Multiterminal-Net Routing by Grid Stretching

Teofilo F. Gonzalez
Department of Computer Science
University of California
Santa Barbara, CA 93106

and

Si-Qing Zheng
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803

I. Introduction

Let R be a rectangle uniformly partitioned by w-1 vertical line segments and h-1 horizontal line segments. The set of lines (which include the rectangle boundary sides) is called the *grid* and the lines are called *grid lines.* The intersection of two grid lines is referred to as a *grid point.* A subset of grid points on the boundary of R without including the corners of R are referred to as *terminal points.* A terminal t is denoted by a pair ($x(t)$, $y(t)$) of the x and y coordinate values of t. The vertical (horizontal) grid lines are called *columns (rows).* The columns (rows) are labeled from left to right (bottom to top) with the integers 0 to w (0 to h). The set of terminal points is partitioned into m sets, $N_1, N_2, ..., N_m$. Each set N_i is called a *net*, and the set of nets is denoted by N. The problem of routing though a rectangle, which we call the *RRP problem* (which is also referred to as the *switch-box routing problem*), is denoted by $I = (R, N)$, and consists of finding a *layout* under the knock-knee wiring model for the set N of nets inside R. A layout under the knock-knee model for the set N of m nets consists of m edge-disjoint connected subgraphs $W_1, W_2, ..., W_m$ of R such that each W_i connects all terminals in N_i. It is well known that any knock-knee layout is wirable in four layers by using the algorithm in [1]. An *RRP* problem in which every net has at most k terminals is called an *RRP of degree k* or a *k-terminal-net RRP problem.* We define a *vertical cut* of R as the region between a pair of adjacent columns (c, c+1). Note that the two columns are not included in the cut. The *capacity of a vertical cut* is h+1, the total number of rows in R. The *density* of a vertical cut (c, c+1), denoted by $d_c^v(N)$, is the number of nets with at least one terminal to the left of the cut and at least one terminal to the right of the cut. A vertical cut (c, c+1) is *not saturated* if its capacity exceeds its density. These notions are similarly defined for horizontal cuts. We define $d^v(N) = \max \{d_c^v(N) \mid 0 \leq c < w\}$ and $d^h(N) = \max \{d_c^h(N) \mid 0 \leq c < h\}$ as the *vertical density* and *horizontal density* of I, respectively. The two-terminal-net *RRP* has been extensively studied. The fundamental theorem for routability of a two-terminal-net *RRP* was established by Frank ([2]), and Mehlhorn and Preparata ([5]).

Theorem 1.1: A two-terminal-net *RRP* is routable iff the revised row and column criteria hold ([2]). Furthermore, if these conditions are satisfied a layout can be constructed in $O(n \log n)$ time, where n is the number of terminals [5].

The concept of revised row and column criteria is required in theorem 1.1. However, since this concept is not relevant to our discussion we do not elaborate on it. Interested readers can find additional details in [2] and [5]. The following corollary of theorem 1.1 allows us to simplify the presentation of our results.

Corollary 1.1: For a two-terminal-net *RRP*, $I = (R, N)$, if every vertical cut and horizontal cut of R is not saturated, then I has a layout in R [2]. Furthermore, such a layout can be generated in $O(n \log n)$ time, where n is the number of nets [5].

