

General area router based on planning techniques

S.-J. Chen
C.-C. Tsai
Y.-L. Chen
Y.-H. Hu

Indexing terms: Area routing, Graceful retreat, Least impact, Planning

Abstract: A general area router (GEAR) based on the planning approach is proposed and implemented. Two meta-planning techniques, graceful retreat and least impact, are used to manage the selection of net segments and the assignment of track resources. Apart from the novel application of these planning techniques, we have also consolidated many effective routing heuristics into GEAR so that it is able to solve a variety of difficult routing problems. These include channel routing, switchbox routing, staircase routing, rectilinear area routing with obstacles, and other general area routing problems. Extensive simulation results indicate that GEAR is very competitive compared with the best known special-purpose routers.

1 Introduction

Routing is the process of allocating paths on an integrated-circuit chip or a printed-circuit board to physically interconnect terminals of different circuit modules with wire segments. It is a difficult (NP-complete) combinatorial optimisation problem [1], and has attracted much attention in past decades.

Owing to the complexity of the routing problem, the routing process is divided into two successive stages: global routing and detailed routing. In global routing, the task is to divide the entire routing region into disjoint, smaller routing areas, such as channels and switchboxes, and to assign nets to be routed into these different routing areas. In detailed routing, the goal is to complete the net routing within each subarea using channel routers, switchbox routers, or other special-purpose area routers [2-12].

However, for new design styles such as the macrocell (or building block) design [13], and the sea-of-gates design styles [14], the subdivided routing regions need not be strictly rectangular, as required by conventional channel or switchbox routers. Other routing area shapes,

such as staircase (or L-shape) channels, switchboxes with obstacles, or rectilinear areas with odd-shaped obstacles, may also emerge as a result of the global routing.

Many existing channel and switchbox routers are incapable of handling routing in general areas with more complex shapes. Although the conventional maze-running method [15] can handle the general area routing, it suffers the following drawbacks: First, and most important, as this approach routes only one net at a time, the interaction between different nets is not taken into account, and this leads to a suboptimal result or even a failure to complete routing. Moreover, the memory requirement and computing time of the maze-running router can easily become too large to be practical for real-world problems.

Some previous works on general area routers are summarised as follows. CODAR is a congestion-directed general area router proposed by Tzeng and Sequin [7] which uses 'ripup-and-reroute' techniques in its 'coarse wiring' and 'local modification' routing stages. CRACKER is a stepwise-reshaping area router proposed by Gerez and Herrmann [10], which first connects all the nets without considering conflicts to previously routed nets or obstacles, and then repeats the stages of 'connectivity preserving local transformations' until a solution without conflicts is finally achieved. Recently, Guruswamy and Wong [11] proposed a general multilayer area router which performs initial routing on a capacity-varying grid and calls for 'local and global modifications' to achieve a better layout result.

In this paper, a general area router (GEAR) [16] based on the planning approach [17, 18] is proposed, in which the routing process is decomposed into a conjunction of subgoals, with each subgoal consisting of the selection of net segments and the assignment of track resources. These interdependent subgoals compete with each other for track segments in the routing area. Two meta-planning techniques, graceful retreat and least impact, are adopted to manage the subgoal interactions and to minimise the need for backtracking. GEAR is designed to handle a variety of routing problems, which include channel routing, switchbox routing, staircase routing, rectilinear area routing with obstacles, and so on. We have conducted extensive experiments using examples taken from the literature to compare GEAR with the best existing routing methods, and the results are encouraging.

This work was supported by the National Science Council, Taipei, Taiwan, under grants NSC 82-0404-E002-166 and NSC 82-0404-E027-020

© IEE, 1994

Paper 1160E (E10), first received 5th May 1993 and in revised form 4th January 1994

Prof. S.-J. Chen and Y.-L. Chen are with the Department of Electrical Engineering, National Taiwan University, Taipei 10764, Taiwan, Republic of China

C.-C. Tsai is with the Department of Electronic Engineering, National Taipei Institute of Technology, Taipei 10643, Taiwan, Republic of China

Y.-H. Hu is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

2 Routing model

A grid-based general area routing model is shown in Fig. 1, where the routing area is symbolised as a collection of grid points, the tracks are aligned with the vertical or

