E9.

GEAR: A GENERAL AREA ROUTER USING PLANNING APPROACH

Yuh-Lin Chen, Sao-Jie Chen, Chia-Chun Tsai, and Yu-Hen Hu*

Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, 10764, R.O.C.

*Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706, U.S.A.

ABSTRACT

A general area router (GEAR) based on the planning approach is proposed and implemented. Two meta-planning techniques, graceful retreat and least impact form the least commitment strategy in our router. These techniques are used to manage effectively the selection of net segments and the assignment of track resources, and to deliver a reasonable solution. The router can be applied to solve various kinds of routing problems, such as the channel routing, switch-box routing, staircase routing, rectilinear area routing with obstacles, and so on. Many examples extracted from the literature are experimentally tested and the obtained results can compete with many other routers in quality.

1. INTRODUCTION

Routing is an important step in realizing interconnections on the layout of VLSI chips, and it is alsoa well-known NP complete problem [1]. Conventionally, given an environment of module or block placement, the routing problem can be divided into two sequential stages: global and detailed routings. The goal of global routing stage is to complete the logical layout for all nets (that is, to divide the routing region into different routing areas, such as channels and switch boxes, and to assign uniformly the nets into these different routing areas). Therefore, the whole routing problem is divided into a collection of subproblems and each subproblem itself is a single area routing. In the detailed routing stage, the major work is to complete sequentially the physical layout of each area using a channel router or a switch-box router [2-10], and to obtain the final chip layout result.

However, for general macrocell (hierarchical module or building block) [11] and sea-of-gate design styles [12], the routing regions may not be completely defined as channels or switch boxes such that a channel or switch-box router can be directly applied to them. In fact, some other types of area routing, such as staircase channel (or L shape), switch box with obstacles, and rectilinear area region with odd-shaped obstacles, may also be generated in the stage of global routing. This would increase the load of detailed routing stage. Hence, a general area routing problem is arose and designing a general area router that can treat various different routing types becomes prevailing.

Although many routing tools [3-5, 8-10] may complete the layout of a channel routing and/or a switch-box routing, but may not succeed in coping with all types of routing in a general area. For example, the maze-running method [13-15] is a rather good candidate to the general area routing problem. But this method has some drawbacks. First, its memory requirement and time consumption will become unendurable when the problem size is large. Furthermore, since this approach routes only one net at a time without considering the existence and the influence of those unrouted nets; therefore, using this kind of router supplies no information on how to avoid the conflicts between connected and unconnected nets or to assure that some of the early connections will not block the successive connections. This blindness is usually caused by the myopia in identifying critical nets (which should be connected first) and crucial tracks (which should be assigned to the proper nets) at each step of the routing process.

In this paper, a general area router (GEAR) based on the planning approach [16-17] is proposed and implemented which attempts to solve all types of routing problems in a general area. The overall goal of the general area routing problem is to complete successfully all net connections with two or more layers. The goal can be programmed as a planning process in which the routing resources (e.g., tracks) are monotonously decreased. During the planning process, the goal is hierarchically decomposed into a conjunction of subgoals consisting of the selection of net segments and the assignment of track resources. These subgoals are interdependent on each other because they compete for some common and finite resources. In GEAR, two meta-planning techniques, derived from the least commitment strategy in [18] and called graceful retreat and least impact, are adopted to solve the above subgoal interactions. These techniques manage effectively the tasks of net selection and track assignment, and can thus promise a reasonable solution. With the same planning strategies our router can be applied to various types of routing problems, such as channel routing, switch-box routing, staircase routing, rectilinear area routing with obstacles, and so on. Many examples from the literature are used to testify our router. Experimentally, the router always generates competent layout results for all different types of routing examples.

Section 2 introduces the routing model with some definitions and the problem statement. The algorithm of GEAR is described in details in Section 3. Some experimental results are shown in Section 4. Finally, a conclusion is presented in Section 5.

2. ROUTING MODEL

For convenience, some terms (see Fig. 1) used throughout this article are defined as follows:

Chia-Chun Tsai is also with the Department of Electronic Engineering, National Taipei Institute of Technology, Taipei, Taiwan, R.O.C.