The problem of determining whether or not an *RRP* instance of arbitrary degree is routable is an NP-complete problem [9], and thus it is unlikely that an efficient algorithm for its solution exists. However, any *RRP* problem instance is routable if enough rows and columns are introduced. In [5], an algorithm for routing any instance of the *RRP* problem by introducing additional rows and columns is presented. This algorithm is based on theorem 1.1. For any rectangle R we use $A(R)$ to represent the area of R. We say that rectangle R' is a *stretched version of rectangle R* if R' is obtained from R by adding zero or more rows and columns. We say that *OPT* is an *optimal area layout* for $I = (R, N)$ if R^*, the smallest rectangle that includes *OPT*, is a stretched version of rectangle R, and $A(R^*) \leq A(R')$ for any rectangle R' that is a stretched version of R and (R', N) is a routable *RRP* problem instance. Note that our definition of optimality is with respect to all layouts with a number of rows and columns that is at least as large as the number of rows and columns of R, respectively. Let R'' be the rectangle obtained from R by adding $d^v(N)$-(h+1) ($d^h(N)$-(w+1)) columns (rows) between two adjacent columns (rows) if $d^v(N) > h+1$ ($d^h(N) > w+1$). Clearly, $A(R'')$ is a lower bound for the area of an optimal layout for R. Hereafter, we assume that for any given *RRP*, $I = (R, N)$, $d^v(N) < h+1$ and $d^h(N) < w+1$; and we use $A(R)$ as a lower bound for $A(R^*)$, where R^* is the smallest rectangle enclosing *OPT*. For a rectangle R with height h and width w, the *aspect ratio of R*, denoted as $r(R)$, is defined as $\max\{h,w\}/\min\{h,w\}$. We assume without loss of generality that $w \leq h$.

In [5] it is shown that for any *RRP* problem $I = (R, N)$ a layout can be constructed inside R^f, a stretched version of rectangle R, such that asymptotically $A(R^f)/A(R) \leq 4$. The idea behind this algorithm is to stretch R into R^f and introduce a set of wires so that $I = (R, N)$ is transformed into a routable two-terminal-net *RRP* problem instance $I' = (R', N')$. Since I' is routable, its layout can be constructed by the algorithm for the two-terminal-net *RRP* given in [5]. The area bound of four for this transformation method results from the indiscriminating rule of introducing new grid lines. In this paper, we present a set of transformations *different* from the ones given in [5] that provide smaller approximation bounds for the unrestricted *RRP* and the three-terminal-net *RRP* problems. In section II, we show that if every net in a routable *RRP* contains no more than three terminals, then a layout can be constructed in a rectangle R^f such that asymptotically $A(R^f)/A(R) < 24/13$. The three-terminal-net *RRP* is very important because in practice nets have degrees are bounded by a small constant [8]. For the unrestricted *RRP* problem, we present in section III an algorithm that generates a layout in a rectangle R^f such that asymptotically $A(R^f)/A(R) < 3.5$. Due to the limitation of the space, we omit all proofs of our approximation bounds. It is important to point out the difference between theorem 1.1 and corollary 1.1. Corollary 1.1 guarantees a layout solution for two-terminal-net *RRP* instances $I = (R, N)$ such that $d^v(N) < h+1$ and $d^h(N) < w+1$, whereas theorem 1.1 guarantees a layout for some two-terminal-net *RRP* instances with terminals located at corners of R, and even when $d^v(N) = h+1$ and/or $d^h(N) = w+1$. Our algorithms can be easily modified when we choose to use theorem 1.1 by introducing a very small constant number of additional grid lines.

II. Three-Terminal-Net RRP Approximation Algorithm

In this section we present an approximation algorithm for the three-terminal-net *RRP* problem. Given a three-terminal-net *RRP* problem $I = (R, N)$, we define a total ordering on terminal points as follows: we say that terminals $t' < t''$ iff $x(t') < x(t'')$, or $x(t') = x(t'')$ and $y(t') < y(t'')$. We define a net N_i with p terminals as a sequence $(t_{i,1}, t_{i,2}, ..., t_{i,p})$ such that $t_{i,j} < t_{i,j+1}$, $1 \leq j < p$. With respect to this ordering we say that $t_{i,2}$ is the *middle terminal* of the three-terminal net N_i. Note that with respect to the y ordering another terminal from net N_i might be the middle terminal. When

CH2909-0/90/0000/0396$01.00 © 1990 IEEE

we refer to the middle terminal we mean with respect to x. Let N^3 denote the set of three-terminal nets in N. The set N^3 is the same as set N in the problem instance given in figure 2.1. We partition N^3 into two subsets $TB = \{N_i \mid N_i \in N^3$ and at least two of its terminals are located on the horizontal boundaries of the rectangle R $\}$ and $LR = N^3 - TB$. Let $tb = |TB|$ and $lr = |LR|$. Assume that tb and lr are even numbers. When tb (lr) is odd, an additional column (row) is required. For the problem instance given in figure 2.1, $TB = \{N_1, N_2, N_5, N_7\}$, $LR = \{N_3, N_4, N_6, N_8\}$, $tb = 4$ and $lr = 4$. Our algorithm, *ROUTE3*, is given below.