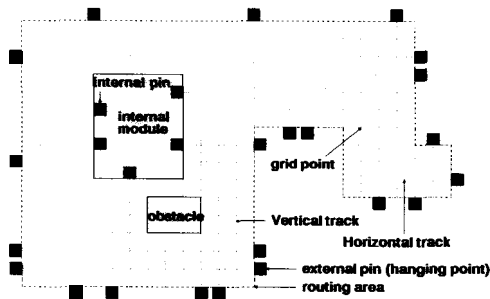


Fig. 1 General area routing model

horizontal line segments between grid points, and the pin locations are aligned to grid points and distributed on the boundaries of the internal circuit modules (internal pins) or on the boundaries of the routing area (external pins). A net is a collection of several connected wire segments which interconnect a set of (internal or external) pins. A path is the trajectory of a net. Two or more layers are available for horizontal and vertical wires, and a contact or via is used to interconnect two wire segments of the same net but on different layers. A partially routed net is called a partial net. A hanging point is the end point of an incomplete partial net. Initially, each pin is a hanging point. An obstacle is the prohibited area for all nets and may be an internal circuit module or a pre-routed wire segment. In addition, the following assumptions are made in GEAR:

- (i) Two or more layers are available for routing.
- (ii) Each layer may accommodate both horizontal or vertical wire segments.
- (iii) The pins of a net can be located at any position in the routing area.

With the above definitions and assumptions, the general area routing problem can now be stated as:

Given a net list and a rectilinear routing area, possibly containing obstacles, it is required to devise a routing scheme to achieve a 100% routing completion while minimising the number of vias and the wire length.

3 General area routing algorithm

3.1 Problem solving strategy-planning

Planning is a general problem-solving technique developed in artificial intelligence research [18]. A plan is a course of actions (subplans) to achieve a certain goal which usually consists of a collection of subgoals. Subgoals often interfere (interact) with each other, rendering suboptimal or infeasible solutions. The objective of planning, therefore is to derive a partial ordering of the subplans such that the overhead due to interaction among subgoals can be reduced. The main procedure for formulating problems in hand (e.g. VLSI physical design problems) into a planning scheme consists of the following two types of decision:

- (i) The decision to select a critical subgoal as the next task to perform.
- (ii) The decision to choose the least-impact subplan as a solution for the selected subgoal.

Note that, in the above formulation, the major concern is with the high-level problem-solving control strategies, which are domain-independent and are termed meta-planning (meta-control) strategies, whereas domain-specific heuristics are needed only at the implementation stage. The versatility of this planning scheme has been proven by the fact that it has been successfully applied in solving many VLSI physical design problems, such as, the switchbox-routing problem [17], the gate-matrix-layout problem [18], the circuit-partitioning problem [19], and now the general-area-routing problem. Fig. 2

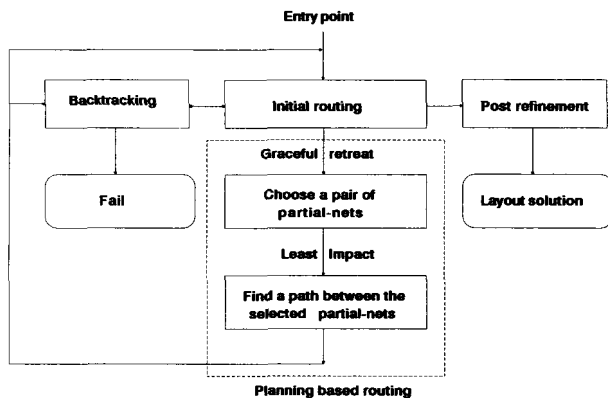


Fig. 2 Planning scheme of GEAR

shows a planning scheme for the general area router, and its details are described in the following Sections.

3.2 Overview of the GEAR algorithm

In addition to the use of the planning scheme, which takes care of the meta-control strategies only, GEAR is designed with both performance and efficiency in mind. Therefore, domain-specific routing heuristics developed by our own group, as well as reported in the literature, are adopted to improve the efficiency of the algorithm.

According to the scheme shown in Fig. 2, the GEAR algorithm is divided into four phases: initial routing, planning-based routing, backtracking, and post refinement. During the initial routing phase, unique extension of partial nets is made first, and nets which satisfy certain special patterns are routed. In the second phase, meta-planning techniques are applied to complete the routing of remaining nets by (1) selecting the most critical net(s) to route first, and (2) choosing the least-impact solution among many alternative routing paths. Here, the ordering of which net is the most critical and which selection has the least impact are determined using domain-specific heuristics. The third backtracking phase is evoked when some nets fail to be routed in phase two. The partial solution is retracted and the affected nets are rerouted. Finally, after successful routing of all nets, post adjustment is made to reduce the number of vias and the wire length.

3.3 Initial routing

(a) *Unique extension*: At the beginning, each pin is on the boundary of a routing area or an internal module. To make the connection to a pin, the only feasible path is through the line segment perpendicular to the edge on which the pin is positioned. As a two-pin net or a multi-pin net is decomposed into two or more pins, where each pin is considered as a hanging point in GEAR, the initial

extension of each pin in the routing region must be a 'unique extension' [4, 17]. The objective of unique extension is to extend these partial nets until they have more than one potential extension direction. We are not completing any connection in this step, but only repeatedly extending wires when possible.

