(L + \frac{n(R_b - R_d)}{r} + \frac{(C_s - C_b)}{c} \right) \quad (3)$$

$$y_L^* = \frac{1}{n+1} \left(L - \frac{(R_b - R_d)}{r} + \frac{(C_s - C_b)}{c} \right). \quad (4)$$

We denote the optimal delay for the net \mathbf{N} of length L with n repeaters by

$$D_{\text{opt}}^{\mathbf{N}}(n, L) = D^{\mathbf{N}}(x_1^*, x_2^*, \dots, x_n^*, L).$$

B. FR for Repeater Insertion

In [1], the FR for repeater insertion was defined as the region in which a repeater could be placed, assuming that all the remaining repeaters were optimally placed with respect to it, in order to satisfy the target delay constraint, denoted by $D_{\text{tgt}}^{\mathbf{N}}$. We denote the width of the FR for a given repeater by W_{FR} . An analytical expression for W_{FR} was given in [1]. The following theorem presents an alternative but equivalent analytical expression, of which the proof is presented in the Appendix.

Theorem 1: For $D_{\text{tgt}}^{\mathbf{N}} \geq D_{\text{opt}}^{\mathbf{N}}(n, L)$, the width of the FR for the i th repeater ($i \leq n$) of the net \mathbf{N} is

$$W_{\text{FR}} = 2 \cdot \sqrt{\frac{2(D_{\text{tgt}}^{\mathbf{N}} - D_{\text{opt}}^{\mathbf{N}}(n, L))(n-i+1)(i)}{rc(n+1)}}. \quad \square$$

However, such a definition of FRs has the disadvantage that if we assign a repeater to a location that is on or near the boundary of its FR, the FRs of the other repeaters are almost entirely eliminated.

C. IFR

As opposed to the definition of FR, the IFR of a repeater is the region where it can be placed while meeting the timing specifications of the net, assuming that the other repeaters are placed within their respective IFRs.

Formally, we define the IFR for the i th repeater of a net \mathbf{N} as

$$\text{IFR}_i = (x_i^* - W_{\text{IFR}}/2, x_i^* + W_{\text{IFR}}/2) \cap (0, L)$$

such that $\forall (x_1, x_2, \dots, x_i, \dots, x_n) \in \text{IFR}_1 \times \text{IFR}_2 \times \dots \times \text{IFR}_n$, $D^{\mathbf{N}}(x_1, x_2, \dots, x_n, L) \leq D_{\text{tgt}}^{\mathbf{N}}$. Here, W_{IFR} and $D_{\text{tgt}}^{\mathbf{N}}$, respectively, denote the width of IFR_i and the target delay associated with the net.

Note that the final placement of a repeater in its IFR does not depend on the placement of the other repeaters so long as they are placed within their respective IFRs. To allocate an equal degree of freedom to each repeater in the net, we choose the IFR intervals to be of equal width (see Fig. 2). We have the following theorem for the width of the IFR of a repeater.

Theorem 2: For $D_{\text{tgt}}^{\mathbf{N}} \geq D_{\text{opt}}^{\mathbf{N}}(n, L)$, the width of the IFR for the i th repeater ($i \leq n$) of the net \mathbf{N} is

$$W_{\text{IFR}} = 2 \cdot \sqrt{\frac{D_{\text{tgt}}^{\mathbf{N}} - D_{\text{opt}}^{\mathbf{N}}(n, L)}{rc(2n-1)}}. \quad \square$$

The proof is presented in the Appendix. It can be trivially shown that when $n = 1$, $W_{\text{FR}} = W_{\text{IFR}}$. Although $W_{\text{IFR}} < W_{\text{FR}}$ for $n > 1$, using IFRs for repeater planning results in a solution of higher quality because it facilitates a higher degree of fairness during global optimization of repeater block clustering.

Note that when the objective is to minimize the number of repeaters inserted in a net, the IFRs of repeaters belonging to the same net do not overlap each other. Otherwise, at least one repeater can be eliminated without violating the given timing constraint.

D. Two-Dimensional IFR

In the preceding discussions, we were limited to repeater insertion along a one-dimensional (1-D) line. Implicit in the discussions was the assumption that the route from source to sink is specified by some global router. For repeater block planning during floorplanning, however, no routing information is available. We assume that each net would be routed with a shortest path within the bounding box containing the two terminals. Therefore, we need to compute two-dimensional (2-D) regions in which the repeaters can be placed. The 2-D IFR of a repeater is defined as the union of the 1-D IFRs of that repeater on all monotonic Manhattan routes between source and sink. Therefore, 2-D IFRs, as in [1], are convex octilinear polygons with horizontal, vertical, and ± 1 -slope boundaries (see Fig. 3).

However, 2-D IFRs of repeaters belonging to the same net are not completely independent of each other. As the widths and locations of a 2-D IFR are valid only under the assump-

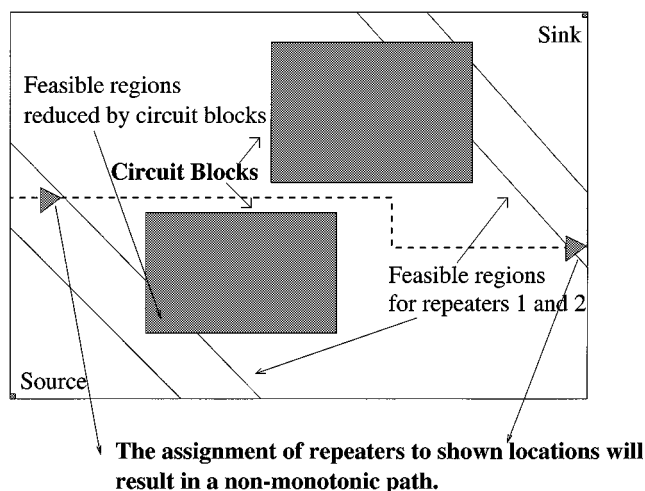


Fig. 3. Nonmonotonic repeater assignments.

tion that a monotonic Manhattan route exists between the source and the sink, the assignments of repeaters to locations within their respective 2-D IFRs must be made such that they constitute a monotone path from source to sink. In Fig. 3, for example, the repeater assignments, which form a nonmonotonic sequence from the source to the sink, violate the monotonicity constraint even though the repeaters are within their respective 2-D IFRs. Therefore, whenever the 2-D IFR of a repeater is modified, the 2-D IFRs for all other repeaters in the net may need to be updated.¹ In this paper, we develop an efficient scheme to perform the update. Details of the updating scheme are presented in Section IV-D. In the subsequent discussions, we shall deal with only 2-D IFRs, which we simply refer to as IFRs.

IV. REPEATER BLOCK PLANNING

In this section, we describe in detail our routability-driven repeater block planning algorithm. Given a floorplan, we assume that repeaters can only be inserted within the vertical and horizontal channels [19] defined by circuit blocks as in [1]. We also assume that no repeaters can be inserted into circuit blocks.

For the purpose of routing congestion computation (to be described in Section IV-A), we divide the entire chip area into a set of rectangular routing tiles as shown in Fig. 4. The routing tiles correspond to higher metal layers reserved for the routing of global nets. We also divide the channel between circuit blocks into a set of repeater-block tiles, which are of a finer resolution than routing tiles. As in [1], these repeater-block tiles represent locations where the repeaters may be placed. Fig. 4 shows a subset of repeater-block tiles. In essence, we construct a *two-level* tile structure: one for the purpose of estimating routing-congestion and the other for defining candidate repeater block (CRB) locations.

