A New Approach To Sea-Of-Gates Global Routing+

Tai-Ming Parng* and Ren-Song Tsay**

Department of EECS and Electronic Research Laboratory
University of California, Berkeley, CA 94720

* on leave from Department of Electrical Engineering
National Taiwan University, Taipei,Taiwan

Abstract

The new approach integrates the concepts and techniques of two existing complementary approaches, namely the rerouting approach and the top down hierarchical approach based on linear assignment. It combines and enhances the advantages of the two approaches and results in a fast and high quality router which can handle large sea-of-gates design. In addition, it solves the problems of routing interdependence and routing resource estimation which heretofore have not been well addressed yet. The method has been implemented and successfully tested on three real gate-array chips: Primary1, Primary2, and a channelless industrial example with 100k gates.

1 Introduction

As a modern generation of gate-array design style, sea-of-gates architectures have introduced a new set of routing problems. Sea-of-gates chips can be channelled or channelless. They are usually very large in size and can provide designs of 100,000 or more gates. They could contain RAM, ROM, and other large blocks with the provision of routing over the cells. These characteristics impose new requirements on routing, for example, support of general area routing, using more than two metal layers, and being able to handle large chips with 100% completion rate.

As in other design styles, gate-array routing is divided into two steps: global routing and detail routing. Global routing usually breaks the array space into a two dimensional array of global routing cells (GRCs). By utilizing some congestion information on the boundaries of all GRCs, a global router determines which GRCs are used for each net and produces a coarse routing solution for the entire chip. To accomplish detail routing, a detail router needs to perform two tasks for each GRC. One is to decide an exact interface-pin location for every net on the GRC boundary to which the net is assigned in global routing. This task is usually called *interface-pin assignment*. The other task is to complete the routing by assigning a specific physical track or tracks for each signal within a GRC.

It is worth noting that interface-pin assignment can be treated as a *linear assignment problem* [1,2]. The problem can be briefly defined as follows: Given N nets, N routing tracks, and an N by N cost matrix which gives the cost of each net for a routing track, find an assignment of net to track such that a properly defined cost is minimum. As reported in [11], a linear assignment algorithm has been used to carry out the interface-pin assignment task of a gate-array routing system.

Related to interface-pin assignment, there exists another problem in existing routing systems. During a detail routing process, the GRCs can only be routed in a certain specific order. This is because each GRC boundary, with the exception of those on the chip boundaries, is shared by two neighboring GRCs and the GRC routed first decides the interface-pin assignment of the boundary of each of its neighboring GRCs. This makes the final detail routing result dependent on the routing sequence of GRCs and is generally referred to as the *routing interdependence problem*.

As for global routing, there have been many different published approaches, each of which has its own advantages and disadvantages. A general review of them for a variety of layout styles can be found in [5]. Here we focus on those for the gate-array layout style and briefly discuss some that are related to our approach.

First, the rerouting approach and the top down hierarchical approach are discussed. Typical examples for the rerouting approaches can be found in [8, 11]. This approach is based on a flat routing scheme, i.e., a non-hierarchical scheme. First, it uses maze routing or an approximate Steiner tree method to find an initial wiring for all the nets simultaneously regardless of routability (or overflow). Next, it applies repeated rip-up and reroute to improve the congestion. Due to the availability of detail congestion information on all GRC boundaries and the use of iterative improvement technique, tools based on this approach usually can achieve a very satisfactory completion rate. However, because the search space of rerouting grows very fast with the increase of chip size, this approach usually can not finish routing large chips in reasonable time.

The top down hierarchical approach, on the contrary, is very fast and can handle large chips by iteratively decomposing them into smaller sub-chips and solving them independently. To assign nets to GRC boundaries or coarse sections on the cut-lines, two techniques have been used. One [3] is based on linear programming, and the other one [6,10] uses a linear assignment algorithm. Both of them feature simultaneous processing of nets and can have a routing result that is independent of the routing sequence of nets. Nevertheless, they both have the intrinsic drawback of a pure hierarchical approach, i.e. lacking a clear view of local congestion of the whole chip being routed. Therefore, they may make final decisions at a higher level of decomposition that might turn out to be unfavorable at a lower level.