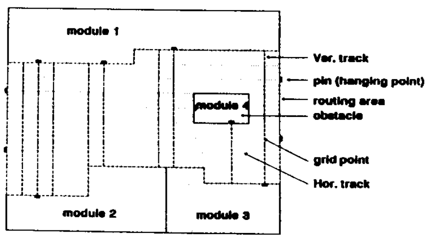

Fig. 1 General area routing model.

Rectilinear routing region or routing area: A region with rectilinear boundaries used for interconnection, which can be found among circuit modules in the environment of a layout after placement.

Pin: A terminal of a circuit module facing the routing area, which is to be connected to other terminals of a same net. Note that the pin locations must be aligned on grid points because a symbolic grid space is assumed in our router.

Net: A set of pins to be connected by some wire segments.

Contact or via: An electrical connection between two orthogonal wire segments on adjacent layers.

Path: A set of pins, vias, and wire segments implementing the interconnection between the pins of a net.

Component: A partial path of a net. Note that a unconnected pin may be viewed as a trivial component.

Hanging point: A point which belongs to a component of a net and needs to be connected to another points or components within the same net. Basically, each pin is a hanging point.

Obstacle: A prohibited area for all nets within the routing area. An obstacle may be a circuit module or pre-routed wire segments.

Track: An horizontal grid line (or row) or a vertical grid line (or column) in the routing area.

Since GEAR is designed to treat various different routing types, a general routing model, as shown in Fig. 1, is needed;some special characteristics in the model are described as follows:

1. An orthogonal grid space is spread on the whole routing region.
2. Two or more metal layers are available for routing.
3. Each layer can accommodate horizontal or vertical segments.
4. The pins of a net can be located at any position in the routing region.
5. Obstacles from blockage on specific layers.

The general area routing problem can be stated as following. Given a rectilinear routing area with obstacles, pins distributed on the whole routing area except on obstacles, and a net-list information, how to perform the interconnections of each net. The goal of this problem has two major objectives, one is to achieve 100% routing completion under a limited routing area and another is to minimize the number of vias and the wire length of each net. Our general area router based on planning techniques can achieve the above objectives and its algorithm is described in the next section.

3. ALGORITHM

Since AI planning approach is adopted, our routing algorithm is different from other heuristic routers. Basically, a method adopting AI approach needs heuristic experience collected from other routers and a powerful inference engine to cooperatively help reaching the goal of a problem. These obtained experience from other routers are collected as a set of local rules, and they form the so called crucial connection rules of GEAR. The graceful retreat and least impact meta-planning techniques [16-17], derived from the idea of least commitment strategy [18], are retreated as an inference engine. However, backtracking may be required if it is needed during the routing process and post refinement to improve the layout quality. Hence, our routing algorithm consists of four major parts: crucial connection rules, least commitment strategy, backtracking, and post refinement These four parts will be depicted in details as follows.

3.A Crucial Connection Rules

Since a routing area consists a set of finite tracks and a set of nets which will compete for these limited tracks, it is necessary to carefully assign the nets to these tracks in order to obtain a successful layout result for a given routing area. According to our experience, some of tracks are dedicated for some of nets and these tracks have to be assigned for those nets to avoid routing failure, and some of nets can be firstly completed to obtain the shortest path and minimal number of contacts or vias. Fortunately, these routing heuristics can be easily collected from routers published in the literature. These experience can be classified into four local rules: line-connect, corner-connect, one-bend-connect, and unique-extension [6,9], as discussed below. The first stage of the general area routing, called crucial connection, uses these local rules to form an interacting planning scheme.

To obtain the crucial connections, we have to assume that each hanging point (initially, each pin is also a hanging point as defined in Section 2) has an extension direction toward the routing area according to which layer the hanging point is on. Basically, odd layers prefer the horizontal direction while even layers prefer the vertical direction. Along the extension direction, each hanging point is extended out until reaching another point which is either the end-extended point of another hanging point or another un-extended hanging point. The extension of a hanging point is defined as a part of the horizontal or vertical track between the hanging point and the end-extended point. As in Fig. 2(a), the point n and the line qn are the end-extended point and the extension of the hanging point q, respectively. If the end-extended point is occupied by another net. the crucial cost for the hanging point is set to be some percentage of the extension. This percentage which is a user defined parameter and should be equal to or below 50% of the extension because there are two nets competing for this whole track. If the end-extended point is occupied by the same net or is a free point, the extension of the hanging point is set to be the whole track. Hence, the extension of each hanging point can be found by using the above method.