For each repeater b to be inserted, we find \mathbf{S}_b , the set of CRBs in which it can be placed. As shown in Fig. 4, each repeater has

¹Even though 2-D IFRs are not truly independent, they do not exhibit the same problem as FRs when a repeater is assigned to the boundary of its FR, i.e., the FRs of remaining repeaters are almost entirely eliminated. Note that for 2-D FRs, repeaters of a net must also constitute a monotone path from source to sink because a 2-D FR is similarly obtained as the union of the 1-D FRs of that repeater on all monotonic Manhattan routes between source and sink.

several CRBs to which it may be assigned. The objective of the repeater block planner is to assign each repeater to a single CRB. To solve this problem, we take the approach of first generating the candidate set for each repeater and then using a routing-congestion driven *iterative deletion* [20] algorithm to obtain an assignment for each repeater.

The iterative deletion procedure operates on a *bipartite graph* \mathcal{G} that represents the set of all possible repeater assignments. Let \mathbf{B} be the set of all repeaters that need to be inserted. The edge set of \mathcal{G} is defined as $E(\mathcal{G}) = \{(b, c) : b \in \mathbf{B}, c \in \mathbf{S}_b\}$. The iterative deletion algorithm starts with a redundant solution space that contains all possible assignments of repeaters to CRBs. One at a time, the algorithm deletes an incompatible repeater assignment, i.e., an assignment that results in high routing congestion or too many repeater blocks. Equivalently, the iterative deletion algorithm removes an edge from the bipartite graph one at a time. When the iterative deletion algorithm terminates, we obtain assignments of repeaters to CRBs that are compatible to each other in terms of routability and repeater block count.

Fig. 5 gives the overall flow of our repeater block planning algorithm. Steps 1 through 4 are data preparation stages. Step 1 constructs a two-level tile structure as shown in Fig. 4. In Steps 2 and 3, we compute the minimum number of repeaters required for each two-pin net to meet its target delay [1], [18] and assign W_{IFR} of each repeater based on the available net slack ($D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L)$) as defined in Theorem 2. Then, we generate the candidate set \mathbf{S}_b for each repeater b by intersecting the IFR of the repeater with the set of repeater-block tiles within the bounding box of the net. If the IFR for a repeater does not have any intersections with the set of repeater-block tiles, then we consider possible repeater locations along the boundaries of circuit blocks. For such nets, we allow repeaters to be placed along the circuit block boundaries in order to meet their timing constraints. Step 4 constructs the bipartite graph \mathcal{G} .

Steps 5–10 perform the iterative deletion operations. The iterative deletion procedure assigns each repeater to one of its CRBs. To accomplish the task of deleting incompatible repeater assignments, the edges in \mathcal{G} have dynamic weights. The weight of edge (b, c) reflects the “cost” of assigning repeater b to CRB c , assuming that the other repeaters can go into any of their CRBs with equal probability. An edge of a higher cost implies that the repeater block assignment is likely to be incompatible with the rest and the iterative deletion operation in Steps 6–10 (to be described in Section IV-C) seeks to remove the highest cost edge or the most incompatible repeater block assignment. As edges are iteratively deleted from graph \mathcal{G} , the routability of the floorplan is modified and so is the expected repeater block count. These changes are reflected onto the edge costs of the graph at every iteration, thus making the edge weights dynamic. We shall present the routing congestion model and the associated cost function in Section IV-A and the edge weight, which is a composite cost function of the routing congestion cost and repeater block cost, in Section IV-B.

A. Congestion Model

The congestion model employed is essentially a 2-D rectangular grid based probabilistic map assuming *two-bend* routing

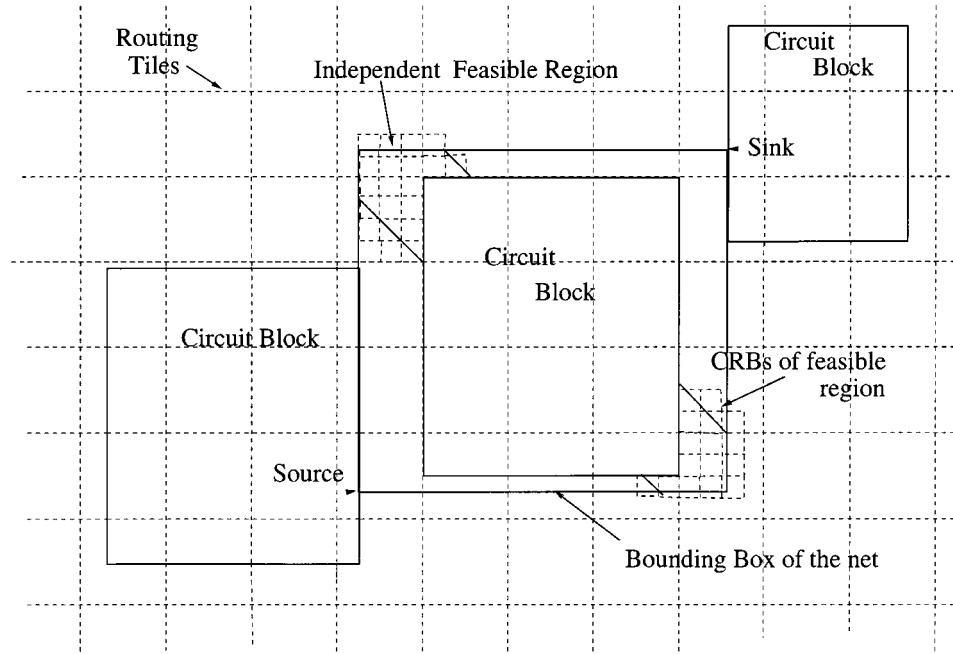


Fig. 4. Creation of routing tiles and CRBs.

1. Build a two-level tile structure on a given floorplan;
2. Compute IFR for each repeater $b \in \mathbf{B}$;
3. Obtain CRB set \mathbf{S}_b for each repeater $b \in \mathbf{B}$;
4. Generate the bipartite graph \mathcal{G} ;
5. **while** there exists a repeater to be assigned **do**
6. Delete the highest cost edge of \mathcal{G} ;
7. Update monotonicity;
8. Update congestion matrix;
9. Update edge costs;
10. Assign repeater to a CRB if required;

Fig. 5. Repeater block planning algorithm.

for each segment. This is similar to that developed in [14]. The *congestion* of a routing tile $\text{tile}(i, j)$ is defined as the following.

$C_h(i, j)$ Expected number of horizontal routes passing through $\text{tile}(i, j)$;

$C_v(i, j)$ Expected number of vertical routes passing through $\text{tile}(i, j)$.

To derive the *congestion numbers* for the routing grid, we first compute the expected number of routes passing through every routing tile for a wire segment with a fixed source and a fixed sink.

Without loss of generality, we consider the source to be located in $\text{tile}(0, 0)$ and the sink to be located in $\text{tile}(m, n)$ and assume that a *bend* consumes both horizontal and vertical routing resources. It is easy to see that there are a total of $(m + n)$ possible *two-bend* routes from source to sink. Assuming that all these routes are equally likely, we obtain the *probability matrix* shown in Fig. 6. $\delta C_h(i, j)$ is defined as the probability of a horizontal route passing through $\text{tile}(i, j)$. Similarly, $\delta C_v(i, j)$ is defined for vertical routes.