In addition to the problems discussed above, the problem of *how to evaluate routing resources* is one that has not been well addressed. To have a good routing result, it is very important to have accurate congestion information of all GRCs during the whole global routing process. Many existing literatures on global routing assume that the wiring capacity on each GRC boundary is given. Some proposed a method for estimating the demand and supply information by enumerating a set of possible paths of nets and computing the average number of crossing wires on each GRC boundary [7]. The estimates can provide a good global view of wire distribution, but not an accurate estimation of GRC boundary capacities. On the other hand, those systems based on the rerouting approach can obtain accurate GRC boundary capacities from existing wiring distribution at any instant. However, the fact that one can do a global wiring without exceeding the capacity at any boundary does not guarantee the completion of detail routing for all GRCs. Therefore, we also need to know detail congestion information inside every GRC. This is especially true for the channelless, general-area architecture. To sum up, an effective method for accurately estimating routing resources of sea-of-gates chips needs to be developed.

Finally, we present two observations that we got from the background information discussed above:

+The work is partially supported by the National Science Foundation under the grant MIP-88-3711 and the Joint Service Electronics Program under the grant F49620-87-C-0041.

* On leave under the support of National Science Council and National Taiwan University, Taiwan, ROC.

** Currently with IBM T.J. Research Center, Yorktown Heights, New York.

- The two global routing approaches, namely, the rerouting approach and the top down hierarchical approach, are complementary to each other. One's advantage constitutes the disadvantage of the other.
- Global routing and interface-pin assignment are assignment problem of different granularity. The former assigns nets of a whole chip to GRC boundaries, while the latter assigns each net of a GRC boundary to an exact routing track on the boundary. They are separated in existing systems to reduce complexity of a routing task at the cost of degraded routing quality.

Based on the above we have proposed a new approach to sea-of-gates global routing. The approach is aimed at meeting the routing requirements of sea-of-gates chips and resolving the difficulties of existing routing systems as noted above. The following sections will describe the main ideas and strategies of our approach as well as the detailed algorithm together with its runtime complexity, and implementation results.

2 The Proposed Approach

2.1 Routing Model

In deriving the routing model for general sea-of-gates architectures, we have made the following assumptions: (1) the chip size is fixed, (2) more than two routing layers may be used, (3) each routing layer has a predefined routing direction, (4) the distribution of physical pins and obstacles on the chip is arbitrary, and (5) the whole chip is superimposed by a set of horizontal and vertical coarse global routing grids and divided into a two-dimensional array of GRCs.

The inputs to the routing problem is a net list and an obstacle list. Each net in the net list describes a set of pins to be interconnected. A pin or an obstacle, which is represented as a point, a line segment, or a rectangle, is described in terms of its location, geometrical shape, and the routing layer or layers on which it resides.

Given the above assumptions and definitions, our global routing problem is to: (1) determine for every net which cells are used to route that net, and (2) perform the interface-pin assignment task for every GRC boundary.

2.2 Main Ideas and Key Strategies

The two main ideas of our approach are as follows:

1. Combine concepts/techniques of the two complementary approaches, i.e. the rerouting approach and the top down hierarchical approach based on linear assignment, and use them to their best advantage. The former can provide detail congestion information which can be used in the hierarchical approach to achieve high completion rate. In addition, the divide-and-conquer feature of the latter facilitates fast processing and is capable of handling large chips.
2. Linear assignment algorithm can be used to intergrate global routing and interface-pin assignment into a single task.

Based on these ideas and the routing model, a set of important strategies have been developed. They are presented as follows:

1. Apply top down decomposition to iteratively partition the chip into smaller and smaller routing blocks as the global routing process proceeds.
2. Apply pseudo-routing and use track usage profiles to produce a coarse global wiring and derive the detail congestion information.