Once the extensions of all hanging points have been defined, the information will serves as a guidance for the local rule planning to decide how to connect these hanging points at the stage of crucial connection, and if some other nets try to use these extensions, they will be penalized by assigning a large crucial cost. During the local-rule planning, if the extensions of any two hanging points within the same net have one of following three cases: (1) a crossing intersection, (2) co-occupy the whole track, and (3) a crossing intersection via one-bend extension which consists of free grid points, then interconnect them immediately depending on the local rules of corner-connect, line-connect, and one-bend-connect. These cases are shown in Fig. 2(a), 2(b), and 2(c), respectively. These crucial connections can match the requirement of minimum path and vias, and impose the least possible impact for other unconnected nets.

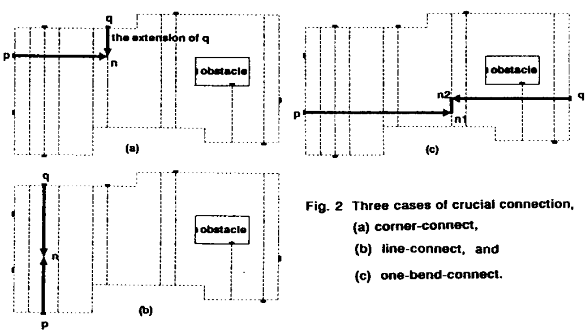

Fig. 2 Three cases of crucial connection,
(a) corner-connect,
(b) line-connect, and
(c) one-bend-connect.

During the process of above three crucial connections, some of other hanging points near in the current hanging point are possibly forced to extend by the local rule of unique-extension. However, if the extension of any one of these hanging point will be blocked, the process of backtracking is called to solve this problem.

3.B Least Commitment

As discussed above, only a few net paths are connected at the stage of the crucial connection, but there are many other waited-nets to be routed. These connected net paths and unconnected net-pins are uniformly called components as defined in Section 2. When the number of components to be interconnected becomes large, there is no better way to effectively solve this problem except using heuristics. Here, our router employs a heuristic-based AI-planning approach to solve it. The AI-planning approach is based on an important strategy called least commitment that consists of two meta-planning techniques, graceful retreat and least impact. The least commitment strategy makes two major decisions: how to select a proper component pair and how to connect this pair of components. The graceful retreat is used to determine which component pair will be chosen to be connected next, and the least impact is used to decide how to assign tracks to the chosen component pair with the purpose of maximizing the routability for other unconnected nets.

In the graceful retreat phase, a plan is formed after choosing one component-pair from a net according to the following factors: solution space, routing cost, and Manhattan distance. The solution space is defined as the estimated number of optimal paths between any pair of components. One is said critical when the solution space of a pair of components is less than the threshold solution space which is a parameter defined by user. The routing cost is defined when the expansion wave propagates from one component (source)to the other one (target), by the following parameters: preferred direction, non-preferred direction, introduction of via and involvement of congestion regions. Experimentally, the cost is 1 for one unit of connection in the preferred direction and 10 in the non-preferred one, and the cost for a via is 25 and for a congestion region is 50. If the wave-expansion of the source component cannot reach the target component, no path exists between the pair of components and the routing cost of source component is set to be a quite large value. The Manhattan distance is defined as the rectilinear distance between any pair of components.

After completing the cost estimations of a component to other components for all nets, the strategy of graceful retreat is used to select a proper pair of components. First, it checks if there exists any component-pair which can be considered as critical pair. If there exists at least one critical pair (or no any critical pairs), choose the one(s) with the smallest routing cost from those critical pairs. If there is more than one such pair, choose the one(s) with the longest Manhattan distance. Finally, if there is still more than one, arbitrary choose one pair and pass to the phase of least impact for determining the interconnection to this pair of components.