(b) *Pattern-directed routing*: Once the unique extensions of all hanging points have been accomplished, these extensions serve as a guide to decide how to eventually connect the hanging points. In three special cases, we make a connection immediately. These are: (1) when two hanging points of the same net cooccupy the same track; (2) when two hanging points of the same net intersect at a cross-section, and (3) when two hanging points of the same net occupy adjacent parallel tracks.

Making the above connections immediately is a greedy heuristic which seeks to connect nets satisfying any of the above three patterns with their shortest possible paths. In terms of the planning-problem formulation, nets which satisfy one or more of these patterns are subgoals which have the least interaction (impact) with the remaining subgoals, and hence can be accomplished independently in the initial phase. Connections made based on the above three cases are called line connection, corner connection, and one-bend connection, as illustrated, respectively, by the nets 5, 7, and 8 in Fig. 3.

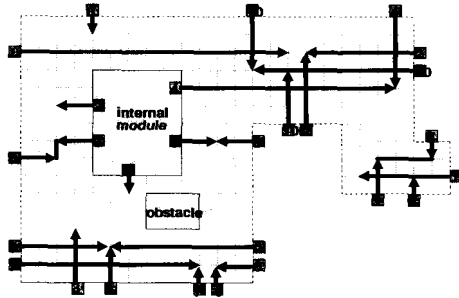


Fig. 3 An example of initial routing

3.4 Planning-based routing

Usually, there are only very few nets which are connected in the initial routing phase. The vast number of remaining partial nets will be routed during the second planning-based routing phase. In GEAR, the routing of an individual net is accomplished using a modified maze router, where two critical design decisions need to be made. (1) Which partial net will be selected to be routed first, and (2) among many alternative routing paths for the selected partial net, which one to choose?

The two meta-planning strategies elaborated in the following Sections, 3.4a and 3.4b, help to answer these two difficult questions. In particular, the graceful-retreat (GR) strategy is employed to select the next partial net to be routed; and the least-impact (LI) strategy is used to choose an optimal path for the selected partial net.

(a) *Selection of the most critical partial net*: A heuristic cost function consisting of the Manhattan distance (f_1) between the source and destination hanging points, the estimated routing cost (f_2), and the estimated number of potential paths (the size of solution space f_3) are evaluated for each partial net to determine its 'criticality'. Then, according to the GR strategy, partial nets with the highest criticality (i.e. lowest cost) are selected first.

The Manhattan distance is defined as the rectilinear distance between each pair of hanging points to be con-

nected:

$$f_1 = |x_m - x_0| + |y_n - y_0| \quad (1)$$

where (x_0, y_0) and (x_m, y_n) are, respectively, the coordinates of the source and the target hanging points.

The routing cost (f_2) consists of four factors:

$$f_2 = (g_1 \text{ or } g_2) + g_3 + g_4 \quad (2)$$

where g_1 is the cost to route the net along a 'preferred direction' (which means that the routing direction is expanded from the source hanging point towards the target hanging point); g_2 is the cost to route the net along other directions; g_3 is the cost when an extra via is needed during routing; and g_4 is the cost when a congested routing area is trespassed. Here, a 'congested' area means that those grid points of the area are already occupied by some other nets and thus available layers on those grid points are reduced.

The evaluation of the routing cost (f_2) and of the size of solution space (f_3) requires generating a wavefront propagation with the modified maze router. Empirically, in calculating f_2 , the weights in g_1 to g_4 are, respectively, determined as follows: $g_1 = 1*$ (the extension/path length along the preferred direction), $g_2 = 10*$ (the extension/path length along other directions), $g_3 = 25*$ (the number of new vias introduced during routing), and $g_4 = 50$ each time a designated 'congested area' is trespassed. If there is no path available between the pair of hanging points, the routing cost of such a partial net will be set to a very large value to represent infinity. In the above calculation of g_1 and g_2 , an estimated 'extension length' is used in the critical partial-net selection stage, and an exact 'path length' is used for choosing a least-impact optimal path among the many other feasible paths (described in Section 3.4b).

The size of the solution space (f_3) is defined as an estimated number of optimal paths between the source and the destination hanging points. It is the number of shortest paths between the source hanging point $S = (x_0, y_0)$ and the target hanging point $T = (x_m, y_n)$. If there are no obstacles inside the rectangular region formed with these two hanging points at opposite corners, the total number of optimal paths between these two points can be calculated easily. Specifically, the length of the optimal path is $L = |x_m - x_0| + |y_n - y_0|$. Out of the L track segments, $V = |y_n - y_0|$ must be vertical segments. Hence, the total number of optimal paths is $P(T) = TP = L!/[V!(L-V)!]$. However, owing to the existence of obstacles, it is more convenient to use the wavefront-propagation process of a maze router to compute the size of the solution space.