   Pseudo-routing generates approximate rectilinear minimal Steiner tree for each net. Basically, it depicts the concept of the rerouting approach and utilizes both the techniques of routing and rerouting. However, instead of performing routing and rerouting to find a final solution, pseudo-routing explores only paths of minimal wire length and aims only at evenly distributing the wires among the GRCs.

   For every GRC, a horizontal track usage profile and a vertical track usage profile are generated to keep track of the number of used tracks on each horizontal and vertical detail grid lines of the GRC. Being derived from the result of pseudo-routing, the track usage profiles can provide detail and accurate congestion information for both the pseudo-routing procedure itself and the top down decomposition procedure.
3. For each top down decomposition step, apply a linear assignment algorithm to assign those crossing nets to a detail track position on the selected cut-line.

   This procedure not only carries out interface-pin assignment but also removes congestion within those congested GRC boundaries on the cut-line. These congested GRC boundaries are left unresolved during the pseudo-routing step. Viewing from the point of congestion removing capability, in some sense, this procedure is a parallel, incremental rerouting scheme because it *partially reroutes all* crossing nets on the cut-line. This capability helps greatly in reducing the number of rerouting iterations of the pseudo-routing step and hence the routing time.

We believe that these strategies form an ideal approach to sea-of-gates global routing. They make it possible to combine the advantages of the two existing complementary approaches. Moreover, the combination of pseudo-routing and the use of track usage profiles provides an effective way of routing resource estimation, while including interface-pin assignment into global routing solves the routing interdependence problem. Furthermore, due to the use of detail grids, the representations and computations of costs, congestion, and wire length are more accurate than those of other approaches, thus contributing to better routing results.

3 The Global Routing Algorithm

Based on the proposed approach, a global routing algorithm has been developed. In the algorithm the whole global routing process is divided into three phases: initial pseudo-routing, pseudo-rerouting, and block partitioning.

3.1 The Initial Pseudo-Routing Phase

This phase is to construct an initial wiring and generate preliminary congestion information. It is composed of two operation steps, namely the net tree generation and the tree edge rectilinearization. In addition, two track usage profiles and a set of capacity indices are also generated for every GRC during this phase.

Net Tree Generation The net tree generation step is to generate independently for each net a tree which is optimal in wire length with proper detour for big obstacles. This step involves two algorithms: the minimal spanning tree and the approximate minimal Steiner tree.

Tree Edge Rectilinearization The tree edge rectilinearization step is to find a rectilinear path for each tree edge which has end points located in different GRCs. At this step, except to avoid choosing impossible or heavily congested paths, only 1-bend paths of minimal length are generated.

Track Usage Profiles and Capacity Indices Associated with the net tree generation and the tree edge rectilinearization, two track usage profiles, one for horizontal component and one for vertical component of wires and obstacles, are also generated for each GRC. The track usage profiles can clearly reflect the congestion status. They indicate the lower bound of used routing tracks on each individual horizontal and vertical detail grid line in a GRC and are employed to compute a set of capacity indices for each boundary of the GRC. Both the profiles and the indices are used for guiding tree edge re-rectilinearization as well as tree regeneration in the later pseudo-rerouting phase. The capacity indices are also utilized in the block partitioning phase to guide the selection and partitioning of a cut-line.

In the following discussions we are interested in only those routing tracks that are perpendicular to each boundary.

Associated with each GRC boundary, two capacity indices C_f and C_s are maintained. C_f, the *free capacity*, indicates the number of free routing tracks of the boundary that are available

for accommodating wires from other congested GRC boundaries. C_s represents the *slack capacity*, i.e. an overestimation of routing track demand at the GRC boundary. For a GRC boundary, C_s indicates the number of tracks to be reserved and can not be assigned during the interface-pin assignment step in the block partitioning phase to be discussed later.