Algorithm *ROUTE3*

(1) Let $l_1, l_2, \ldots, l_{tb}$ be such that the middle terminals of the nets in TB appear in sorted order, i.e., $t_{l_1,2} < t_{l_2,2} < \ldots < t_{l_{tb},2}$.
(2) Insert a row between rows h-1 and h.
(3) Let net $N_{l_{2i-1}}$ and $N_{l_{2i}}$ form a pair p_i for $i = 1, 2, \ldots, tb/2$.
(4) Transform each pair p_i of nets into two-terminal nets by following the rules given in Appendix I.
(5) Apply steps (1)-(4) to the nets in LR after rotating the rectangle 90 degrees.
(6) Add enough rows and columns so that the resulting two-terminal net problem $I^f = (R^f, N^f)$ is routable after deleting the row introduced by step (2) and the column introduced by step (5).
(7) Introduce the wire segment generated by the rules given in Appendix I (case 3); project all terminal points one grid unit inside rectangle R^f; and let $I' = (R', N')$ be the resulting two-terminal-net routable *RRP* problem.
(8) Apply the routing algorithm given in [5] to the two-terminal-net routable *RRP* problem $I' = (R', N')$.
(9) Use the layout generated by the previous steps to construct a layout for N^f inside R^f.

end of *ROUTE3*

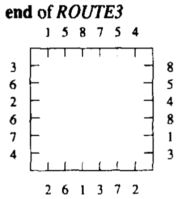

Figure 2.1

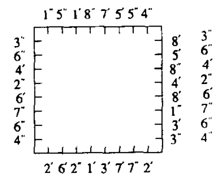

Figure 2.2

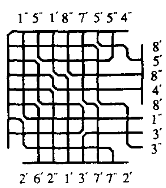

Figure 2.3

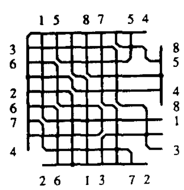

Figure 2.4

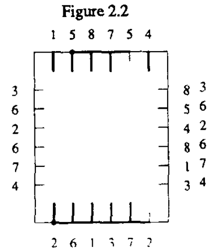

Figure 2.5

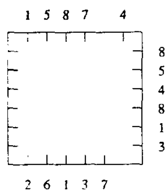

Figure 2.6

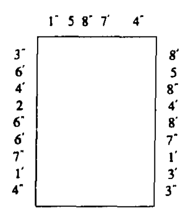

Figure 2.7

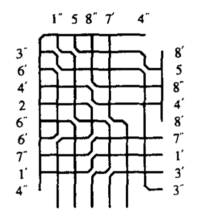

Figure 2.8

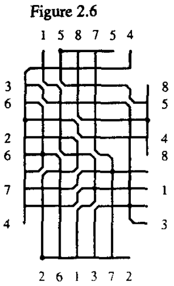

Figure 2.9

Algorithm *ROUTE3* transforms the problem instance given in figure 2.1 into the one given in figure 2.2. The pairs formed by step (3) are $p_1 = (N_2, N_1)$ and $p_2 = (N_7, N_5)$. In step (5) the pairs formed by the algorithm are $p_1 = (N_8, N_4)$ and $p_2 = (N_6, N_3)$. Since the rules given in appendix I do not introduce in this case fixed wire segments, our figures do not include the additional row and column introduced by steps (2) and (5). Note that no additional grid lines are introduced in step (6). Figure 2.3 shows a layout for I'. The layout was not constructed by the algorithm given in [5]. The reason is that for small problem instances a simple Ad-hoc layout can be easily constructed. All of our figures are drawn this way. Figure 2.4 shows the final layout. For *ROUTE*3, we have

Theorem 2.1: For any three-terminal-net *RRP* problem $I = (R, N)$ such that $d^v(N) < h + 1$ and $d^h(N) < w + 1$, algorithm *ROUTE3* constructs a layout in R^f such that asymptotically $A(R^f)/A(R) < 2$. Furthermore, algorithm *ROUTE3* takes $O(n \log n)$ time, where n is the number of terminals.