Let $P(X)$ denote the number of possible optimal paths from a source hanging point $S = (x_0, y_0)$ to an intermediate point $X = (x_i, y_j)$. Initially, let $P(S) = P(x_0, y_0) = 1$. Then, $P(X)$ can be calculated recursively according to the following formula (along the northwest direction):

$$P(X) = P(x, y_j) = P(x, y_{j-1}) + P(x_{i-1}, y_j) \quad (3)$$

Finally, the cost f_3 can be obtained using eqn. 3; that is

$$f_3 = P(T) = P(x_m, y_n) \quad (4)$$

To illustrate, assume that the source point S and target point T are to be connected through a region with obstacles (solid squares), as shown in Fig. 4. The size of solution space f_3 and the number of possible paths for each intermediate point X are listed in the figure. Note that, although the above estimation is made on a single layer, the multiple-layer case can be handled by assuming

a three-dimensional routing region and extending the wave propagation process of the modified maze router to handle this situation.

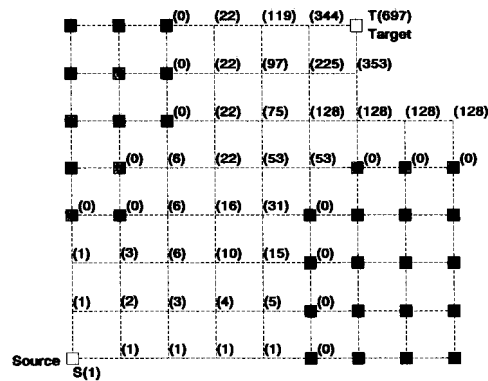


Fig. 4 Computation of the size of solution space for the partial-net pair (S, T)

■ Grid point occupied by other nets
 (#) Number of optimal paths from source to this grid

Experimentally, let the solution space threshold (SST) be defined as one-third of the total number of optimal paths between the source and the target handing points; that is, $SST = TP/3$. If the size of the solution space of a pair of hanging points is smaller than the above SST , the corresponding partial net will be labeled as 'critical'. Of course, critical partial nets always have a higher priority to be routed before 'noncritical' ones.

Finally, the selection of the most critical partial net is accomplished through the following decision rules based on the GR strategy:

- (1) Check there is any partial-net pair which is considered as 'critical'.
 - (1a) If there is more than one such pair, select the one(s) with the smallest size of solution space f_3 .
 - (1b) If there is more than one such pair, select the one(s) with the smallest routing cost f_2 .
 - (1c) If there is more than one such pair, select the one(s) with the longest Manhattan distance f_1 .
 - (1d) If there is more than one such pair, arbitrarily select one pair (using any other tie-breaking technique).
- (2) If there is no critical pair,
 - (2a) Select the one(s) with the smallest routing cost f_2 .
 - (2b) If there is more than one such pair, select the one(s) with the shortest Manhattan distance f_1 .
 - (2c) If there is more than one such pair, arbitrarily select one.

(b) *Choosing the least-impact routing path:* The routing of a selected partial-net pair is accomplished based on the the LI strategy. The modified maze router discussed in Section 3.4a is applied to find all feasible connection paths between the pair of selected hanging points, and the found 'path length' will be used in the calculation of g_1 or g_2 in the routing cost f_2 (as described in Section 3.4a) for each feasible path. To avoid trespassing on congested routing areas, a higher weight $g_4 = 100$ is used in calculating f_2 during this physical routing phase. This will produce the effect of the LI strategy, because if such a path which yields the smallest routing cost is chosen, it will have less chance to block the paths of other unrouted

partial nets. Therefore, this chosen path can serve as a candidate for the optimal path.

3.5 Backtracking

When the initial routing or planning-based routing fails to find a feasible track to route a selected net, previously routed nets will be removed to make room for rerouting. This procedure is called backtracking. Two backtracking strategies, chronological backtracking [20] and ripup-and-reroute [21], are adopted in GEAR.

If a dead-end is found in the phase of unique extension during the initial routing phase, all the routed partial nets will be discarded, and the chronological backtracking strategy is first applied to undo the previous track commitment and to try to find another feasible path for continuing later connections. If the first backtracking cannot find such a path after a finite number of trials, the second backtracking strategy will be considered to handle the case.

When a dead-end is detected in the least-impact net-routing phase, the ripup-and-reroute backtracking strategy is employed to undo part of the already connected nets to make room for routing the current net. To perform this kind of backtracking, first we need to find a path with the minimum ripup cost (defined below). Then we remove all the net connections in the ripup path to make space for those blocked hanging points or net pins. Finally, all nets disconnected during the ripup process are rerouted.

