

Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning

Probir Sarkar Vivek Sundararaman Cheng-Kok Koh
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN-47907-1285
{sarkarp,sundarav,chengkoh}@ecn.purdue.edu

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% in [8] to 85%. Meanwhile, our approach is able to minimize 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.

1. INTRODUCTION

Due to the continued scaling of 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. [7] provides a comprehensive survey of these techniques.

Studies have shown that repeater insertion is among the most effective methods to optimize signal delay and noise [27; 15; 22; 4]. Without repeaters, the delay of a long RC wire is quadratic to the wire length. With judicious insertion of repeaters, the delay becomes linear [2; 19; 12]. However, most of these interconnect synthesis techniques were designed for post-placement interconnect optimization. In [5], it was projected that over 700K repeaters will be inserted for 50nm 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.

Floorplanning [28; 21], being 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 [14], 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 insertion of repeaters. To overcome this problem, [8] 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 in [3], 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 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 [8], we cluster individual repeaters into repeater blocks to result in 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 [8]. 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 2, we present the problem formulation. Section 3 derives the independent feasible region for repeater insertion. Section 4 presents the routability-driven repeater block planning algorithm. Experimental results are shown in Section 5, followed by the conclusion in Section 6.

2. 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, location, assignment and size of the repeater blocks

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD 2000, San Diego, CA.

Copyright 2000 ACM 1-581 13-191-7/00/0004...\$5.00

to be inserted in order to meet the timing constraints.

While 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.

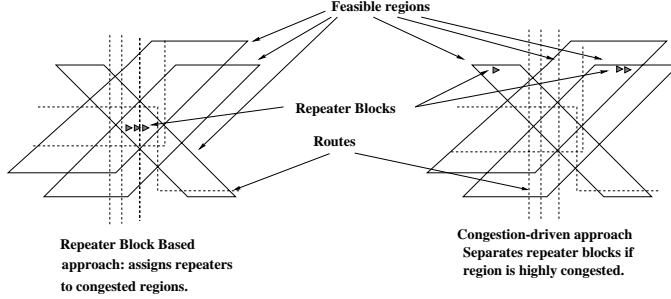


Figure 1: Impact of routing congestion on repeater block planning.

Figure 1 compares a routability-driven repeater block planner with a repeater planner that tries to minimize repeater blocks. Three feasible regions from repeaters of 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 on the left in Figure 1. On the other hand, a routability aware repeater planner would try to move the repeater blocks away from the high congestion zone. As shown on the right in Figure 1, 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 towards minimum number of repeater blocks; and the other one minimizes the routing congestion of the solution. 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.

3. FEASIBLE REGION COMPUTATION

In this section, we introduce the idea of *independent feasible region* (IFR) for repeater placement and obtain an expression for computing its width. We define the independent feasible region for a repeater to be the region where it can be placed such that the timing constraint of the net is satisfied, assuming that the other repeaters are also located within their respective independent feasible regions. The effectiveness of its use in repeater block planning is demonstrated by the significantly higher completion rates we obtain as compared to [8].

3.1 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 [6], and the Elmore delay formula [13] 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

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

Using the above expression, the Elmore delay of a single source, single sink net N (two pin net) of length l with n repeaters can be expressed as,

$$D_{net}^N(x_1, x_2, \dots, x_n) = D(R_d, C_b, x_1) + D(R_b, C_s, l - x_n) + \sum_{i=2}^{n-1} D(R_b, C_b, x_i - x_{i-1}) + nT_b,$$

where R_d is the driver resistance, C_s is the sink capacitance, and x_i is the location of the i^{th} repeater.

The optimal locations of the n repeaters for delay minimization of the net as shown in [1] are

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

where

$$x^* = \frac{1}{n+1} \left(l + \frac{n(R_b - R_d)}{r} + \frac{(C_s - C_b)}{c} \right),$$

$$y^* = \frac{1}{n+1} \left(l - \frac{(R_b - R_d)}{r} + \frac{(C_s - C_b)}{c} \right).$$

We denote the optimal delay for the net with n repeaters by

$$D_{opt}^N = D_{net}^N(x_1^*, x_2^*, \dots, x_n^*).$$

3.2 Independent Feasible Region

In [8], the “feasible region” for repeater insertion has been defined as the region where a repeater can be placed, assuming all the remaining repeaters are optimally placed, in order to satisfy a target delay constraint. Let W_{FR} denote the width of the “feasible region” for a given repeater.