C_f and C_s are computed as follows:

$$D_e = max\{D_b, \frac{(D_b + P_{ll} + P_{ur})}{3}\} + C_{so} \quad (1)$$

$$C_f = C_t - D_e \quad (2)$$

$$C_s = D_e - D_b \quad (3)$$

where D_b is the *boundary demand*, i.e. the number of used tracks at the boundary;
P_{ll} and P_{ur} represent *peak demands*, the peak value of numbers of used tracks inside the two neighboring GRCs, i.e. the lower (for a horizontal boundary) or left (for a vertical boundary) one and the upper (for a horizontal boundary) or right (for a vertical boundary) one, of the boundary; these demand values can be obtained from the horizontal or vertical track usage profiles of associated GRCs;
C_t is the total number of tracks on the boundary and is called the *total capacity*;
C_{so} is the *slack offset capacity*, a static slack offset value assigned to the boundary and is assigned a value of about one tenth that of C_t;
D_e represents the *equivalent demand*, an equivalent routing track demand at the boundary and has taken into account the congestion status inside the two neighboring GRCs.

3.2 The Pseudo-Rerouting Phase

This phase is to iteratively adjust the net trees and tree edges such that the overall wiring distribution becomes as even as possible, by using the congestion information provided by the track usage profiles and the capacity indices. The pseudo-rerouting phase is composed of two operation steps, namely the tree edge re-rectilinearization step and the tree regeneration step. They are the same as the tree edge rectilinearization step and the tree generation step of the initial pseudo-routing phase, except that these steps involve both wire length and congestion as cost function parameters. Besides, associated with modifications on net trees and edges, both the track usage profiles and the capacity indices of each GRC are also updated during this phase.

3.3 The Block Partitioning Phase

This phase is to select a proper cut-line, assign those tree edges that cross the cut-line (hereafter called *global wires*), to track positions along the cut-line, and then divide the current routing block into two smaller subblocks. To do crossing position assignment, i.e. interface-pin assignment, for each global wire a cost function is constructed first. For any given track position on the cut-line, the cost function gives the cost for the global wire at that position. Then, a linear assignment algorithm is called to do the position assignment. The algorithm uses an object function that minimizes the total summation of the cost. After the assignment, each crossing position on the cut-line becomes an interface-pin for the global wire being assigned.

Cut-Line Selection Our method of cut-line selection is as follows: First,the preferred orientation of the cut-line, either a horizontal one or a vertical one, is decided. The horizontal one is chosen if the number of rows of the routing block is greater than the number of columns. Otherwise, the vertical one is chosen. After that, a cut-line is selected from all those candidate global grids of the selected orientation. The selection is based on the capacity index C_f, the free capacity, associated with each GRC boundary. For any candidate cut-line of the selected orientation, the cut will divide each of the global routing grids of the other orientation into two segments. Each segment may consist of one or more boundaries of GRCs. Only if the total free capacity of all boundaries of each such segment is greater than zero, then the candidate is qualified to be the cut-line of the current routing block.

Since there can be many qualified candidates, the one that is closest to the center of the current routing block has the highest priority. Because the wiring distribution has already been evened to a certain degree during the pseudo-rerouting phase, our method of cut-line selection can, without any degradation of routing result, achieve a quite balanced decomposition tree and hence the total runtime of the whole global routing process is reduced.

Cut-Line Partitioning The cut-line partitioning step divides the cut-line into sections, each of which corresponds to one or more contiguous GRC boundaries on the cut-line. For each section, the total free capacity of all GRC boundaries contained in the section should be greater than zero. In this step the global wires are also partitioned into groups. Those global wires that cross the same section are assigned into the same group. The purpose of this step is to reduce the problem size of the linear assignment algorithm which is applied later to perform interface-pin assignment.

Cost Function Generation The cost function plays an important role in the linear assignment algorithm. The final wire length and wiring quality around the cut-line are dependent on how good the cost function is. For a given global wire, the assignment cost of a track position on the cut-lines is computed based on four cost factors: routing length, preferred crossing boundary, existence of obstacles or pins, and number of vias. In general, for a global wire, a track positions that satisfy one or more of the following conditions will be given lower cost values: (1) it falls inside the bounding box of the global wire, (2) it is on the GRC boundary crossed by the rectilinearized tree edge of the global wire, (3) it has no neighboring obstacles (including pins of other nets), and (4) it is aligned with either end of the global wire. Details on generating cost functions for the linear assignment algorithm can be found in [6,9,10].