The ripup cost of a path is composed of the removal cost of the nets, the wire length, and via count of the path. Initially, the removal cost for a net is set to zero and it will be increased by one each time a net is being ripped up to avoid infinite looping. When the cost of backtracking (ripup cost) becomes larger and larger, it means that some nets are possibly being ripped up and rerouted repeatedly. If this cost overruns a predefined limit cost, the GEAR algorithm will stop the process and report no layout solution for this type of area routing.

3.6 Post refinement

When all nets are interconnected successfully during the first three phases of routing, a feasible layout solution will be generated. To improve the quality of the layout solution, for example to minimise the wire length and the number of vias, and to maximise the use of metal wires, a post refinement phase is needed. In this phase, each problematic low-quality netpath is removed and rerouted with a different set of routing parameters in f_2 . For example, trade-off between the total wire length and the number of vias can be made by changing the weighting parameters in g_1 (or g_2) and g_3 accordingly. Metal use maximisation can also be achieved by assigning a smaller weight on g_1 for the metal layers and a higher weight on g_2 for the polysilicon layer.

3.7 Extensions

In practice, some important nets, such as power/ground nets and critical nets, should be prerouted before detailed routing is carried out; these special cases can be taken into account in our router.

First, an area-routing problem with prerouted nets will be considered. If a whole net is needed to be prerouted, GEAR marks all the interconnections of the net as a series of obstacles, this area is identified as a prohibited resource and cannot be used. If only part of a net is prerouted, GEAR marks the portions of tracks used by this prerouted connection with a net name (which indic-

ates that these portions have already been assigned to that net), and then considers such a prerouted connection as a partial net, and continues routing from the current status.

Sometimes, electrically critical nets should be routed first using the minimum number of vias and wire length. To deal with this case, GEAR uses a different set of cost parameters (similar to the case described in Section 3.6). For example, the goal that a power net is preferably routed on metal (or metal 1) layer can be achieved by simply assigning a very high weight on g_2 for the polysilicon (or metal 2) layer.

In addition, GEAR can easily be extended to area routings with multiple layers. The only thing to do is to modify the computation of solution space for a partial-net pair (discussed in Section 3.4a) and the modified maze router (Section 3.4b) to the situation of a three-dimensional wavefront propagation.

4 Time complexity analysis

As shown in Fig. 2, there are four phases used in GEAR: (1) initial routing, (2) planning-based routing, (3) backtracking, and (4) post refinement, which have been discussed in detail in Sections 3.3 to 3.6.

To analyse the time complexity of the algorithm, suppose that an input net list has P pins and N nets, and T is the time complexity to finish the routing of a net within the routing area using a modified version of the maze router. If a routing area has m rows and n columns with l routing layers, then clearly $T = O(mnl)$. Furthermore, without loss of generality, assume that $m < n$ and l is a small constant, Then, $T = O(mnl) = O(n^2)$.

As to the complexity of subgoal (partial-net) selection in phase 2, most time is spent on the calculation of f_3 with a maze-running based method which takes $O(n^2)$, and there are at most N nets to select. Therefore, time spent on selecting a most critical partial net is $S = O(Nn^2)$. The complexity of the initial routing routine in phase 1 is $O(m + n) = O(n)$, which is negligible because it is far smaller than the complexity of connection path finding in the least impact of phase 2, which has a complexity of $T = O(n^2)$. During the backtracking in phase 3, because each net has a removal cost which is increased each time the net is ripped up and is compared with the predefined limited cost whenever a backtracking is called, the algorithm must terminate after a finite number of operations.

Therefore, the polynomial time bounds of GEAR can be derived as follows.

Case 1: Routing without backtracking: As the selection of a partial-net pair takes S time units, and there are at most $P/2$ connections to make, each taking T time, the time complexity is $(P/2)(S + T) = O(PNn^2)$ in the worst case.

Case 2: Routing with backtracking: Let the present 'limited cost' in the ripup stage be $C\text{-max}$. Then, the maximum number of backtracking is $O(N * C\text{-max}) = O(N)$ because $C\text{-max}$ is a constant. This implies that the algorithm must be terminated after a finite number of iterations, and the number of operations needed either to complete the routing or to report failure is $O(PNn^2)O(N) = O(PN^2n^2)$.