In the phase of least impact, a modified maze router is applied to find a shortest connection path between the pair of components generated in the phase of graceful retreat. The strategy of least impact propagates the expansion wave with a propagated cost from the source component to the target component. The propagated cost function consists of connection in preferred and non-preferred directions, introduction of vias, and involvement of critical cost of other nets. The cost definition for a unit connection in the preferred and non-preferred directions and via were given above, the crucial cost is defined to be 100. With this propagated cost function, if a path between the pair of components exists, it will assign this path to the proper tracks for the purpose of maximizing the possibility of success of other unconnected nets; otherwise, call backtracking to relax the previous least commitment.

3.C Backtracking

From above discussion, when the planning solution reaches a dead-end, undoing the previous commitment is required. This work is called backtracking. Two kind of backtracking strategies, historic backtracking [10] and rip-up and rerouting techniques [19], are adopted to solve this problem in our router. If a dead-end is found in the process of unique-extension at the stage of crucial connection, the historic backtracking strategy is first applied to undo the previous track commitment and try to find another feasible path for continuing the later connections. If the first back-tracking cannot find such a path under a limited-cost defined by the user, the second backtracking strategy is considered to continuously handle the case.

When a dead-end occurred in a hanging point or net -pin is detected in the least commitment phase, the second backtracking strategy is tried to relax some part of previous committed connections such that the solution space is large enough for solving the dead-end. First, try to find a rip-up path with the minimum cost, then remove all net-connections in the path and connect those blocked hanging-points or net-pins. To determine a reasonable rip-up path, all disconnected nets are planned during the rip-up process, and the rip-up cost of a path composed of removal-cost of the nets, and the wire length and vias of the path. The removal-cost of a net is increased by a parameter defined by the user when a net is rip-up. Initially, the removal-cost for all nets are set to be zero. If the cost of backtracking exceeds the limited cost, the router algorithm stops processing and reports no layout solution for the area routing type.

3.D Post Refinement

When all nets are interconnected successfully during the planning process, it means that a feasible layout solution can be generated. To improve the quality of the layout solution, such as the wire length and the number of vias, a post refinement is finally considered. The major work of the post refinement is to do a rip-up and rerouting for each net at a time again, and try to refine a net path such that the net-path length and the requirement of used vias are minimal. Of course, the design rules must be preserved during the post refinement. Finally, a better layout result can thus be obtained by our router.

The general area routing algorithm discussed in previous sections can be summarized as below:

Step1: Crucial connection rules

Define the extension for each hanging point and make all possible crucial connections using local rules (line-connect, corner-connect, bend-connect, and unique-extension) to generate components. If any one of the hanging points is blocked, go to step 3 to do backtracking. Otherwise, if there are no disconnected nets, go to step 4 to do post refinement.

Step2: Least commitment

Choose a proper pair from the components using the strategy of graceful retreat and try to find a reasonable path for this pair of components using the strategy of least impact. If a connected path is interconnected successfully between the pair of components, go to step 1 to continue crucial connection; otherwise, go to step 3 to do backtracking.

Step3: Backtracking

Calculate the current backtracking cost for the historic backtracking and the rip-up and rerouting techniques. If the current backtracking cost is less than the user defined cost, perform undoing previous commitment and go to step 1; otherwise report failure to this area routing type and stop the program.

Step4: Post refinement

Do a rip-up and rerouting for each net again and output the successful layout result.

4. EXPERIMENTAL RESULTS

A general area router, GEAR, has been implemented using the standard C language and run on a SUN-4 workstation under BSD 4.2 UNIX operating system. The router can be applied to route various routing types which may be channel, staircase channel, switch box (with/without obstacles), rectilinear area with/without internal circuit modules, and so on. Some different routing types extracted from the literature [3-10] are tested by our router and used to present the effectiveness of our router.