We define the segment of a net between two consecutive repeaters between the source and the first repeater or between the

| Tile | $\delta C_h(i, j)$ | $\delta C_v(i, j)$ |
|----------------------------------|---------------------|---------------------|
| $0 < i < m, 0 < j < n$ | $\frac{1}{m+n}$ | $\frac{1}{m+n}$ |
| $0 < i \leq m, j = 0$ | $\frac{m-i+1}{m+n}$ | $\frac{1}{m+n}$ |
| $0 \leq i < m, j = n$ | $\frac{i+1}{m+n}$ | $\frac{1}{m+n}$ |
| $i = 0, 0 < j \leq n$ | $\frac{1}{m+n}$ | $\frac{n-j+1}{m+n}$ |
| $i = m, 0 \leq j < n$ | $\frac{1}{m+n}$ | $\frac{j+1}{m+n}$ |
| $i = 0, j = 0$ or $i = m, j = n$ | $\frac{m}{m+n}$ | $\frac{n}{m+n}$ |

Fig. 6. Probability matrix.

last repeater and the sink as a *subnet*. In the problem at hand, the source and the sink of a subnet may have several candidate locations if they are repeaters. We compute the contribution of each subnet to the congestion matrices \mathbf{C}_h and \mathbf{C}_v by fixing the source and sink of a subnet segment to the *centroids* of their CRBs. Thus, given a set of two-pin nets and candidate locations for the repeaters we can compute the congestion matrices \mathbf{C}_h and \mathbf{C}_v as follows:

$$C_h(i, j) = \sum_{\forall \text{subnets}} \delta C_h(i, j)$$

$$C_v(i, j) = \sum_{\forall \text{subnets}} \delta C_v(i, j).$$

As is done in a number of other routers, we assign a congestion cost to each routing tile. A number of ways for modeling the congestion cost have been proposed in literature [21]–[23]. For the purpose of this paper, we use a monotonic piecewise cost function of the type proposed in [24] (see Fig. 7).

B. Dynamic Edge Weights

The cost of assigning a repeater to a CRB (or the edge cost) is a weighted composite function comprising of the congestion cost and the repeater block cost.

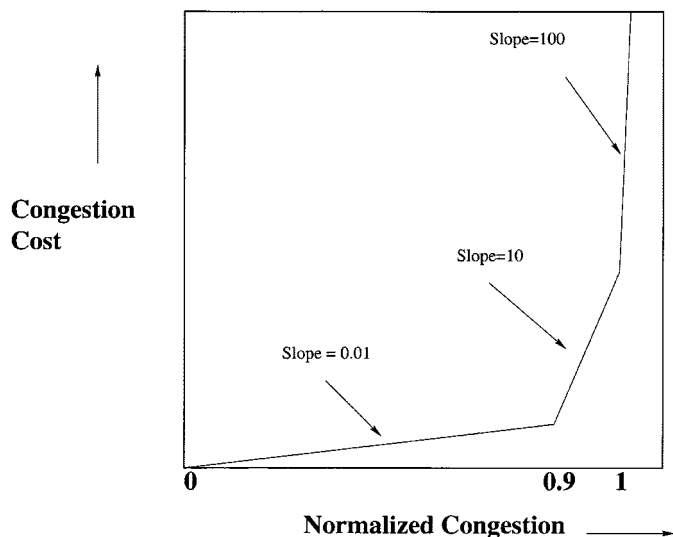


Fig. 7. Piecewise-linear congestion cost function.

- 1) *Congestion Cost*: The congestion cost of an edge is defined as the maximum congestion cost among all routing tiles in the *one-bend* routing path for the subnet that has the repeater as its source and the subnet that has the repeater as its sink. The respective sink and source of the subnets are assumed to be located at the centroids of their CRBs. One-bend routing instead of two-bend routing is used for congestion cost calculation because of its efficiency. The congestion metric and the associated cost functions used have been described in Section IV-A. We denote the congestion cost of an edge $e = (b, c) \in E(\mathcal{G})$ as $CC(e)$.
- 2) *Repeater Block Cost*: The repeater block cost function is used to guide the iterative deletion procedure to converge to a solution with minimum number of repeater blocks. Let c be a CRB. Let B_c be the number of IFRs intersecting on this CRB. Also, define B_{\max} to be the maximum number of repeaters that can be inserted into a CRB. The repeater block cost of the CRB c , $BB(c)$ is defined as follows. If the number of repeaters assigned to the channel tile is less than B_{\max} , then we define $BB(c)$ to be $1/\min(B_c, B_{\max})$. Otherwise, we define $BB(c) = \infty$.

The composite cost function is a weighted product of the two costs. The cost of an edge $e = (b, c) \in E(\mathcal{G})$ is defined as

$$COST(e) = (CC(e))^{p_1} \times (BB(c))^{p_2}$$

where p_1 and p_2 are positive parameters such that $p_1 + p_2 = 1$. Changing the values of p_1 and p_2 allows us to perform a tradeoff between the two contradictory criteria involved.

C. Iterative Deletion

The iterative improvement method modifies the solution space repeatedly by changing a small portion at each step. Our procedure begins with a redundant set of possible locations for each repeater and removes a single highest cost redundant assignment at each step while attempting to minimize the *cost* of the solution. It proceeds by deleting the highest cost edge $e = (b, c)$ of \mathcal{G} . Also, let \mathbf{N} denote the net to which repeater b

belongs. We update the bipartite graph and its associated edge costs as follows.

- 1) *Monotonicity Update*: As mentioned in Section III-D, the formula for the IFR of each repeater has been derived assuming that a monotonic route between the source and the sink through the repeater locations is generated. To prevent nonmonotonic repeater location assignments, we update the candidate sets for repeaters in the net \mathbf{N} to ensure the existence of a monotonic path. A CRB in \mathbf{N} lies on a monotonic path if there exists a sequence of CRBs, one in each CRB set of the remaining repeaters, that forms a monotonic path from source to sink. In Fig. 3, for example, the repeater assignments shown form a nonmonotonic sequence from the source to the sink. CRBs that do not lie on a monotonic path are removed. In Section IV-D, we present an efficient technique to remove nonmonotone CRBs.
- 2) *Congestion Update*: Deletion of the highest cost edge and the corresponding monotonicity update change the solution set represented by the bipartite graph \mathcal{G} . After monotonicity update, the CRB sets of repeaters on a net change. Once all the changes to the CRB set of a repeater have been determined, the new centroid of the CRB set can be computed in linear time with respect to the number of deleted CRBs. The congestion update procedure reflects the impact of these changes onto the congestion matrices \mathbf{C}_h and \mathbf{C}_v . As the centroids of the CRB sets of repeaters on the net before and after the monotonicity update are known, the congestion matrix can be updated in a very efficient manner (linear with respect to the number of routing tiles through which the affected subnets pass through). Note that the update is necessary only when the new and old centroids of a subnet lie in different routing tiles. It is important to note that the centroids usually change their routing tile locations only after several iterations of monotonicity updates. We also observe that the congestion matrix does not change significantly with each congestion update.
- 3) *Edge Cost Update*: Due to monotonicity and congestion updates, the cost of assigning a repeater to a particular tile changes. This procedure updates the cost of the edges in the graph \mathcal{G} . The updates of the repeater block costs can be done by simply recomputing the function $BB(c)$ for all CRBs that have been deleted or to which repeater assignments have been made. The update of $BB(c)$ for all edges incident on c is linear with respect to the degree of c in \mathcal{G} . However, it is very costly to reflect the changes of the congestion cost in the edge cost with each congestion update. As the congestion matrix does not change significantly with each occurrence of congestion update, we perform a lazy update of the routing cost for *all* the edges after every K occurrences of congestion updates. In our implementation, $K = 20$ provides a satisfactory compromise between efficiency and solution quality.

The assignment of a repeater to a repeater block is accomplished when the candidate set for the repeater consists of only one repeater block.

D. Efficient Technique for Monotonicity Update

As stated in Section III-D, it is important that the eventual repeater assignments of a net constitute a monotonic path from source to sink. To ensure this, the candidate sets of other repeaters belonging to the same net need to be properly updated upon the deletion of a CRB from the candidate set of a repeater of that net. This procedure is referred to as *monotonicity update*.

In the rest of this section, we shall consider the case where the net \mathbf{N} , whose source is located at (X_s, Y_s) and destination at (X_d, Y_d) , has $X_s \leq X_d$ and $Y_s \leq Y_d$.² The other three cases— $X_s \leq X_d$ and $Y_s \geq Y_d$, $X_s \geq X_d$ and $Y_s \leq Y_d$, and $X_s \geq X_d$ and $Y_s \geq Y_d$ —can be handled in a similar fashion. Let (X_i, Y_i) be the location of the CRB to which the i th repeater b_i is assigned. The *monotonicity update* procedure must ensure that the following property is preserved by the eventual assignments of n repeaters.