Interface-Pin Assignment After the cost function generation step, for each cut-line section and its associated group of global wires, the interface-pin assignment step is applied. A standard implementation of linear assignment algorithm [2] can be used for this step. Based on an object function that minimizes the total summation of cost, this step does an optimal assignment of crossing locations for all global wires of the section.

Subblock Generation This is the last step of the block partition phase. It divides the routing block into two subblocks by taking the linear assignment result of all cut-line sections and dividing each global wires into two subedges. This makes each net tree which crosses the cut-line become two independent subnets. Each subnet has one or more (for those nets that have multiple edges crossing the cut-line) interface-pins on the cut-line. At this time, the whole routing block becomes two independent subblocks and the block partition phase ends.

3.4 Runtime Complexity

The total runtime can be classified into two major items: the net processing time and the linear assignment time. The net processing time includes those for file input/output, pseudo-routing, track usage profile maintenance, and others that involve a complete scan of all nets. The linear assignment time is the time consumed in executing the linear assignment algorithm, which has a runtime complexity ranging from $O(N^2)$ to $O(N^3)$, where N is the number of nets to be assigned [1,2]. In general, the linear assignment time dominates over the net processing time for very tight designs. The reasons are: (1) many cut-lines are heavily congested and hard to be partitioned, and (2) the assignment algorithm runs at the complexity of cubic order with big problem sizes most of the time. On the contrary, for loose designs the net processing time dominates because the wire can be distributed evenly during pseudo-routing and most cut-lines can be fully partitioned into the smallest size, i.e. a GRC boundary. Based on a rough estimation, the overall time complexity of our global routing algorithm ranges from $O(n \log n)$ to $O(n^{1.5})$, where n is the total number of nets. For details of analysis, see [9].

4 Implementation Results and Status

Based on the proposed approach, a sea-of-gates global router SOGR has been implemented. There are about 12k lines of C codes in current SOGR implementation. Although a very simple pseudo-routing scheme is used and only 1-bend wiring paths are explored, current version SOGR has been successfully tested on three real circuits (without overflows on all GRC boundaries), including two channelled gate-array benchmark examples, namely Primary1 and Primary2 [4] and a three-metal-layer 100K sea-of-gates chip *h9* from the Hughes Aircraft Co. Table 1 shows the design statistics of these chips. Table 2 shows the statistics of the global routing results on a VAX 8800 (a 6-MIPs machine). Note that in Table 2: (1) the placement results used are generated from PROUD-I [12], (2) to reduce memory usage, the routing of *h9* chip has made use of disk storage for storing the first level and second level of unprocessed chip partitions, and (3) two values are given for the column of wire length vs. half perimeter ratio; the parenthesized one is computed with the length of physical pins included in the wire length. Fig. 1 shows a sample routing result of the *h9* chip.

5 Conclusion

In this paper, we present a new approach to sea-of-gates global routing. The sea-of-gates chips to be routed can be channelled, channelless, or of any arbitrary general architecture. There are four innovative ideas involved in our approach: (1) global routing with interface-pin assignment capability, (2) a methodology for combining two complementary global routing approaches, i.e. the rerouting approach and the top down hierarchical approach based on linear assignment, (3) use pseudo-routing and track usage profiles to provide accurate congestion information, and (4) use linear assignment technique as a parallel, incremental rerouting method. These ideas not only produce the new approach to global routing, they also induce solutions to two problems, namely the routing interdependence problem and the routing resource estimation problem, of existing gate-array routing systems.

Based on the new approach, a prototype system has been implemented. Several real circuits have been tested, and the results are very encouraging. Currently, we are working on implementing a complete pseudo-routing scheme using 2-bend wiring paths to explore further performance improvements. In addition, a detail router is being interfaced with the SOGR global router.