As opposed to the definition of “feasible region”, the *independent feasible region* 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 independent feasible regions.

Formally, we define the independent feasible region (IFR) for the i^{th} repeater of a net N as

$$IFR_i = (x_i^* - W_{IFR}/2, x_i^* + W_{IFR}/2) \cap (0, l),$$

such that $(x_1, x_2, \dots, x_i, \dots, x_n) \in IFR_1 \times IFR_2 \times \dots \times IFR_n$ and $D_{net}^N(x_1, x_2, \dots, x_n) \leq D_{tgt}^N$. Here, W_{IFR} and D_{tgt}^N respectively denote the width of independent feasible region IFR_i and the target delay associated with the net.

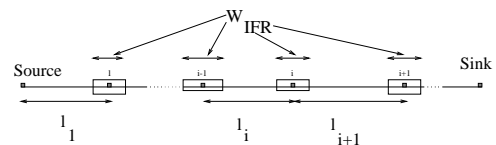


Figure 2: Independent feasible regions.

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 Figure 2). We have the following theorem for the width of the independent feasible region of a repeater.

THEOREM 1. For $D_{igt}^N \geq D_{opt}^N$, the width of the independent feasible region for the i^{th} repeater ($i \leq n$) of the net N is

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

□

The proof of the theorem is omitted due to page limitation. It is presented in [23]. It can be shown that when $n = 1$, $W_{FR} = W_{IFR}$. Although $W_{IFR} \leq W_{FR}$ for $n > 1$, its use results in a solution of higher quality because it facilitates a higher degree of fairness during global optimization of repeater block clustering. Moreover, the repeater block planning algorithm does not have to iteratively compute the new feasible region widths for repeaters in every iteration as in [8].

4. 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 [18] defined by circuit blocks as in [8]. We also assume that no repeaters can be inserted into circuit blocks.

For the purpose of routing congestion computation (to be described in Section 4.1), we divide the entire chip area into a set of rectangular routing tiles as shown in Figure 3. 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 [8], these repeater-block tiles represent locations where the repeaters may be placed. Not all repeater-block tiles are shown in Figure 3. In effect, we construct a *two-level* tile structure, one for the purpose of routing-congestion and the other for defining candidate repeater block (CRB) locations.

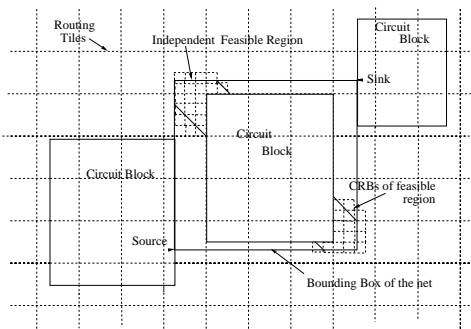


Figure 3: Creation of routing tiles and candidate repeater blocks.

For each repeater b to be inserted, we find S_b , the set of CRBs into which it can be placed. As shown in Figure 3, each repeater has several CRBs to which it may be assigned. The objective of the repeater block planner is to assign each repeater to a single CRB. Our approach to solving this problem is to first generate the candidate set for each repeater, and then use 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 $E(\mathcal{G}) = \{(b, c) : b \in \mathbf{B}, c \in S_b\}$. The essence of iterative deletion is to start 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 algorithm terminates, the output of the iterative deletion is a compatible assignment of repeaters to CRBs in terms of routability and repeater block count.

Repeater Block Planning Algorithm	
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 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;

Figure 4: Repeater Block Planning Algorithm

Figure 4 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 Figure 3. In Steps 2 and 3, we compute the minimum number of repeaters required for each two-pin net to meet its target delay [8; 1] and assign W_{IFR} of each repeater based on the available net slack ($D_{igt}^N - D_{opt}^N$) as defined in Theorem 1. Then, we generate the candidate set 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. The result is a bipartite graph \mathcal{G} constructed in Step 4.