1) *Monotonicity Property (MP)*: $\forall i, j, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j$

$$X_i \leq X_j, \quad \forall i < j \quad \text{Prop. (1),}$$

$$Y_i \leq Y_j, \quad \forall i < j \quad \text{Prop. (2).} \quad \square$$

In order to preserve the monotonicity property, we make use of the *dominance relation* of CRBs. Consider two CRBs c and c' at locations (X_c, Y_c) and $(X_{c'}, Y_{c'})$, respectively. We say that c' *dominates* c if and only if $X_c \leq X_{c'}$ and $Y_c \leq Y_{c'}$. If neither c nor c' dominates the other, they are *mutually independent*. Given the set of CRBs \mathbf{S}_b of repeater b , we define an independent dominated set (IDS) IDS_b to be a subset of \mathbf{S}_b such that: 1) all CRBs in IDS_b are mutually independent; 2) other than self-dominance, CRBs from IDS_b do not dominate CRBs from \mathbf{S}_b ; and 3) every CRB from \mathbf{S}_b dominates (self-dominance included) at least one CRB from IDS_b . Fig. 8 shows the IDS of a repeater.

We arrange the CRBs in an IDS by their X -coordinates in increasing order. From each CRB in the IDS, we extend a vertical line segment northward either infinitely if the CRB is the first in the set or until it reaches the Y -coordinate of the previous CRB otherwise. We also extend from each CRB in the IDS a horizontal line segment eastward either infinitely if the CRB is the last in the set or until it reaches the X -coordinate of the next CRB otherwise. These line segments define a monotone contour of the IDS, which is dominated by CRBs in the northeast region of the contour. We refer to the monotone contour as the independent dominated contour (IDC) (see Fig. 8).

Consider two IDSs IDS_{b_i} and IDS_{b_j} , respectively, for repeaters b_i and b_j , $i < j$. We say that IDS_{b_j} dominates IDS_{b_i} or $\text{IDS}_{b_i} \leq \text{IDS}_{b_j}$ if and only if IDC_{b_j} is completely in the northeast region of IDC_{b_i} . When IDS_{b_j} dominates IDS_{b_i} , every CRB in IDS_{b_j} dominates at least one CRB in IDS_{b_i} . In other words, there exist some repeater assignments that constitute a monotone path. In order for the eventual repeater assignments to preserve the monotone property, we maintain the following *relaxed monotone property*.

²We use the notation (X, Y) for coordinates to avoid any confusion with the notation x and y used for denoting optimal repeater placement in Section III.

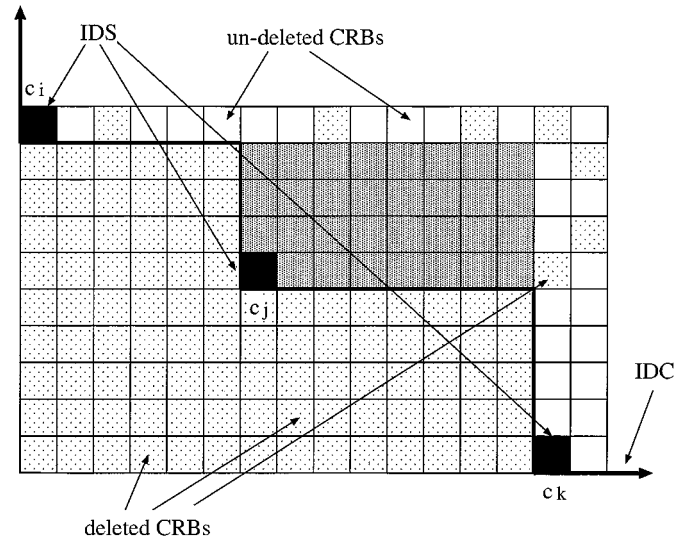


Fig. 8. IDS and its IDC. White squares corresponds to undeleted CRBs. Lightly shaded squares correspond to deleted CRBs. Black squares correspond to CRBs in the IDS. IDC is depicted by thick lines. Darker shaded squares correspond to the region in which one should search for new members of the IDS should c_j be deleted.

Relaxed Monotonicity Property (RMP): $\forall i, j, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j$

$$\text{IDS}_{b_i} \leq \text{IDS}_{b_j}, \quad \forall i < j \quad \text{Prop. (3).} \quad \square$$

When a CRB c of repeater b_i is considered for deletion, we have to update IDS_{b_i} if c happens to be in IDS_{b_i} . As a result, we have to update the IDSs of subsequent repeaters b_j , $i < j$ if Prop. (3) is violated. If the sequence of updates results in an empty \mathbf{S}_{b_k} for some b_k , then c should not be deleted in the first place. Instead, b_i should be assigned to c . Based on the above observation, a monotonicity update procedure that preserves the relaxed monotonicity property has been developed. Fig. 9 gives the details of the update procedure. We assume that $e = (b_i, c)$ is the edge to be deleted, where b_i is the i th repeater of net \mathbf{N} , and c is a CRB in \mathbf{S}_{b_i} . To be more general, the monotonicity update procedure takes in a subset of edges (adjacent to b_i) that are to be deleted. We denote the subset of edges by \hat{E}_{b_i} .

The Boolean variable `Violation_Flag` monitors whether an IFR has become empty because of the monotonicity update. When that happens, we undelete all CRBs that has been deleted and, making use of the changes stored in the variable `IDS/IDC_Changes`, revert to the original IDS and IDC for each IFR. It also signals to the iterative deletion procedure in Fig. 5 that b_i should be assigned to c , where $\hat{E}_{b_i} = \{e = (b_i, c)\}$ is the input edge set.

While the other steps are fairly straightforward, Steps 6 and 7 require further explanation. The procedure `IDS/IDC_Updates` updates the IDS_{b_i} and accordingly IDC_{b_i} . Consider deleting a CRB from \mathbf{S}_{b_i} ; we do not have to update IDS_{b_i} if that CRB is not originally in IDS_{b_i} . Now, we consider the case when we delete a CRB from IDS_{b_i} . In the IDS shown in Fig. 8, for example, we have three ordered CRBs: c_i , c_j , and c_k . When we delete c_j from the IDS, we search for new members of the IDS from within the darker shaded region

```

1. for each  $c \in \hat{E}_{b_i}$  do
2.   mark  $c$  deleted;
3. if  $S_{b_i} = \emptyset$  then
4.   Violation_Flag  $\leftarrow$  TRUE;
5. else /* Check if the next IFR becomes empty */
6.   IDS/IDC_Changes  $\leftarrow$  IDS/IDC_Updates( $\hat{E}_{b_i}, S_{b_i}, S_{b_{i+1}}$ );
7.    $\hat{E}_{b_{i+1}} \leftarrow \{c' \in S_{b_{i+1}} \text{ that violates RMP}\}$ ;
8.   if  $\hat{E}_{b_{i+1}} = \emptyset$  then
9.     Violation_Flag  $\leftarrow$  FALSE;
10.  else
11.    Violation_Flag  $\leftarrow$  Monotonicity Update Procedure( $\hat{E}_{b_{i+1}}$ );
12.  if Violation_Flag = TRUE then
13.    for each  $c \in \hat{E}_{b_i}$  do
14.      mark  $c$  undeleted;
15.    un-do IDS/IDC_Changes;
16.  return Violation_Flag;

```

Fig. 9. Monotonicity update procedure.

defined by the rectangles bounded by X_{c_j} , $X_{c_k} - 1$, Y_{c_j} , and $Y_{c_i} - 1$, assuming that every CRB occupies a unit square.