5 Experimental results

GEAR has been implemented using the standard C language and runs on a SUN-4 workstation under the BSD 4.2 UNIX operating system. Many different routing examples extracted from the literature [2-5, 7-10, 22-24] were tested by our router and used to illustrate its effectiveness. Tables 1 to 5 show the comparison of GEAR with other routers, such as SILK (using C on VAX-8550) [3], WEAVER (using OPS5 on VAX 11/780) [4], MIGHTY (using C on VAX 11/780) [5], CODAR (using C on Micro-VAX) [7], BEAVER (using C on SUN3) [8], and CRACKER (using LISP on Apollo DN4000) [10]. Data listed in the Tables are: the routing area formed with rows and columns (# Row_# Col), total wire segments (Wire_length), the number of vias (Vias), the number of backtracks (Backtracks), and running time measured in seconds (Cpu_time). The run time (Cpu_time) results are listed for reference only, as it is hard to compare routers running on different machines and using different languages. From the Tables, the number of backtracks required for each example was very small, and some examples showed no backtracking requirement. The symbol '-' means that the related data are not available.

5.1 Results of switchbox benchmarks

First, several switchbox-routing examples are considered. These examples are acknowledged as benchmarks for switchbox routing. Table 1 shows that our router always

Table 1: Results of switchbox benchmarks

Example (# row_# col)	Router	Wire_length	Vias	Cpu_time (s)	Backtracks
More_difficult (15 × 22) [5, Fig. 6]	BEAVER [8]	536	42	1	-
	MIGHTY [5]	541	39	4	-
	SILK [3]	528	36	69	-
	CODAR [7]	545	-	17.4	-
Terminal_intensive (16 × 23) [9, Fig. 3.4]	GEAR	540	39	3	2
	WEAVER [4]	615	49	1847	-
	BEAVER [8]	632	53	1	-
	MIGHTY [5]	629	50	11.5	-
Augmented_dense (18 × 16) [9, Figs. 3.5]	SILK [3]	616	49	-	-
	CODAR [7]	630	-	21.2	-
	GEAR	625	50	4	0
	BEAVER [8]	529	32	1	-
Comp_1 (14 × 22) [7, Fig. 2]	MIGHTY [5]	530	32	8	-
	CODAR [7]	529	-	10.5	-
	CRACKER [10]	529	31	2	0
	GEAR	521	41	2	0

Table 2: Results of irregular channels

Example (# row_# col)	Router	Wire_length	Vias	Cpu_time (s)	Backtracks
L_Shape [23, Fig. 11]	Chen [23]	159	33	-	-
	CRACKER [10]	161	21	110	-
	GEAR	164	21	2	0
Irregular_L_shape [23, Fig. 12]	Chen [23]	439	65	-	-
	GEAR	430	47	8	1
Irregular_boundary [24, Fig. 6]	Liu [24]	375	31	-	-
	CRACKER [10]	380	32	61	-
	GEAR	375	31	3	0
Irregular_channel (64 × 23) [2, Fig. 3.9]	GEAR	1717	117	58	1
Staircase_channel	GEAR	431	37	13	4

delivers 100% routability and the results of 'augmented_dense' [9, Fig. 3.5], and 'comp_1' [7, Fig. 2] are much better than those of other routers. Fig. 5 shows the layout result of example 'augmented_dense'.

5.2 Results of irregular channels

GEAR was also used to route some irregular channels; Table 2 shows our results. The channel example 'irregular_channel' is from Reference 2, Fig. 3.9. Two L-shape examples, 'L_shape' and 'irregular_L_shape' are from Reference 23, Figs. 11 and 12, respectively. The other example 'irregular_boundary', from Reference 24, Fig. 6, is a channel with irregular boundaries. In addition, we have created a 'staircase_channel' for testing. Fig. 6 shows the layout result for this example.

5.3 Results of areas with obstacles and internal modules

Some switchbox routings have rectilinear boundaries with a few obstacles or modules inside. These examples are called areas with obstacles and internal modules. In Table 3, the example 'switchbox_obstacle', from Reference 8, Fig. 23, is a switchbox with two obstacles, and the other example, 'area_2' [22] is an area with one internal module. Fig. 7 shows the layout result of example 'area_2'.

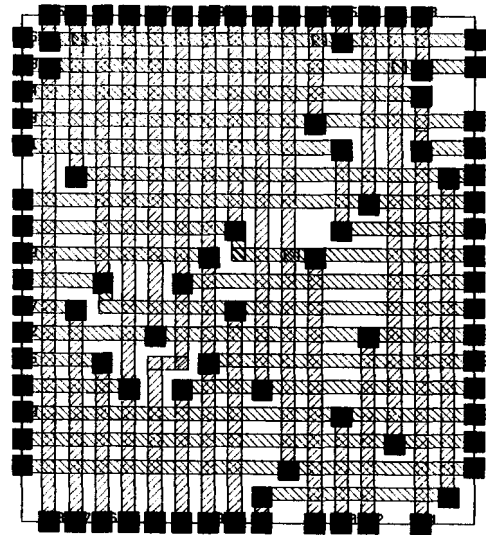


Fig. 5 GEAR's solution to the 'augmented_dense' switchbox [9, Fig. 3.5]