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 4.3) seeks to remove the highest cost edge or the most incompatible repeater block assignment. As edges are iteratively deleted from the 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 4.1, and the edge weight, which is a composite cost function of the routing congestion cost and repeater block cost, in Section 4.2.

4.1 Congestion Model

The congestion model employed is essentially a two dimensional rectangular grid based probabilistic map assuming *two-bend* routing for each segment. This is similar to that developed in [3]. The *congestion* of a routing tile $tile(i, j)$ is defined as :

$$\begin{aligned} C_h(i, j) &= \text{Expected number of horizontal routes} \\ &\quad \text{passing through } tile(i, j), \\ C_v(i, j) &= \text{Expected number of vertical routes} \\ &\quad \text{passing through } tile(i, j). \end{aligned}$$

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 $tile(0, 0)$ and the sink to be located in $tile(m, n)$ and assume that a *bend* consumes both horizontal and vertical routing resources. It is easy to see that there are a total of $(m + n + 2)$ possible *two-bend* routes from source to sink. Assuming that all these routes are equally likely, we obtain the *probability* matrix shown in Table 1. $\delta C_h(i, j)$ is defined as the probability of a horizontal route passing through $tile(i, j)$. Similarly $\delta C_v(i, j)$ is defined for vertical routes.

Tile	$\delta C_h(i, j)$	$\delta C_v(i, j)$
$0 < i < m, 0 < j < n$	$\frac{1}{n+m+2}$	$\frac{1}{n+m+2}$
$0 < i \leq m, j = 0$	$\frac{n-i}{n+m+2}$	$\frac{1}{(m+n+2)}$
$i = 0, 0 < j \leq n$	$\frac{1}{(m+n+2)}$	$\frac{m-j}{n+m+2}$
$i = 0, j = 0$	1	1
$i = m, j = n$	1	1

Table 1: Probability matrix.

We define a *subnet* as the segment of a net between two consecutive repeaters, between the source and the first repeater, or between the last repeater and the sink. 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, C_h and 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 C_h and C_v as follows:

$$\begin{aligned} C_h(i, j) &= \sum_{\forall \text{subnets}} \delta C_h(i, j), \\ C_v(i, j) &= \sum_{\forall \text{subnets}} \delta C_v(i, j). \end{aligned}$$

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 [26; 25; 16]. For the purpose of this work, we use a monotonic piecewise cost function of the type proposed in [11] (see Figure 5).

4.2 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.

- *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

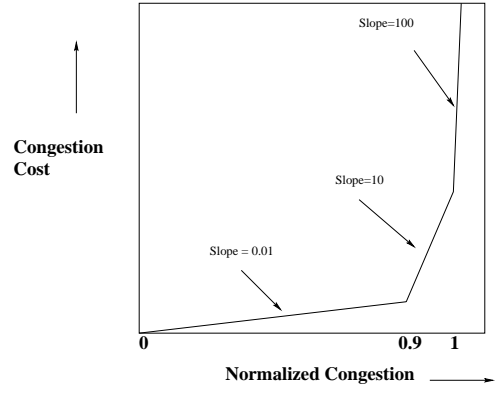


Figure 5: Piecewise-linear congestion cost function.

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 4.1. We denote the congestion cost of an edge $e = (b, c) \in E(\mathcal{G})$ as $CC(e)$.

- *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.

4.3 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 N denote the net to which repeater b belongs. We update the bipartite graph and its associated edge costs as follows:

- *Monotonicity Updates*: 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 non-monotonic repeater location assignments, we update the candidate sets for repeaters in the net N to make it monotonic. A CRB in 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 Figure 6, for example, the repeater assignments shown form a non-monotonic sequence

from the source to the sink. CRBs that do not lie on a monotonic path are removed.

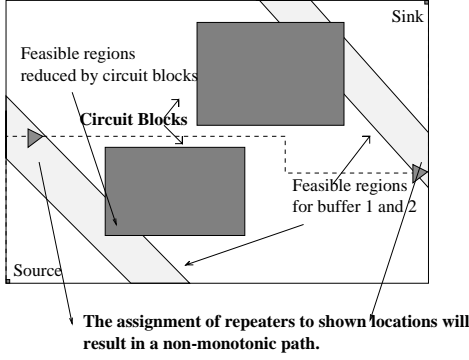


Figure 6: Non-monotonic repeater assignment.