The new IDS members can be computed by sweeping either rowwise from Y_{c_j} to $Y_{c_i} - 1$ or columnwise from X_{c_j} to $X_{c_k} - 1$. Let c_r be the first CRB that we encounter when we sweep rowwise from Y_{c_j} to $Y_{c_i} - 1$ such that $X_{c_r} < X_{c_k}$. Then, any remaining new IDS members lie in the rectangular region defined by X_{c_j} , $X_{c_r} - 1$, $Y_{c_r} + 1$, and $Y_{c_i} - 1$. When we sweep columnwise, we search for the first CRB c_c such that $Y_{c_c} < Y_{c_i}$. Then, the next new IDS member, if any, lies in the rectangular region bounded by $X_{c_c} + 1$, $X_{c_k} - 1$, Y_{c_i} , and $Y_{c_c} - 1$.

Although the number of columns or rows that the algorithm has to sweep varies, we are able to amortize the operation cost such that the runtime complexity of a CRB deletion operation is $O(1)$. The cost of marking a CRB as deleted, be it a IDS member or not, is charged to the CRB itself. In the following analysis, we assume that the algorithm performs a rowwise sweep after deleting a IDS member. If sweeping a row produces a new IDS member, we charge the operation cost to that CRB. An IDS member, once created, will not be involved in any sweeping operation again. The next time it is charged with an operation cost again is when it is marked deleted. If the sweep produces no new IDS member, then we charge the cost to a previously deleted CRB, which is located at the eastmost boundary of the rectangular region. For the example used in the preceding two paragraphs, we charge the cost of sweeping rows between Y_{c_j} to $Y_{c_r} - 1$ to CRBs with coordinates $(X_{c_k} - 1, Y_{c_j}), \dots, (X_{c_k} - 1, Y_{c_r} - 1)$. Such CRBs will not be charged with another sweep as they now lie outside the northeast region of the IDC. Therefore, every CRB is charged at most twice, i.e., $O(1)$.

Propagating the change in the contour from IDC_{b_i} to $IDC_{b_{i+1}}$ may turn out to be slightly more costly. Using the example in Fig. 8, let c_r be the CRB that is ordered before c_k in the new IDS after c_j is deleted. If $(X_{c_k} - 1, Y_{c_r} - 1)$ is in the northeast region of $IDC_{b_{i+1}}$, then we have to update $IDC_{b_{i+1}}$. As $IDS_{b_{i+1}}$ is ordered, we can perform a binary search in $O(\log |IDS_{b_{i+1}}|)$ to find the two adjacent CRBs in $IDS_{b_{i+1}}$ whose Y -coordinates sandwich $Y_{c_r} - 1$. Note that $|IDS_{b_{i+1}}|$ is no greater than the

minimum of the numbers of the columns and rows in $IDS_{b_{i+1}}$. If $(X_{c_k} - 1, Y_{c_r} - 1)$ does not dominate the two adjacent CRBs, then no deletion of CRBs in $S_{b_{i+1}}$ is required. We can achieve a speedup by taking advantage of the geometrical properties of IFRs. As mentioned in III-D, all IFRs are octilinear in shape. Therefore, we can define for $S_{b_{i+1}}$ a -1 -slope line $Y = -X + M$ such that for each $c \in S_{b_{i+1}}$, $X_c + Y_c \geq M$. Only when $X_{c_k} + Y_{c_r} - 2 \geq M$ do we have to perform a binary search. In general, we have to verify the new contour between c_i and c_k , which does not invade the northeast region of $IDC_{b_{i+1}}$ after deleting c_j .

The algorithm outlined in Fig. 9 illustrates how the deletion of a CRB of b_i affects the IFRs of repeaters b_j , $j > i$ in net \mathbf{N} . Such a deletion also affects the IFRs of repeaters b_j , $j < i$ in net \mathbf{N} . Similar to the algorithm in Fig. 9, we develop an algorithm to perform monotonicity updates of IFRs for b_j , $j < i$ by defining the *independent dominating set* of S_b as follows: 1) all CRBs in the independent dominating set are mutually independent; 2) other than self-dominance, CRBs from S_b do not dominate CRBs from the independent dominating set; and 3) every CRB from S_b is dominated (self-dominance included) by at least one CRB from the independent dominating set. The independent dominating set also defines an independent dominating contour, which dominates all CRBs in the southwest region of the contour. As the algorithm to update the independent dominating contours of preceding repeaters is very similar to the algorithm in Fig. 9, we do not present its details here.

V. EXPERIMENTAL RESULTS

We have implemented our repeater block planning algorithm using C on a SUN UltraSPARC II machine. In this section, we present the details of our experimental set up and the results obtained. The interconnect and repeater parameters and the Elmore delay model have been described in Section III-A. The values (see Table I) used for these parameters are based on a $0.18 \mu\text{m}$ technology in [25].

We report the results of our repeater block planner for six MCNC [26] benchmark circuits. The relevant details of these benchmarks are shown in Table II. In this paper, we focus on solving the problem of repeater block planning for two-pin (single source, single sink) nets. As the benchmark files do not provide information on signal direction, we choose the first pin to be the source and all the others to be sinks and decompose a multiple terminal net into a set of two-pin nets. This may not be good especially for path based timing optimizations, but it provides a sufficiently good model of the input to a repeater block planner. We ignore all single pin, power, and ground interconnects. The initial floorplans of the MCNC benchmark circuits used for this paper were obtained from [27] and are the same as those used in [1]. These floorplans were generated by running simulated tempering using an improved Monte-Carlo technique [28].

Since the MCNC benchmarks do not come with any timing information, we assign target delays to the two-pin nets as follows. We ignore all two-pin nets whose lengths are smaller than the critical length L_{\min} [18], above which repeater insertion

TABLE I
PARAMETER VALUES FOR INTERCONNECT, REPEATER, DRIVER, AND SINK

| | Description | Value |
|-----------|---|-------|
| r | wire resistance per unit length ($\Omega\mu\text{m}$) | 0.075 |
| c | wire capacitance per unit length (fF/ μm) | 0.118 |
| T_b | intrinsic repeater delay (ps) | 36.4 |
| C_s/C_b | sink/repeater input capacitance (fF) | 23.4 |
| R_b/R_d | driver/repeater output resistance (Ω) | 180 |

TABLE II
DETAILS OF MCNC BENCHMARKS

| Circuit | Modules | Nets | 2-Pin Nets |
|---------|---------|------|------------|
| apte | 9 | 97 | 172 |
| xerox | 10 | 203 | 455 |
| hp | 11 | 83 | 226 |
| ami33 | 33 | 123 | 363 |
| ami49 | 49 | 408 | 545 |
| playout | 62 | 2506 | 2150 |

can be used for delay reduction. For each net, we then compute the optimal delay T_{opt} obtainable by repeater insertion [18] and then randomly assign a target delay between 1.05 and 1.20 $\times T_{\text{opt}}$ as in [1]. As we generate these timing constraints on our own, a direct comparison between our method and [1] may not be fair even though we do use their results for reference in the following discussion.

In Table III, we report the following results from our repeater block planning algorithm: 1) ratio of number of nets for which the delay constraint is met to the number of nets for which the delay constraint is not satisfied $MET/NOTMET$; 2) the total number of repeaters inserted to meet timing constraints, N_{REP} ; 3) the maximum tile congestion C_{MAX} ; 4) the chip area increase δA expressed as a percentage of the original chip area; and 5) CPU time required, TCPU. We also include the results for these benchmark circuits reported in [1].

Compared to [1], MET is significantly higher. The success rates of meeting the timing constraints (or completion rates) range from 71% (for apte) to 95% (for playout). The average completion rate is 85%. On the other hand, the average completion rate for the same examples in [1] is 67.5% and the completion rates range from 58% (for Xerox and HP) to 84% (for ami33). Our completion rates are higher for all benchmark circuits in Table II.

In Table III, we report an identical N_{REP} under three different combinations of p_1 and p_2 for each benchmark circuit even though the completion rates are different. The completion rates are different because some nets cannot meet the timing constraints after we expand the floorplan to accommodate the repeater blocks along the boundaries of circuit blocks (see Section IV). In fact, all entries in N_{REP} , C_{MAX} , and δA include nets that do not meet the timing constraints after floorplan expansions.