Based on algorithm *ROUTE*3, we can obtain a new algorithm *ROUTE3_ALT* which guarantees a smaller area bound.

Algorithm *ROUTE3_ALT*

(1) If $h \le 13w/8$, then apply algorithm *ROUTE*3 and stop;
(2) Let R^1 be a copy of R and let N^1 be N. Let α_t (α_b) be the number of nets with at least two terminals located on the top (bottom) side of R. We introduce α_t (α_b) rows between the topmost (bottommost) row and the top (bottom) side of R^1. The topmost (bottommost) rows are used to route the nets with two or more terminals located on the top (bottom) side of R^1. The layout for these nets is constructed by the algorithm given in [4]. For each net with exactly one terminal on the top (bottom) side of R^1, we project this terminal to the topmost (bottommost) empty row and for each net with exactly two terminals on the top (bottom) side of R^1, we project one of these two terminals to the topmost (bottommost) empty row.
(3) Let R^2 be the empty portion (without wires) of R^1. At this point there are two- and three-terminal nets. All the middle terminals (with respect to y) of the three-terminal nets are located on the left or right side of R^2. This routing problem is referred to as $I^2 = (R^2, N^2)$. The remaining three-terminal nets are split into two-terminal nets and rows are introduced using the transformation rules given in step (5) of algorithm *ROUTE3*. If the rules in appendix *I* introduce fixed wire segments, add a column between columns 0 and 1. When this additional column is introduced, project each terminal point located on the left side of the rectangle one unit towards the inside of the rectangle. Add enough columns so that the resulting problem, which we call $I^3 = (R^3, N^3)$ is routable.
(4) Let $I' = (R', N')$ be the resulting problem;
(5) Construct a layout for the two-terminal-net *RRP* problem I' using the algorithm given in [5];
(6) Construct from the layout for N' in R' and the partial layouts constructed in previous steps the final layout. Let R^f be the smallest rectangle enclosing the final layout.

end of *ROUTE3_ALT*

Since the problem instance given in figure 2.1 does not satisfy the condition $h > 13w/8$, let us apply steps (2)-(6) to the problem instance given in figure 2.1. Step (2) introduces the wire segments shown in figure 2.5 and the new problem instance given in figure 2.6. The resulting problem after step (3) of algorithm *ROUTE3_ALT* is given in figure 2.7 below. Figure 2.8 shows a layout for problem I' constructed by an Ad-hoc method rather than by the algorithm given in [5] for step (4) in algorithm *ROUTE3_ALT*. Figure 2.9 shows the final layout. For *ROUTE3_ALT*, we have

Theorem 2.2: For any three-terminal-net *RRP* problem $I = (R, N)$ such that $d^v(N) < h+1$ and $d^h(N) < w+1$, algorithm *ROUTE3_ALT* constructs a layout in R^f such that asymptotically $A(R^f)/A(R) < 24/13 < 1.85$ in $O(n \log n)$ time, where n is the number of terminals.

III. Approximations for the RRP Problem of Arbitrary Degree

In this section we present an approximation algorithm for the unrestricted *RRP* problem. We call net N_i a *k-side net* if its terminals are located on exactly k sides of R. We refer to the leftmost (rightmost) terminal of net N_i located on the top side R as the *left representative* (*right representative*) of N_i on the top side of R. Similarly, we define left representative and right representative for the terminals located on the bottom side of R. A net with at least two terminals located on the top (bottom) side of R will be referred to as a $t(b)$ net. Let d_t (d_b) be the vertical density of all t (b) nets when considering only their terminals located on the top (bottom) side of R. We apply two transformations that introduce new rows and a set of wires on these new rows to transform the given multiterminal-net *RRP* problem into a routable two-terminal-net *RRP* problem. Let us consider the first transformation. Let R^0 be a copy of R and let each terminal point in R^0 be in exactly the same position as in R. Between rows h (0) and h-1 (1) in R^0 add d_t (d_b) rows. The bottommost (topmost) of these rows will be called the *top (bottom) free row*. In case d_t (d_b) is zero the top (bottom) boundary is called the top (bottom) free row.