- **Congestion Updates:** Deletion of the highest cost edge and the corresponding monotonicity updates change the solution set represented by the bipartite graph \mathcal{G} . The congestion update procedure reflects the impact of these changes onto the congestion matrices C_h and C_v .
- **Edge Cost Updates:** 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 assignment of a repeater to a repeater block is accomplished when the candidate set for the repeater consists of only one repeater block.

5. 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 3.1. The values (see Table 2) used for these parameters are based on a $0.18\mu\text{m}$ technology in the NTRS'97 roadmap [24].

	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 2: Parameter values for interconnect, repeater, driver, and sink.

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

Table 3: Details of MCNC benchmarks.

We report the results of our repeater block planner for six MCNC [17] benchmark circuits. The relevant details of these benchmarks are shown in Table 3. In this work 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 work were obtained from [9], and are the same as those used in [8]. These floorplans were generated by running simulated tempering using an improved Monte-Carlo technique [10].

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} [1], above which repeater insertion can be used for delay reduction. For each net we then compute the optimal delay T_{opt} obtainable by repeater insertion [1] and then randomly assign a target delay between 1.05 and 1.20 times T_{opt} as in [8]. As we generate these timing constraints on our own, a direct comparison between our method and that in [8] may not be fair even though we do use their results for reference in the following discussion.

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
[8]	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
[8]	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
[8]	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
[8]	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
[8]	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
[8]	1533 / 617	4263	-	0.84	13.98

Table 4: Comparison of repeater block planning solutions.

In Table 4 we report the following results from our repeater block planning algorithm : (i) 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, $\frac{MET}{NOTMET}$; (ii) the total number of repeaters inserted to meet timing constraints, N_{REP} ; (iii) the maximum tile congestion, C_{MAX} ; (iv) the chip area increase, δA expressed as

a percentage of the original chip area; and (v) CPU time required, TCPU. We also include the results for these benchmark circuits reported in [8].

Compared to [8], 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 [8] 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 3.

In Table 4, 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 4). In fact, all entries in N_{REP} , C_{MAX} , and δA include nets that do not meet the timing constraints after floorplan expansions.

Table 4 also shows that we generally require a smaller number of repeaters to achieve a higher completion rates, compared to [8]. In [8], all channels between blocks are expanded at the beginning of the planning step [9]. 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 4 also show that our algorithm is able to reduce routing congestion. With $p_1 = 1$ and $p_2 = 0$ the objective function is oriented towards the reduction of congestion cost, while 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. The Table 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 [8]. Moreover, the run-times of our algorithm are higher than those in [8].

6. CONCLUSION

In this paper we have presented a repeater block planning algorithm that uses the concept of independent feasible regions to do repeater clustering. The assignment of repeaters to repeater blocks is routability-driven and thus avoids regions of high routing congestion. Experimental results show that our technique performs significantly better than existing repeater planning methods.

7. ACKNOWLEDGMENTS

We would like to thank Prof. Jason Cong, Mr. David Z. Pan, and Mr. Tianming Kong of UCLA for providing us the floorplans for the MCNC benchmark circuits. We also thank them and Dr. Kevin K.-Y. Chao at Intel for their helpful discussions.