Table III also shows that we generally require a smaller number of repeaters to achieve a higher completion rates compared to [1]. In [1], all channels between blocks are expanded

TABLE III
COMPARISON OF REPEATER BLOCK PLANNING SOLUTIONS

| Circuit | $\frac{MET}{NOTMET}$ | N_{REP} | C_{MAX} | δA | TCPU (s) |
|--------------------|----------------------|-----------|-----------|------------|----------|
| apte | | | | | |
| $p_1 = 1, p_2 = 0$ | 122 / 50 | 176 | 10.77 | 1.44 | 1.7 |
| $p_1 = 0, p_2 = 1$ | 120 / 52 | 176 | 11.09 | 1.44 | 1.5 |
| $p_1 = p_2 = 0.5$ | 121 / 51 | 176 | 11.00 | 1.44 | 1.2 |
| [1] | 102 / 70 | 185 | - | 0.69 | 0.23 |
| xerox | | | | | |
| $p_1 = 1, p_2 = 0$ | 368 / 87 | 354 | 22.02 | 1.24 | 8.0 |
| $p_1 = 0, p_2 = 1$ | 361 / 94 | 354 | 34.98 | 1.24 | 7.0 |
| $p_1 = p_2 = 0.5$ | 361 / 94 | 354 | 19.74 | 1.24 | 7.5 |
| [1] | 260 / 195 | 399 | - | 1.38 | 0.53 |
| hp | | | | | |
| $p_1 = 1, p_2 = 0$ | 185 / 41 | 258 | 30.68 | 1.03 | 2.1 |
| $p_1 = 0, p_2 = 1$ | 175 / 51 | 258 | 32.56 | 1.03 | 2.3 |
| $p_1 = p_2 = 0.5$ | 169 / 57 | 258 | 31.36 | 1.03 | 2.2 |
| [1] | 131 / 95 | 280 | - | 1.24 | 0.48 |
| ami33 | | | | | |
| $p_1 = 1, p_2 = 0$ | 326 / 37 | 243 | 54.27 | 1.44 | 4.2 |
| $p_1 = 0, p_2 = 1$ | 324 / 39 | 243 | 61.24 | 1.44 | 4.7 |
| $p_1 = p_2 = 0.5$ | 324 / 39 | 243 | 51.71 | 1.44 | 3.9 |
| [1] | 305 / 58 | 667 | - | 1.36 | 1.63 |
| ami49 | | | | | |
| $p_1 = 1, p_2 = 0$ | 497 / 48 | 287 | 15.66 | 1.04 | 8.5 |
| $p_1 = 0, p_2 = 1$ | 496 / 49 | 287 | 19.99 | 1.04 | 7.2 |
| $p_1 = p_2 = 0.5$ | 496 / 49 | 287 | 15.74 | 1.04 | 7.0 |
| [1] | 412 / 133 | 946 | - | 0.78 | 3.25 |
| playout | | | | | |
| $p_1 = 1, p_2 = 0$ | 2053 / 97 | 1090 | 274.44 | 1.32 | 63.0 |
| $p_1 = 0, p_2 = 1$ | 2017 / 133 | 1090 | 281.73 | 1.32 | 62.1 |
| $p_1 = p_2 = 0.5$ | 2020 / 130 | 1090 | 253.36 | 1.32 | 66.2 |
| [1] | 1533 / 617 | 4263 | - | 0.84 | 13.98 |

at the beginning of the planning step [27]. As a result, net lengths become longer and more repeaters are required. In our algorithm, we expand the channels only when required. Therefore, our net lengths are "shorter" in general and we require a smaller number of repeaters to meet the timing constraints.

The results in Table III also show that our algorithm is able to reduce routing congestion. With $p_1 = 1$ and $p_2 = 0$, the objective function is oriented toward the reduction of congestion cost, whereas setting $p_1 = 0$ and $p_2 = 1$ causes the repeater block count to be minimized. $p_1 = p_2 = 0.5$ represents a solution with a tradeoff between the two contradictory cost functions. Table III shows that routing congestion levels can be lowered when they are accounted for during repeater block planning. In Xerox, for example, C_{MAX} for $p_1 = 0$ and $p_2 = 1$ (i.e., the objective of minimizing the number of repeater blocks) is about 50% higher than the C_{MAX} for $p_1 = 1$ and $p_2 = 0$ (i.e., the objective of minimizing the routing congestion). Clearly, if the design is routing-limited, the actual completion rates for the former case will be reduced accordingly. However, the average increase in chip area due to repeater insertion is 1.25%, which is higher than the 1.05% average increase reported in [1]. Moreover, the runtimes of our algorithm are higher than those in [1].

VI. CONCLUSION

In this paper, we have presented a repeater block planning algorithm that uses the concept of IFRs to do repeater clustering. The assignments of repeaters to repeater blocks are routability-driven and, thus, we are able to avoid regions of high routing congestion. Experimental results show that our technique performs significantly better than existing repeater planning methods.

APPENDIX
PROOFS OF THEOREMS 1 AND 2

We consider a net \mathbf{N} of length L optimally inserted with n repeaters. We denote the optimal placement of the first repeater by x_L^* , the distance between two optimally placed repeaters by y_L^* , and the distance between the last repeater and the sink by z_L^* . The expressions for x_L^* and y_L^* are given by (3) and (4), respectively, and $z_L^* = L - (n-1)y_L^* - x_L^*$.

In the following lemma and theorems, we deal with repeaters displaced from their optimal locations. As a result, the distances between the driver and the first repeater, between repeaters, and between the last repeater and the sink change. We capture the resultant change in the delay by the following expression:

$$\begin{aligned} \Delta(R, C, w, l) &= D(R, C, l+w) - D(R, C, l) \\ &= \frac{rc}{2} w^2 + (rcd + Rc + rC)w \end{aligned} \quad (5)$$

where

| | |
|--------------|-----------------------|
| w | change in net length; |
| l | original net length; |
| R | driver resistance; |
| C | sink capacitance; |
| $D(R, C, l)$ | Elmore delay in (1). |

The coefficient of w in (5) has an interesting property summarized in the following lemma.

Lemma 1: The coefficient of w in (5) satisfies the following equalities:

$$\begin{aligned} rcx_L^* + R_d c + rC_b &= rcy_L^* + R_b c + rC_b \\ &= rcz_L^* + R_b c + rC_s. \end{aligned} \quad (6)$$

Proof: From (2)–(4), we can show that

$$\begin{aligned} &rcx_L^* + R_d c + rC_b \\ &= \frac{rc}{n+1} L + \frac{c \cdot n}{n+1} R_b + \frac{c}{n+1} R_d + \frac{r}{n+1} C_s + \frac{r \cdot n}{n+1} C_b \\ &= \frac{rc}{n+1} \left[L - \frac{(R_b - R_d)}{r} + \frac{(C_s - C_b)}{c} \right] + R_b c + rC_b \\ &= rcy_L^* + R_b c + rC_b \end{aligned}$$

and

$$\begin{aligned} &rcz_L^* + R_b c + rC_s \\ &= rc(L - (n-1)y_L^* - x_L^*) + R_b c + rC_s \\ &= \frac{rc}{n+1} L + \frac{c \cdot n}{n+1} R_b + \frac{c}{n+1} R_d + \frac{c}{n+1} C_s + \frac{c \cdot n}{n+1} C_b \\ &= rcy_L^* + R_b c + rC_b. \quad \square \end{aligned}$$

Theorem 1: For $D_{tgt}^{\mathbf{N}} \geq D_{opt}^{\mathbf{N}}(n, L)$, the width of the FR for the i th repeater ($i \leq n$) of the net \mathbf{N} is

$$W_{FR} = 2 \cdot \sqrt{\frac{2(D_{tgt}^{\mathbf{N}} - D_{opt}^{\mathbf{N}}(n, L))(n-i+1)(i)}{rc(n+1)}}.$$