Algorithm *FIRST_TRANS*
Let R^0 be a copy of R and let each terminal point in R^0 be in exactly the same position in R. Introduce d_t (d_b) rows between row h (0) and row h-1 (1) in R^0;
The terminals located on the top (bottom) side of R^0 from all the t (b) nets are connected by wires which are routed in the topmost (bottommost) d_t (d_b) rows in R^0. This partial layout is constructed by the algorithm given in [4]. The rectangle, which we call R', is defined by the left and right boundary together with the top and bottom free rows of R^0. We shall refer to these rows as row h and row 0 of R'.
For each net $N_i \in N$ with at least one terminal located on the top (bottom) side of R^0 perform the following projection operation:
(i) If the net does not have terminals located on the left and right side of R^0 and N_i is a 2-side net, project the left representative of N_i to the top (bottom) side of R'.
(ii) If the topmost (bottommost) terminal located on the left or right side of R^0 is located on the left side of R^0, project the left representative of N_i to the top (bottom) side of R' and skip (iii).
(iii) If the topmost (bottommost) terminal located on the left or right side of R^0 is located on the right side of R^0, project the right representative of N_i to the top (bottom) side of R'.
After these projection operations, we transform each net N_i into another net N_i' which is identical to the original one if N_i does not have two or more terminals on the top or bottom sides of R^0. On the other hand, if N_i has at least two terminals located on the top or bottom sides of R^0, then N_i' is N_i without all the terminal points located on the top and bottom side of R^0 except the projected one.
Let N' be the set of all nets N_i'.
end *FIRST_TRANS*

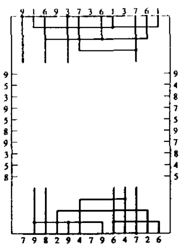

Figure 3.1

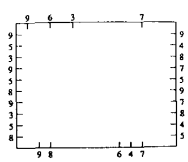

Figure 3.2

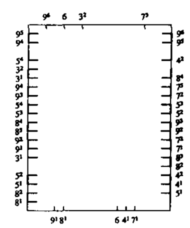

Figure 3.3

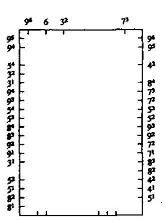

Figure 3.4

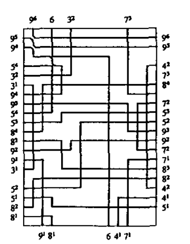

Figure 3.5

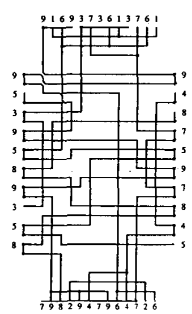

Figure 3.6

Figure 3.1 shows the partial layout constructed by the procedure, and figure 3.2 shows the resulting subproblem $I' = (R', N')$. In the second transformation we convert each net N_i' into a set of two-terminal nets. The transformation procedure SECOND_TRANS is given below.