6 Acknowledgement

The authors would like to Prof. E.S. Kuh, M. Marek-Sadowska, and all students in their group for support and valuable discussions.

7 References

[1] F. Bourgeois and J.C. Lassalle, "An extension of Munkres algorithm for the assignment problem to rectangular matrices," *Commun. Assoc. Comput. Mach.*, Vol. 14, No. 12, pp. 802-805, Dec. 1971.

[2] R.E. Burkard and U. Derigs, *Assignment and matching problems: Solution methods with Fortran-programs*, Springer Verlag, 1980.

[3] M. Burstein and R. Pelavin, " Hierarchical wire routing," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-2, No. 4, pp. 223-234, Oct. 1983.

[4] *IEEE Workshop on Placement and Routing*, Research Triangle Park, NC, May 10-13, 1988.

[5] E.S. Kuh and M. Marek-Sadowska, "Global Routing," in *Layout Design and Verification*, ed. by T. Ohtsuki, Amsterdam: North-Holland, 1986.

[6] U.P. Lauther, "Top down hierarchical global routing for channelless gate arrays based on linear assignment," *VLSI'87*, C.H. Sequin(editor) Elsevier Science Publishers B.V. (North-Holland), IFIP, pp. 141-151, 1988.

[7] R. Nair, S.J. Hong, et al.,"Global wiring on a wire routing machine," *Proc. of ACM/IEEE 19th Design Automation Conference*, pp. 224-231, 1982.

[8] R. Nair, "A simple yet effective techniques for global wiring," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-6, No. 2, pp. 165-172, Mar. 1987.

[9] T.M. Parng, E.S. Kuh, and R.S. Tsay, "A unified approach to general-area routing," manuscript, 1989.

[10] M. Marek-Sadowska, "Route planner for custom chip design," *Proc. of IEEE ICCAD-86*, pp.246-249, 1986.

[11] B.S. Ting and B.N. Tien, "Routing techniques for gate array," *IEEE Trans. on Computer-Aided Design*, Vol. CAD-2, No. 4, pp. 301-312, Oct. 1983.

[12] R.S. Tsay, E.S. Kuh and C.P. Hsu, "PROUD: A sea-of-gates placement algorithm," *IEEE Design & Test of Computers magazine*, pp. 44-56, Dec. 1988.

Table 1: Statistics of tested chips

Chip name	Style	No. of modules	No. of nets	No. of pins
Primary1	channelled	833	904	3303
Primary2	channelled	3014	3029	12014
h9	channelless	26277	29151	92530

Table 2: Statistics of SOGR results

Chip name	GRC size (grids)	Wire Len./ Half Perim.	Mem. (MB)	Net Proc. time(Sec)	L. Assign time(Sec)	Total (Sec)
Primary1	23 x 23	0.99 (1.18)	1.66	30	18	48
Primary2	28 x 28	1.09 (1.27)	4.83	94	49	143
h9	25 x 25	1.17 (1.17)	12.76	836	1706	2542

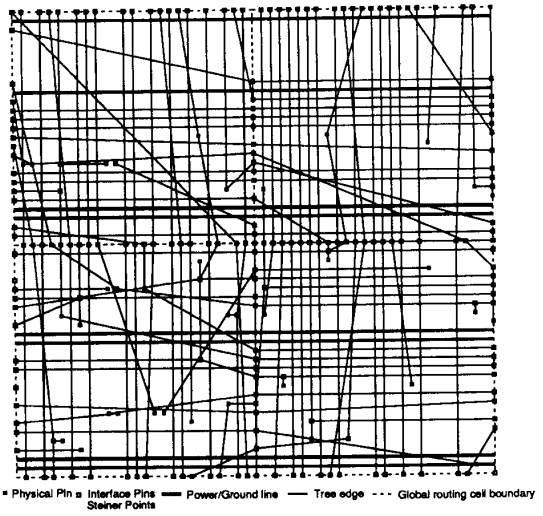

Figure 1: A random sample of an area of 2x2 global routing cells of the global routing result of the design *h9*.