Table 3: Results of areas with obstacles or internal modules

Example (# row_# col)	Router	Wire_length	Vias	Cpu_time (s)	Backtracks
Switchbox_obstacle (10 × 14) [8, Fig. 23]	BEAVER [8]	168	16	-	-
	CRACKER [10]	180	19	70	-
	GEAR	168	20	1	4
Area_2 [22, pp. 70]	GEAR	403	43	21	6

Table 4: Results of areas with internal pins

Example (# row_# col)	Router	Wire_length	Vias	Cpu_time (s)	Backtracks
Array_11 × 11 (51 × 51) [7, A11]	MIGHTY [5]	fail	-	-	-
	CODAR [7]	3275	-	158	-
	GEAR	2757	93	446	1
Array_13 × 13 (71 × 71) [7, A13]	MIGHTY [5]	5220	-	706	-
	CODAR [7]	5544	-	175	-
	GEAR	5311	145	1134	1

Table 5: Results of a three-layer area

Example (# row_# col)	Router	Wire_length	Vias	Cpu_time (s)	Backtracks
Mirror_box (11 × 17) [7, Fig. 3]	CODAR [7]	-	45	-	-
	GEAR	398	42	4	1

5.4 Results of areas with internal pins

A special kind of routing type [7], called an $M \times M$ pin-array area, has the following characteristics: (1) the

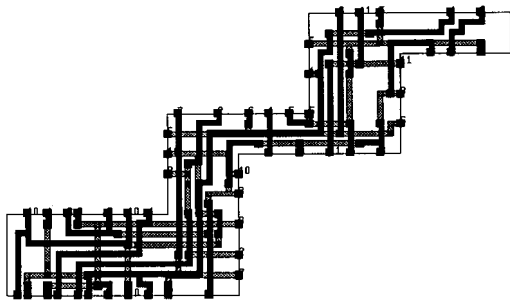


Fig. 6 GEAR's solution to the 'staircase_channel' example

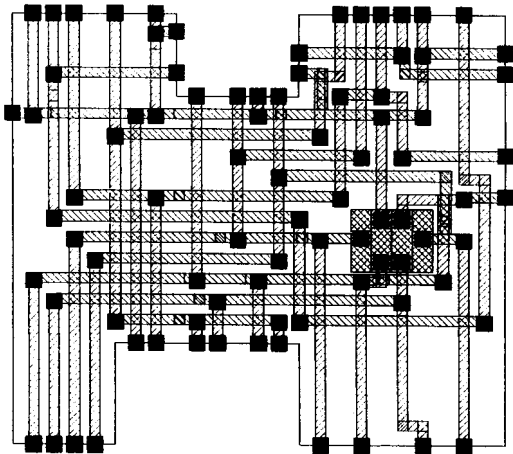


Fig. 7 GEAR's solution to the 'area_2' example [22]

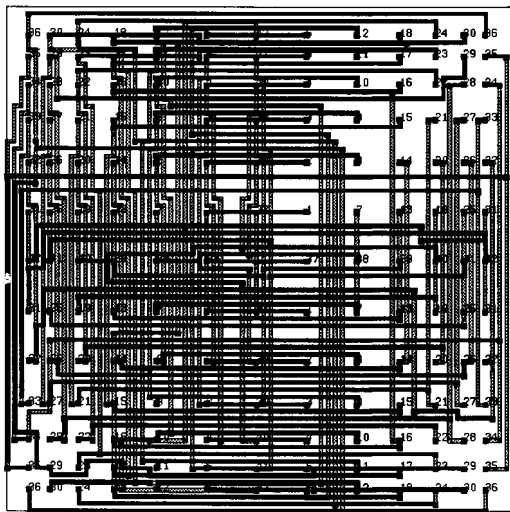


Fig. 8 GEAR's solution to the 'array_13 x 13' example [7, A13]

routing area has M by M pins, (2) it has no centre pin, and (3) nets are formed by pins symmetrically located on coordinates (x, y) , $(x, -y)$, $(-x, y)$, and $(-x, -y)$. Table

4 shows two such problems, the results achieved by GEAR, and the comparison of our results with those of MIGHTY [5] and CODAR [7]. Fig. 8 shows the layout result of example 'array_13 x 13' from Reference 7, A13.

5.5 Results of areas with multiple layers

GEAR can also be applied to multilayer routing problems. Table 5 shows the result of an example, called 'mirror_box', from Reference 7, Fig. 3, which is routed on a three-layer area and has nets with diagonally-opposed pins.

6 Conclusions

A general area routing algorithm based on a meta-planning approach has been presented. Two meta-planning techniques, graceful retreat and least impact, are used to manage the complicated interaction among all net interconnections in a given area routing problem. The strategies are to guide the proper selection of the net-path segment and to assign given resources in such a way that the possibility of future success is maximised, the amount of backtracking is minimised, and a suitable solution is found. With this planning approach, GEAR has been able to complete successfully the layouts of various different routing types, and the results are encouraging.