procedure *SECOND_TRANS*
Let R^1 be R' with all the terminal points from the nets N_i';
Add a column between columns 0 (w-1) and 1 (w);
for j = 1 **to** h-1 **do**
if there is a terminal at any of the boundary grid points in row j of R^1
then begin
insert a new row between row j and row j-1 of R^1;
if left boundary point in row j of R^1 is a terminal t from net N_i'
then make a copy of t at the left boundary point of the newly introduced row;
if right boundary point in row j of R^1 is a terminal t from net N_i'
then make a copy of t at the left boundary point of the newly introduced row;
end
endfor
Add U wires to connect adjacent terminal from the same net and project each terminal one unit to the inside of the rectangle;
Let R'' be the rectangle after deleting the left and right boundary of R^1 and let R'' contain all the terminal points in R^1;
for each N_i' with terminals on the left or right side of R^1 **do**
let u be a terminal of N_i' on the left or right side of R^1 with the smallest y-coordinate value;
if N_i' does not have a terminal on the bottom side of R^1
then assign label i^1 to u and $j \leftarrow 1$;
else begin
assign label i^1 to the copy of u and the terminal in N_i' located on the bottom side of R^1;
if there is another terminal v of N_i' such that $y(v) \geq y(u)$
then assign label i^2 to u **else** $j \leftarrow 2$;
end
while there is an unlabeled terminal of N_i' located on the left or right side of R'' **do**
$j \leftarrow j + 1$;
let u be the unlabeled terminal of N_i' with the smallest y-coordinate value;
assign label i^j and i^{j-1} to u and the copy of u, respectively;
endwhile
if N_i' has a terminal located on the top side of R^1
then assign label i^j to the terminal of N_i' located on the top side of R^1;
else $j \leftarrow j$ - 1;
endfor
end of procedure *SECOND_TRANS*

Figure 3.3 and 3.4 show the resulting subproblem after applying procedure *SECOND_TRANS* to the problem instance I' shown in figure 3.2. Each net N_i' may be split into several two-terminal nets by procedure *SECOND_TRANS*. The k^{th} of such nets is defined by the label i^k. We use R'' to denote the rectangle extended from R^1 by *SECOND_TRANS*, and use N'' to denote all two-terminal nets defined on the boundary of R''. Our algorithm for the unrestricted *RRP* problem is given below.

Algorithm *ROUT_MULTINET*
Apply procedure *FIRST_TRANS* to obtain a routing instance defined in R' for net set N';
Apply procedure *SECOND_TRANS* to obtain a routing instance defined in R'' for net set N'';
Add enough rows and columns so that I'' is a routable two-terminal-net *RRP* problem instance;
Use the algorithm in [5] to route I'';
Construct a layout for N^f in R^f from the layout in N'' in R'' and the partial layouts constructed in previous steps.
end *ROUT_MULTINET*

Theorem 3.1: For any multiterminal *RRP* defined in a rectangle R such that $h \geq w$ and $d^* < w+1$, algorithm *ROUT_MULTINET* constructs a layout in a rectangle R^f extended from R such that asymptotically $A(R^f)/A(R) < 2+(3/2)/r(R) \leq 3.5$, where $r(R)$ is the aspect ratio of R. Furthermore, such a layout can be constructed in $O(n \log n)$ time.

Figure 3.5 shows the layout constructed for (R'', N'') by an Ad-hoc method rather than by the algorithm given in [5]. Figure 3.6 shows the final layout. Careful readers may notice that even if we are given an *RRP* instance $I = (R, N)$ which is not routable, i.e. $d^v(N) > h + 1$, algorithm *ROUT_MULTINET* can still guarantee a layout solution in R^f such that asymptotically $A(R^f)/A(R) < 2 + (3/2)/r(R)$.

IV. Concluding Remarks

Our area bounds are not small. This is mainly because the combinatorial properties of *RRP* are still not well understood. This is reflected in the lower bound for the layout area we used and the approach of splitting multiterminal nets into two-terminal nets that we adopted. One way to improve our results is to develop better lower bounds for the area of an optimal solution. This does not seem to be a simple problem. One of the main problems with the layouts for the unstricted *RRP* problem generated by our algorithms and the algorithm given in [5] is that the wires connecting the nets may be long. Our algorithm will suffer from this problem even if we use Frank's ([2]) algorithm instead of the one in [5]. At this time there is no way around this problem. The problem of minimizing wire length and area seems very interesting and deserves careful study.

V. References

[1] Brady, M. L. and D. J. Brown, "VLSI Routing: Four Layers Suffice," *Advances in Computing Research*, Vol. 2, 1984.