Proof: Let $D_{opt}^{\mathbf{N}}(n, L)$ denote the optimal delay for a net \mathbf{N} of length L inserted with n repeaters. Also, let $D_{opt}^{\mathbf{N}}(n, L+W)$ denote the optimal delay for the same net (i.e., same R_d and C_s), but with a difference of W in its wirelength. First, we consider the difference in the optimal delays of the two configurations

$$\begin{aligned} &D_{opt}^{\mathbf{N}}(n, L+W) - D_{opt}^{\mathbf{N}}(n, L) \\ &= D(R_d, C_b, x_{L+W}^*) - D(R_d, C_b, x_L^*) \\ &\quad + (n-1)[D(R_b, C_b, y_{L+W}^*) - D(R_b, C_b, y_L^*)] \\ &\quad + D(R_b, C_s, z_{L+W}^*) - D(R_b, C_s, z_L^*). \end{aligned}$$

From (2)–(4)

$$x_{L+W}^* - x_L^* = y_{L+W}^* - y_L^* = z_{L+W}^* - z_L^* = \frac{W}{n+1}.$$

Therefore, we can rewrite the difference between the optimal delays as

$$\begin{aligned} &D_{opt}^{\mathbf{N}}(n, L+W) - D_{opt}^{\mathbf{N}}(n, L) \\ &= \Delta\left(R_d, C_b, \frac{W}{n+1}, x_L^*\right) + (n-1)\Delta\left(R_b, C_b, \frac{W}{n+1}, y_L^*\right) \\ &\quad + \Delta\left(R_b, C_s, \frac{W}{n+1}, z_L^*\right). \end{aligned} \quad (7)$$

We split \mathbf{N} into two subnets \mathbf{N}_1 and \mathbf{N}_2 at the i th repeater. Therefore, the length of \mathbf{N}_1 is $L_1 = x_L^* + (i-1)y_L^*$ and the sink of \mathbf{N}_1 is the i th repeater of \mathbf{N} . The length of \mathbf{N}_2 is $L_2 = (n-i)y_L^* + z_L^*$ and the driver of \mathbf{N}_2 is the i -repeater. There are $i-1$ repeaters in \mathbf{N}_1 and $n-i$ repeaters in \mathbf{N}_2 . As $y_{L_1}^*, z_{L_1}^*, x_{L_2}^*$, and $y_{L_2}^*$ correspond to the distances between optimally placed repeaters in \mathbf{N} , the following equalities hold:

$$y_{L_1}^* = z_{L_1}^* = x_{L_2}^* = y_{L_2}^* = y_L^*. \quad (8)$$

Furthermore

$$x_{L_1}^* = x_L^* \quad \text{and} \quad z_{L_2}^* = z_L^*. \quad (9)$$

Let the i th repeater of \mathbf{N} be displaced by a distance W from its optimal location (W being positive toward the sink) and the remaining repeaters be placed optimally with respect to the displaced repeater. Therefore, the lengths of \mathbf{N}_1 and \mathbf{N}_2 change by W and $-W$, respectively. The delay of such a configuration is $\hat{D}(n, L) = D_{opt}^{\mathbf{N}_1}(i-1, L_1+W) + D_{opt}^{\mathbf{N}_2}(n-i, L_2-W) + T_b$.

From (7) and (8)

$$\begin{aligned} &D_{opt}^{\mathbf{N}_1}(i-1, L_1+W) \\ &= D_{opt}^{\mathbf{N}_1}(i-1, L_1) + \Delta\left(R_d, C_b, \frac{W}{i}, x_{L_1}^*\right) \\ &\quad + (i-1)\Delta\left(R_b, C_b, \frac{W}{i}, y_{L_1}^*\right) \\ &D_{opt}^{\mathbf{N}_2}(n-i, L_2-W) \\ &= D_{opt}^{\mathbf{N}_2}(n-i, L_2) + (n-i)\Delta\left(R_b, C_b, \frac{-W}{n-i+1}, y_{L_2}^*\right) \\ &\quad + \Delta\left(R_b, C_s, \frac{-W}{n-i+1}, z_{L_2}^*\right). \end{aligned}$$

Therefore

$$\begin{aligned}\hat{D}(n, L) &= D_{\text{opt}}^{\text{N}}(n, L) + \Delta\left(R_d, C_b, \frac{W}{i}, x_{L_1}^*\right) \\ &+ (i-1)\Delta\left(R_b, C_b, \frac{W}{i}, y_{L_1}^*\right) \\ &+ (n-i)\Delta\left(R_b, C_b, \frac{-W}{n-i+1}, y_{L_2}^*\right) \\ &+ \Delta\left(R_b, C_s, \frac{-W}{n-i+1}, z_{L_2}^*\right).\end{aligned}$$

Expanding the Δ s and substituting x_L^* for $x_{L_1}^*$, y_L^* for $y_{L_1}^*$ and $y_{L_2}^*$, and z_L^* for $z_{L_2}^*$, we obtain

$$\begin{aligned}\hat{D}(n, L) - D_{\text{opt}}^{\text{N}}(n, L) &= i \cdot \left[\left(\frac{rc}{2}\right) \left(\frac{W}{i}\right)^2 \right] + (rcx_L^* + R_dc + rC_b) \left(\frac{W}{i}\right) \\ &+ (i-1)(rcy_L^* + R_bc + rC_b) \left(\frac{W}{i}\right) \\ &+ (n-i+1) \cdot \left[\left(\frac{rc}{2}\right) \left(\frac{W}{n-i+1}\right)^2 \right] \\ &- (n-i)(rcy_L^* + R_bc + rC_b) \left(\frac{W}{n-i+1}\right) \\ &- (rcz_L^* + R_bc + rC_s) \left(\frac{W}{n-i+1}\right).\end{aligned}$$

Making use of the equalities in Lemma 1

$$\begin{aligned}\hat{D}(n, L) - D_{\text{opt}}^{\text{N}}(n, L) &= \left(\frac{rc}{2}\right) W^2 \left(\frac{1}{i} + \frac{1}{n-i+1}\right).\end{aligned}$$

Therefore, the maximum displacement for the i th repeater without violating the timing constraint $D_{\text{tgt}}^{\text{N}}$ is

$$W \leq \sqrt{\frac{2(D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L))(n-i+1)(i)}{(n+1)rc}}.$$

Hence, W_{FR} for the i th repeater is

$$W_{\text{FR}} = 2 \cdot \sqrt{\frac{2(D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L))(n-i+1)(i)}{(n+1)rc}}.$$

□

Theorem 2: For $D_{\text{tgt}}^{\text{N}} \geq D_{\text{opt}}^{\text{N}}(n, L)$, the width of the IFR for the i th repeater ($i \leq n$) of the net **N** is

$$W_{\text{IFR}} = 2 \cdot \sqrt{\frac{D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L)}{rc(2n-1)}}.$$

Proof: Consider the net **N** of length L having n repeaters as shown in Fig. 2. Let w_i be the displacement of the i th repeater from its optimal position (w_i being positive toward the sink). Let $w_0 = 0$ and $w_{n+1} = 0$ denote the “displacements” of the

source and the sink from their original positions, respectively. The delay for this configuration of repeaters is

$$\begin{aligned}D &= D(R_d, C_b, x_L^*) + \Delta(R_d, C_b, w_1 - w_0, x_L^*) \\ &+ \sum_{i=1}^{n-1} [D(R_b, C_b, y_L^*) + \Delta(R_b, C_b, w_{i+1} - w_i, y_L^*)] \\ &+ D(R_b, C_s, z_L^*) + \Delta(R_b, C_s, w_{n+1} - w_n, z_L^*) + nT_b \\ &= D_{\text{opt}}^{\text{N}}(n, L) + \frac{rc}{2} \sum_{i=1}^{n+1} (w_i - w_{i-1})^2 \\ &+ (rcx_L^* + R_dc + rC_b)(w_1 - w_0) \\ &+ (rcy_L^* + R_bc + rC_b) \sum_{i=1}^{n-1} (w_{i+1} - w_i) \\ &+ (rcz_L^* + R_bc + rC_s)(w_{n+1} - w_n).\end{aligned}$$

Making use of the fact that “displacements” $w_0 = w_{n+1} = 0$ and that the second last term of the preceding equation is a telescopic sum, we derive the following expression for the last three terms of the preceding equation:

$$\begin{aligned}(rcx_L^* + R_dc + rC_b)w_1 + (rcy_L^* + R_bc + rC_b)(w_n - w_1) \\ - (rcz_L^* + R_bc + rC_s)w_n\end{aligned}$$

which sum to zero because of the equalities in Lemma 1.