7 References

- HIGHTOWER, D.: 'A solution to the routing problems on the continuous plane'. Proceedings of 6th Design Automation Workshop, 1969, pp. 1-24
- HSU, C.P.: 'Signal routing in integrated circuit layout' (UMI Research Press, Ann Arbor, Michigan, 1986)
- LIN, Y.L., HSU, Y.C., and TSAI, F.S.: 'SILK: a simulation evolution router', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 1989, 8, (10), pp. 1108-1114
- JOBBANI, R., and SIEWIOREK, D.: 'WEAVER: A knowledge-based routing expert', *IEEE Des. Test Comput.*, 1986, 3, (1), pp. 12-23
- SHIN, H., and SANGIOVANNI-VINCENTELLI, A.: 'A detailed router based on incremental routing modifications: Mighty', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, CAD-6, (6), 1987, pp. 942-955
- MAREK-SADOWSKA, M.: 'Switchbox routing: a retrospective', *Integr. VLSI J.*, 1992, 13, pp. 39-65
- TZENG, P.S., and SEQUIN, C.H.: 'CODAR: A congestion-directed general area router'. Proceedings of International Conference on Computer-Aided Design, 1988, pp. 30-33
- COHOON, J.P., and HECK, P.L.: 'BEAVER: a computational-geometry-based tool for switch-box routing', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 7, (6), 1988, pp. 684-697
- LUK, W.K.: 'A greedy switch-box router', *Integr. VLSI J.*, 1985, 3, pp. 129-149
- GEREZ, S.H., and HERRMANN, O.E.: 'CRACKER: a general area router based on stepwise reshaping'. Proceedings of International Conference on Computer-Aided Design, 1989, pp. 44-47
- GURUSWAMY, M., and WONG, D.F.: 'A general multi-layer area router'. Proceedings of 28th Design Automation Conference, 1991, pp. 335-340
- TSAI, C.C., CHEN, S.J., and FENG, W.S.: 'An H-V alternating router', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 1992, 11, (8), pp. 976-991
- DAI, W.M., CHEN, H.H., DUTTA, R., JACKSON, M., KUH, E.S., MAREK-SADOWSKA, M., SATO, M., WANG, D., and XIONG, X.M.: 'BEAR: a new building-block layout system'. Proceedings of International Conference on Computer-Aided Design, 1987, pp. 34-37
- TSAY, R.S., KUH, E.S., and HSU, C.P.: 'PROUD: a sea-of-gates placement algorithm', *IEEE Des. Test Comput.*, 1988, 5, (6), pp. 44-56
- LEE, C.Y.: 'An algorithm for path connection and its applications', *IRE Trans. Electron. Comput.*, 1961, EC-10, (3), pp. 346-365
- TSAI, C.C., CHEN, S.J., CHEN, Y.L., and HU, Y.H.: 'Planning strategies for area routing'. Proceedings of European Conference on Design Automation, 1992, pp. 338-342

- 17 HO, W.P.C., YUN, D.Y.Y., and HU, Y.H.: 'Planning strategies for switchbox routing'. Proceedings of International Conference on Computer Design, 1985, pp. 463-467
- 18 HU, Y.H., and CHEN, S.J.: 'GM_PLAN: a gate matrix layout algorithm based on artificial intelligence planning techniques', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 1989, 9, (8), pp. 836-845
- 19 CHEN, S.J.: 'Iterative planning: a scheme for VLSI physical design problems', *Proceedings of the National Science Council, Part A*, 1991, 15, (5)
- 20 DE KLEER, J., *et al.*: 'Explicit control of reasoning', in WINSTON, P.H., and BROWN, R.H. (Eds.) *Artificial intelligence* (MIT Press, Cambridge, MA, 1979), pp. 83-116
- 21 DEES, W.A., and SMITH, R.J.: 'Performance of interconnection rip-up and reroute strategies'. Proceedings of 18th Design Automation Conference, 1981, pp. 382-390
- 22 LIU, W.: 'A short course on design automation for VLSI'. Lecture notes, National Chiao-Tung University, Hsing-Chu, Taiwan, Aug. 1990, 70
- 23 CHEN, H.H.: 'Routing L-shaped channels in nonslicing structure placement'. Proceedings of 24th Design Automation Conference, 1987, pp. 152-158
- 24 LIU, M.L., ZHUANG, W.J., and CHEN, S.Q.: 'Three-side channel, two adjacent-side channel and their routers'. Proceedings of International Symposium on Circuits and System, 1986, pp. 1226-1229