8. REFERENCES

- [1] C. J. Alpert, A. Devgan, “Wire Segmenting for Improved Buffer Insertion”, *Proc. Design Automation Conference*, pp. 588-593, 1997.
- [2] H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley, 1990.
- [3] H.-M. Chen, H. Zhou, F.Y. Young, D.F. Wong, H.H. Yang and N. Sherwani, “Integrated Floorplanning and Interconnect Planning”, *Proc. International Conference on Computer Aided Design*, pp. 354-357, 1999.
- [4] C.C.N. Chu and D.F. Wong, “Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing”, *Proc. International Symposium on Physical Design*, pp. 192-197, 1997.
- [5] J. Cong, “Challenges and Opportunities for Design Innovation in Nanometer Technologies”, *SRC Working Papers*, 1997.
- [6] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh and D.Z. Pan, “Interconnect Design for Deep Submicron ICs”, *Proc. International Conference on Computer Aided Design*, pp. 478-485, 1997.
- [7] J. Cong, L. He, C.-K. Koh and P.H. Madden, “Performance Optimization of VLSI Interconnect Layout”, *Integration, The VLSI Journal*, Vol. 21, pp. 1-94, 1996.
- [8] Jason Cong, T. Kong and D.Z. Pan, “Buffer Block Planning for Interconnect-Driven Floorplanning”, *Proc. ACM/IEEE International Conference on Computer Aided Design*, pp. 358-363, 1999.
- [9] J. Cong, T. Kong, and D.Z. Pan, *Personal Communication*
- [10] Jason Cong, T. Kong, D. Xu, F. Liang, J. S. Liu and W. H. Wong, “Relaxed Simulated Tempering for VLSI Floorplan Design”, *Proc. Asia and South Pacific Design Automation Conference*, pp. 13-16, 1999.
- [11] J. Cong and P.H. Madden, “Performance Driven Multi-Layer General Area Routing for PCB/MCM Designs”, *Proc. Design Automation Conference*, pp. 356-361, 1998.
- [12] J. Cong and D.Z. Pan, “Interconnect Delay Estimation Models for Synthesis and Design Planning”, *Proc. Asia South Pacific Design Automation Conference*, pp. 97-100, 1999
- [13] W.C. Elmore, “The Transient Response of Damped Linear Networks with particular regard to Wide-Band Amplifiers”, *Journal of Applied Physics*, 19(1), pp. 55-63, 1948.
- [14] M. Kang, W. W.-M. Dai, T. Dillinger, and D. LaPotin, “Delay Bounded Buffered Tree for Timing Driven Floorplanning”, in *Proc. of Int. Conf. on Computer-Aided Design*, pp. 707-712, 1997.
- [15] J. Lillis, C.K. Cheng and T.T.Y. Lin, “Optimal wire sizing and buffer insertion for low power and a generalized delay model”, *Proc. International Conference on Computer Aided Design*, pp. 138-143, 1996.
- [16] R. Linsker, “A Iterative-improvement penalty-function-driven wire routing system”, *IBM Journal of Research and Development*, 28(5), pp. 613-624, 1984.
- [17] <http://www.cbl.ncsu.edu/cbl.docs/lys92.html>
- [18] R. Otten, “Graphs in Floor-plan Design,” *International Journal of Circuit Theory and Applications*, vol. 16, pp. 391-410, Oct. 1988.
- [19] R. Otten, “Global Wires Harmful?”, *Proc. International Symposium on Physical Design*, pp. 104-109, 1998.
- [20] P.H. Madden, “Partitioning by Iterative Deletion”, *Proc. International Symposium on Physical Design*, pp. 83-89, 1999.
- [21] H. Murata, K. Fujiyoshi, S. Nakate and Y. Kajitani, “Rectangle-Packing-Based Module Placement”, in *Proc. of Int. Conf. on Computer-Aided Design*, pp. 472-479, 1995.
- [22] T. Okamoto and J. Cong, “Buffered Steiner Tree construction with wire sizing for interconnect layout optimization”, *Proc. International Conference on Computer Aided Design*, pp. 44-49, 1996.
- [23] P. Sarkar, V. Sundaraman, C.-K. Koh, “Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning”, manuscript available at <http://dynamo.ecn.purdue.edu/~chengkoh/work.html>.
- [24] Semiconductor Industry Association, *National Semiconductor Roadmap for Semiconductors*, 1997.
- [25] H. Shin and A. Sangiovanni-Vincentelli, “A detailed router based on incremental routing modifications: MIGHTY”, *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, CAD-6(6), pp. 942-955, 1987.
- [26] E. Shragowitz and S. Keel, “A Global Router based on a Multi Commodity Flow Model”, *Integration, the VLSI Journal*, Vol. 5, pp. 3-16, 1987.
- [27] L.P.P.P. van Ginneken, “Buffer placement in distributed RC-tree Networks for Minimal Elmore Delay”, *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 865-868, 1990.
- [28] D. Wong and C.L. Liu, “Floorplan Design of VLSI circuits”, in *Algorithmica*, pp. 263-291, 1989.