First, several switch-box routing examples published in the literature [3-10] are considered. These examples are acknowledged as benchmarks of switch-box routing. Our router can successfully complete all these switch-box routing examples. Table 1 shows the experimental results and the comparison of GEAR to others, such as WEAVER [4], BEAVER [9], CODAR [8], PACKER [10], SILK [3], and MIGHTY [5]. The sources of the switch-box examples are introduced here: "difficult" is from the example of Burstein's difficult switch box; examples "more difficult" and "modified dense" are modified with smaller area from the Burstein's difficult and dense examples, respectively; examples "terminal intensive" and "augmented dense" are referred from [10]; and "comp_1" is the example which is the compressed variance of the "difficult" example. From Table 1, our router always obtains the 100% routability with satisfying wire lengths, vias, and run time compared to other routers. Fig. 3 shows the layout result of the example "more difficult".

Next, to show that GEAR can be applied to other routing types, such as channel routing, staircase (or L-shape) routing, and area routing with regula boundary and internal circuit modules, Figs. 4, 5, and 6 are accordingly included in this paper.

5. CONCLUSION

A general area routing algorithm based on AI-planning approach was presented. Two meta-planning techniques, graceful retreat and least impact, are employed to manage the complicated interaction among all net interconnections in a given area routing problem. The strategies are used to guide the proper selection of the net-path segment and the better assignment of given resources in the way that the possibility of success in the future is maximized, the amount of backtracking is minimized, and a satisfying solution should be found. With this planning approach, GEAR can complete successfully the layouts of various different routing types and the results are encouraging.

ACKNOWLEDGMENTS

This work was supported by the National Science Council, Taipei, Taiwan, under Grant NSC 79-0404-E002-33 and Grant NSC 80-0404-E002-27.

REFERENCES

[1] M. R. Garey and D. S. Johanson, Computers and intractability: a guide to the theory of NP-completeness, San Francisco, 1979.

[2] R. L. Rivest and C. M. Fiduccia, "A greedy channel router", Proc. 19th Design Automation Conference, 1982, pp. 418-424.

[3] Y. L. Lin, Y. C. Hsu, and F. S. Tsai, "A detailed router based on simulated evolution", Proc. of ICCAD, 1988, pp. 38-41.

[4] R. Joobbani and D. Siewiorek, "Weaver: A knowledge -based routing expert", IEEE Design & Test, February 1986, pp. 12-23.

[5] H. Shin and A. Sangiovanni-vincentelli, "A detailed router based on incremental routing modifications:Mighty", IEEE Trans. on Computer-Aided Design, November 1987, pp. 942-955.

[6] M. Marek-sadowska, "Two-dimensional router for double layer layout", Proc. 22nd Design Automation Conference, June 1985, pp. 117-123.

[7] G. T. Hamachi and J. K. Ousterhout, "A switch box router with obstacle avoidance", Proc. 21st Design Automation Conference, 1984, pp. 173-179.

[8] P. S. Tzeng and C. H. Sequin, "Codar: A congestion -directed general area router", Proc. of ICCAD, 1988, pp. 30-33.

[9] J. P. Cohoon and P. L. Heck, "Beaver: A computational-geometry-based tool for switch-box routing", IEEE Trans. on Computer-Aided Design, June 1988, pp. 684-697.

[10] S. H. Gerez and O. E. herrmann, "Packer: A switch -box router based on conflict elimination by local transformations", Proc. of ISCAS,1989, pp.961-964.

[11] W. M. Dai, H. H. Chen, R. Dutta, M. Jackson, E. S. Kuh, M. Marek-Sadowska, M. Sato, D. Wang, and X. M. Xiong, "BEAR: A new building-block layout system", Proc. of ICCAD, 1987, pp. 34-37.
[12] R. S. Tsay, E. S. Kuh, and C. P. Hsu, "PROUD: A sea-of-gates placement algorithm", IEEE Design & Test of Computers magazine, Dec. 1988, pp. 44-56.
[13] C. Y. Lee, "An algorithm for path connection and its applications", IRE Trans. Electron. Comput., September 1961, pp. 346-365.
[14] W. Heyns, W. Sansen, and H. Beke, "A line-expansion algorithm for the general routing problem with a guaranteed solution", Proc. 17th Design Automation Conference, 1980, pp. 243-249.
[15] C. C. Tsai, S. J. Chen, and W. S. Feng, "An H -V tile-expansion router," Proc. of National Computer Symposium, 1989, pp. 106-115.
[16] W. P. C. Ho, D. Y. Y. Yun, and Y. H. Hu," Planning strategies for switchboxrouting", Proc. of ICCD, 1985, pp. 463-467.
[17] Y. H. Hu and S. J. Chen, "GM_PLAN: A gate matrix layout algorithm based on artificial intelligence planning techniques", IEEE Trans. on Computer-Aided Design, Vol. 9, No.8, Aug.1989, pp. 836-845.
[18] M. Stefik, "Planning with meta-planning, (MOLGEN: PartII)", Artificial Intelligence,1981,pp.141-169.
[19] W. A. Dees and R. J. Smith, "Performance of inter-connection rip-up and reroute strategies", Proc. 18th Design Automation Conference,1981, pp.382-390