So, we obtain

$$D = D_{\text{opt}}^{\text{N}}(n, L) + \frac{rc}{2} \sum_{i=1}^{n+1} (w_i - w_{i-1})^2.$$

To meet the timing constraint $D_{\text{tgt}}^{\text{N}}$, we need $D \leq D_{\text{tgt}}^{\text{N}}$. Therefore

$$\frac{rc}{2} \sum_{i=1}^{n+1} (w_i - w_{i-1})^2 \leq D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L).$$

In the following, we derive the width of the IFR W_{IFR} . Let W be the maximum difference between w_i and w_{i-1} for $2 \leq i \leq n$. So, $|w_i - w_{i-1}| \leq W$ for $2 \leq i \leq n$, $|w_1 - w_0| \leq W/2$, and $|w_{n+1} - w_n| \leq W/2$. To find an upper bound for W , we solve for the following:

$$\frac{rc}{2} \cdot \left\{ (n-1)(W)^2 + 2 \left(\frac{W}{2}\right)^2 \right\} \leq D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L).$$

Therefore

$$W \leq 2 \cdot \sqrt{\frac{D_{\text{tgt}}^{\text{N}} - D_{\text{opt}}^{\text{N}}(n, L)}{rc(2n-1)}}.$$

By the definition of IFR, W_{IFR} is the maximum value of W for which the timing constraint is satisfied. Hence, the result follows. □

ACKNOWLEDGMENT

The authors would like to thank Professor J. Cong, D. Z. Pan, and T. Kong of UCLA for providing the floorplans of the MCNC

benchmark circuits. They would also like to thank them and Dr. K. K.-Y. Chao of Intel for their helpful discussions, and the anonymous reviewers for their constructive comments.

REFERENCES

- [1] J. Cong, T. Kong, and D. Z. Pan, "Buffer block planning for interconnect-driven floorplanning," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 358–363.
- [2] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integr. VLSI J.*, vol. 21, no. 1-2, pp. 1–94, Nov. 1996.
- [3] L. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1990, pp. 865–868.
- [4] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1995, pp. 138–143.
- [5] T. Okamoto and J. Cong, "Interconnect layout optimization by simultaneous Steiner tree construction and buffer insertion," in *Proc. ACM/SIGDA Physical Design Workshop*, Apr. 1996, pp. 1–6.
- [6] C. C. N. Chu and D. F. Wong, "Closed form solution to simultaneous buffer insertion/sizing and wire sizing," in *Proc. Int. Symp. Physical Design*, Apr. 1997, pp. 192–197.
- [7] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
- [8] R. Otten, "Global wires harmful?," in *Proc. Int. Symp. Physical Design*, Apr. 1998, pp. 104–109.
- [9] J. Cong and D. Z. Pan, "Interconnect delay estimation models for synthesis and design planning," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 1999, pp. 97–100.
- [10] J. Cong, "Challenges and opportunities for design innovations in nanometer technologies," in *Frontiers in Semiconductor Research: A Collection of SRC Working Papers*. San Jose, CA: Semiconductor Research Corp., 1997.
- [11] D. Wong and C. L. Liu, "Floorplan design of VLSI circuits," *Algorithmica*, vol. 4, no. 2, pp. 263–291, 1989.
- [12] H. Murata, K. Fujiyoshi, S. Nakate, and Y. Kajitani, "Rectangle-packing-based module placement," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1995, pp. 472–479.
- [13] M. Kang, W. W.-M. Dai, T. Dillinger, and D. LaPotin, "Delay bounded buffered tree for timing driven floorplanning," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 707–712.
- [14] H.-M. Chen, H. Zhou, F. Y. Young, D. F. Wong, H. H. Yang, and N. Sherwani, "Integrated floorplanning and interconnect planning," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 354–357.
- [15] P. Sarkar, V. Sundararaman, and C.-K. Koh, "Routability-driven repeater block planning for interconnect-centric floorplanning," in *Proc. Int. Symp. Physical Design*, Apr. 2000, pp. 186–191.
- [16] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and Z. Pan, "Interconnect design for deep submicron ICs," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 478–485.
- [17] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, Jan. 1948.
- [18] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *Proc. Design Automation Conf.*, June 1997, pp. 588–593.
- [19] R. Otten, "Graphs in floor-plan design," *Int. J. Circuit Theory Applicat.*, vol. 16, pp. 391–410, Oct. 1988.
- [20] P. H. Madden, "Partitioning by iterative deletion," in *Proc. Int. Symp. Physical Design*, Apr. 1999, pp. 83–89.
- [21] E. Shragowitz and S. Keel, "A global router based on a multi commodity flow model," *Integr. VLSI J.*, vol. 5, no. 1, pp. 3–16, Mar. 1987.
- [22] H. Shin and A. Sangiovanni-Vincentelli, "A detailed router based on incremental routing modifications: MIGHTY," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 942–955, June 1987.
- [23] R. Linsker, "A iterative-improvement penalty-function-driven wire routing system," *IBM J. Res. Devel.*, vol. 28, no. 5, pp. 613–624, 1984.
- [24] J. Cong and P. H. Madden, "Performance driven multi-layer general area routing for PCB/MCM designs," in *Proc. Design Automation Conf.*, June 1998, pp. 356–361.
- [25] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*. San Jose, CA: SIA, 1997.
- [26] LayoutSynth'92 Benchmark Information, Collaborative Benchmarking Laboratory, North Carolina State University. [Online]. Available: http://www.cbl.ncsu.edu/CBL_Docs/lys92.html
- [27] J. Cong, T. Kong, and D. Z. Pan, "Repeater block planning," unpublished, Jan. 2000.
- [28] J. Cong, T. Kong, D. Xu, F. Liang, J. S. Liu, and W. H. Wong, "Relaxed simulated tempering for VLSI floorplan design," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 1999, pp. 13–16.



Probir Sarkar was born in India. He received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, India, 1998 and the M.S. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2000.

He is currently a Design Engineer for Conexant Systems, Newport Beach, CA. His current research interest is VLSI CAD with emphasis on physical design and interconnect optimization.

Mr. Sarkar received the Tata Silver Jubilee Engineering Scholarship from 1994 to 1998, the Jagdish Bose National Science Talent Encouragement Award in 1995, and the Purdue University CSE Fellowship for Graduate Study in 1999.



Cheng-Kok Koh (S'92–M'98) received the B.S. and M.S. degrees in computer science from the National University of Singapore in 1992 and 1996, respectively, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1998.

He is currently an Assistant Professor of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. His current research interests include physical design of high-performance low-power VLSI circuits with an emphasis on VLSI interconnect layout optimization.

Dr. Koh received the Lim Soo Peng Book Prize for Best Computer Science Student from the National University of Singapore in 1990, the Tan Kah Kee Foundation Postgraduate Scholarship in 1993 and 1994, the GTE Fellowship in 1995 and the Chorafas Foundation Prize in 1996, both from the University of California at Los Angeles, the ACM Special Interest Group on Design Automation (SIGDA) Meritorious Service Award in 1998, the Chicago Alumni Award from Purdue University in 1999, and the National Science Foundation CAREER Award in 2000.