[2] Frank, A., "Disjoint Paths in Rectilinear Grid", *Combinatorica*, 2(4), pp. 361-371, 1982.

[3] Gao, S. and M. Kaufmann, "Channel Routing of Multiterminal Nets", *Proceedings of the 28th Symposium on Foundations of Computer Science*, pp. 316-325, 1987.

[4] Hashimoto, A. and J. Stevens, "Wiring Routing by Optimal Channel Assignment Within Large Apertures", *Proceedings of the 8th Design Automation Conference*, pp. 457-464, 1971.

[5] Mehlhorn, K. and F. P. Preparata, "Routing Through a Rectangle", *Journal of ACM*, Vol. 33, No. 1, pp. 60-85, 1986.

[6] Mehlhorn, K., F. P. Preparata and M. Sarrafzadeh, "Channel Routing in Knock-Knee Mode: Simplified Algorithms and Proofs", *Algorithmica*, Vol. 1, No. 2, pp 213-221, 1986.

[7] Preparata, F. P and W. Lipski, Jr, "Optimal Three-layer Channel Routing", *IEEE Transactions on Computers*, Vol. 33, No. 5, pp. 427-437, 1984.

[8] Preparata, F. P. and M. Sarrafzadeh, "Channel Routing of Nets Bounded Degree", *VLSI: Algorithms and Architectures*, North-Holland, Amsterdam, pp. 189-203, 1984.

[9] Sarrafzadeh, M, "Channel-Routing Problem in the Knock-Knee Mode is NP-Complete, *IEEE Transactions on Computer Aided Design*, Vol. CAD-6, No. 4, pp. 293-303, July 1987.

APPENDIX I: TRANSFORMATION RULES.

In this appendix we present the rules for splitting a pair p_i of three-terminal nets into four two-terminal nets. The two three-terminal nets will be referred to as net A and net B. Assume that the middle terminals (with respect to x) of A and B are located on the top or bottom side of rectangle R. The terminals in net A (B) are labeled and ordered as follows: $a_b < a_m < a_e$ ($b_b < b_m < b_e$). The ordering is with respect to their x coordinate values (see section II). Exactly two new terminal points, which are indicated by a dashed line in our figures, are introduced to split nets A and B. These two terminal points are labeled a and b. Each of the two two-terminal nets generated from net A (B) is defined by two terminals in $\{a_b, a_m, a_e, a\}$ ($\{b_b, b_m, b_e, b\}$) joined by a thick line. The number of columns after the transformation increases by one. Without loss of generality, we assume that $a_m \leq b_m$. The connectivity of two new two-terminal nets representing an original three-terminal net is enforced as follows. The terminals of these two nets may be connected by a fixed wire, which is represented by a zig-zag solid line in the figure. If the zig-zag line is not present, the two two-terminal nets generated from a three-terminal net have the property that wires connecting the two new nets in any layout always intersect. At the point they intersect the wires will be made electrically common. There are three cases that need to be considered.

Case 1: a_m and b_m are located on the same side of R.
Assume without loss of generality that a_m and b_m are located on the top side of R. If $x(b_m) \leq x(a_e)$ then let $i = 1$, otherwise let $i = 3$; and if $x(b_m) \leq x(b_e)$ then let $j = 5$, otherwise let $j = 7$. Our procedure applies the transformation given in T_i to net A and the one in T_j to net B. If the horizontal density of the four new nets is 4, then the transformation applied to net A is T_{i+1} and the one for B is T_{j+1}.

Case 2: a_m and b_m are located on opposite sides of R and $x(a_m) \neq x(b_m)$.
The transformation for this case is omitted since it is similar to the one in case 1.

Case 3: a_m and b_m are located on opposite sides of R and $x(a_m) = x(b_m)$.
Depending on the locations of terminals of A and B, one of the transformations T_9 and T_{10} is applied. First, T_9 is applied. If the horizontal density of these four new nets is 4, then T_9 is replaced by T_{10}. Note that a fixed wire (zig-zag line) is introduced in T_{10}.