Example	router	wire length	vias	run time (second)
difficult	WEAVER	531	41	1500
	BEAVER	547	43	1
	CODAR	544	?	15
	PACKER	565	46	56
	GEAR	561	44	3
more difficult	MIGHTY	541	39	4
	BEAVER	536	42	1
	SILK	528	36	69
	CODAR	545	?	17
	PACKER	552	43	1400
	GEAR	537	40	3
terminal intensive	WEAVER	615	49	1800
	MIGHTY	629	50	?
	BEAVER	632	53	1
	SILK	616	49	?
	CODAR	630	?	21
	PACKER	630	53	210
	GEAR	626	50	4
augmented dense	MIGHTY	530	32	?
	BEAVER	529	32	1
	CODAR	529	?	10
	PACKER	529	32	31
	GEAR	529	31	2
modified dense	WEAVER	510	29	920
	MIGHTY	510	29	?
	BEAVER	510	29	1
	SILK	510	29	?
	CODAR	510	?	11
	PACKER	510	30	36
	GEAR	510	29	2
comp_1	MIGHTY	fail	-	-
	CODAR	529	42	50
	PACKER	528	45	160
	GEAR	521	41	2

Table 1 Comparison with other routers.

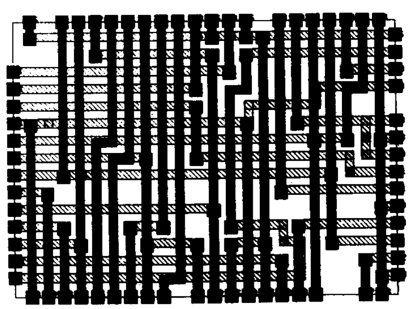

Fig. 3 Layout result of the switch-box example "more difficult".

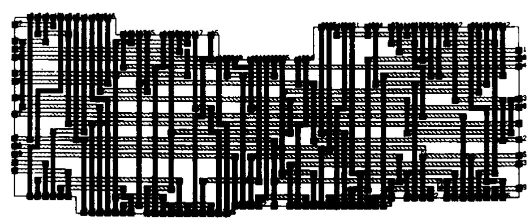

Fig. 4 Layout result of a channel example.

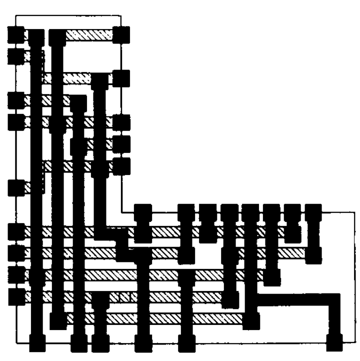

Fig. 5 Layout result of a L - shape example.

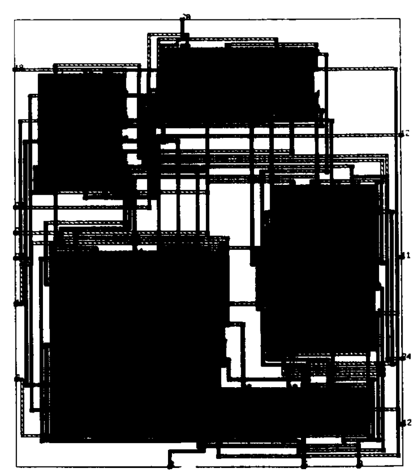

Fig. 6 Layout result of the example of regular boundary with four obstacles.