MULTI-PRIDE : A System for Supporting Multi-Layered Printed Wiring Board Design

Toshimasa Watanabe
Department of Circuits and Systems, Faculty of Engineering, Hiroshima University
4-1, Kagamiyama 1-chome, Higashi-Hiroshima, 739 Japan
Tel:+81-824-24-7662 Fax:+81-824-22-7195 e-mail:watanabe@huis.hiroshima-u.ac.jp

Abstract The purpose of the paper is to outline MULTI-PRIDE, a system for supporting multi-layered printed wiring board design. It consists of (i) circuit bipartition, (ii) placement and routing on each outside layer, (iii) modification of wiring and compaction, and (iv) routing on inside layers.

I. Introduction

We have been developing **MULTI-PRIDE** (MULTI-layered PRInted wiring board DEsign system), a system to support designing multi-layered printed wiring boards [M1]. Here, a multi-layered printed wiring boards means one printed wiring board consisting of n layers for $n \geq 1$ (Fig.1), where if $n \geq 3$ then the first and the last layers are called *outside layers* and any other layer is called an *inside layer*. (In case of $n = 2$, the first layer is called an outside layer, and the last layer is called an inside one.) Elements (ICs, resistors, capacitors, and so on) are placed on one or both of the outside layers and routing is done on outside layers and/or inside layers. Inside layers will be provided if any connection requirement remains unconnected after routing on outside layers: inside layers are increased one by one until all connection requirements are connected.

The designing flow of MULTI-PRIDE consists of (i) circuit bipartition, (ii) placement and routing on each outside layer (often followed by moving some elements to specified positions), (iii) routing on inside layers, (iv) modification of wiring and compaction.

From practical point of view, the following (1)-(4) are required as basic capability of placement and routing methods.

(1) Placing an up-sided element as it should be, where an up-sided element has a specified side to be faced to the board in actual mounting: almost all elements are upside ones;

(2) Given a class of elements, placing each element of the class at the individually specified position of a board;

(3) Maintaining (or controlling) relative relations among elements, such as adjacency, isolation, or even distances;

(4) Reducing the number of nonplanar connections (wires that cannot be embedded without crossing) as well as preserving high routability among terminals of elements.

MULTI-PRIDE currently satisfies (1) through (4), and various capabilities are being added and refined. Handling timing and crosstalk is under investigation.

II. Hardware and Software Requirements

The main system has been implemented in C programming language and runs on a workstation (or even a personal computer) with SunOS 5.5 or UNIX FreeBSD 2.1, by utilizing X Window System(Release 6). Also used is a subsystem consisting of a ring network of ten transputers (INMOS/T-800 with implementation by Parallel C) for parallel processing. The role of this subsystem is being transferred to a network of workstations. MINOS (Ver.5.3), a library of a linear programming (in FORTRAN) developed by Stanford University, CA, U.S.A., is utilized.

III. Outline of MULTI-PRIDE

We briefly explain each step in the design flow of MULTI-PRIDE.

A. Physical conditions

(1) There are three kinds of elements; *linear* elements (Fig.2), *up-sided* elements (Fig.3(a)), each having a specified side which has to be faced to the board in actual mounting, and *free* elements (Fig.3(b)) that have no such constraints. Usually a free element with just two terminals is represented as a linear one.

(2) Each element is to be placed on an outside layer of a board, and each up-sided element must be placed as it should be.

(3) Wires between different layers pass through vias. Wires on outside layers are called *outside wires* and those on inside layers are called *inside wires*.

(4) Any two wires can cross each other only at specified terminals.

(5) Routing through *element-areas* (an area on the board to be occupied by an element) is prohibited. (We add this condition for simplicity of discussion, since this problem itself contains an NP-complete problem [R1], making the discussion too complicated. This restriction may be removed by incorporating post-processing.)

(6) Each of an element, a terminal and an area for wire passing is represented by a rectangle in placement, where overlapping of any two rectangles is prohibited.

ASP-DAC '97
0-89791-851-7$5.00

B. Graph models

We use two kinds of graph models called a terminal-vertex graph and an element-vertex graph. Suppose that a circuit is given by a set of net lists (Fig.4). A maximal set of terminals requiring electrical connection among them is called a *net*, where distinct nets should have no electrical contact.

B.1. Terminal-vertex graph $G_T = (V_T, E_T)$

If we are given a circuit (as in Fig.4) then we represent it as a graph G_T (as in Fig.5). G_T is defined as follows. Represent each two-terminal element by an edge, and other elements as those shown in Fig.2 for linear elements or those in Fig.3(a) for up-sided ones or those in (b) for free ones. These vertices are called *terminal vertices*, and we call these edges *non-removable edges*, which cannot be removed in planar subgraph extraction. Each two-terminal net is represented as a simple edge, and each of other multi-terminal nets is represented by a star (a tree defined by a new vertex, called the *net vertex*, and the edges connecting the net vertex and each of its terminal vertices).

B.2. Element-vertex graph $G_L = (V_L, E_L)$

G_L is obtained by shrinking each of linear elements, two-terminal ones, free or up-sided ones of G_T into individual vertex (called an *element vertex*).

C. Circuit bipartition

Actual bipartitioning is done concerning graph models of circuits. See [CP1]–[CP6] for partition of a graph. We bipartition such a graph model into two subgraphs by means of the UW method [CP7] that tries to minimize both the number of vias and the difference between the sizes of the two outside layers. The graph G_T of Fig.5 is bipartitioned into those of Fig.6. Then layout of each subgraph is designed on an outside layer. The UW method improves any initial bipartition by moving vertices from one group to another, followed by planar subgraph extraction for checking the number of nonplanar edges and estimating the difference between the sizes of outside layers. An initial bipartition is obtained by using the FM method [CP1] or the the WHB method [CP3] with new measures for partition [CP7] incorporated. These new measures are provided in order to reflect not only element sizes but also the number of terminals and/or wiring areas required.

D. Placement and routing on each outside layer

Layout of each partitioned circuit is constructed on an outside layer, mainly based on rectangular duals.

D.1. Extracting a planar spanning subgraph

We partition edges of a graph model into planar edges and nonplanar ones (corresponding to noplanar connections in layout design). This is done by extracting a spanning planar subgraph (Fig.7). Each of many elements has a specified side to be faced to the board in actual mounting. Such a condition has to be preserved when a spanning planar subgraph is going to be extracted from a given graph model. There are some known algorithms that extract a spanning planar subgraph with maximal or almost maximal number of edge. See [PL1] for planarity testing and embedding, and [PL2]–[PL6] for planar subgraph extraction. Unfortunately they are unlikely to be useful in practical design process due to lack of capability of handling physical conditions specific to design of printed wiring boards. Therefore the algorithm PLAN-PWB or PLAN-MW is used. PLAN-PWB, proposed in [PL7], is one of vertex addition algorithms and uses PQR-trees [PL8] for finding a planar spanning subgraph of a given G_T and for handling such physical conditions. PLAM-MW is based on the path addition algorithm and is proposed in [MP1, PL8] for the similar purpose with capability of placing some elements to specified locations of a board. Further reduction of nonplanar connections is being investigated [PL9].

D.2. Rectangular Duals

A *properly triangulated planar (PTP) graph* [RD1] is a connected planar graph satisfying P1-P3:

P1: Every face (except the exterior) is a triangle.

P2: All internal vertices have degree ≥ 4.

P3: All cycles that are not contours of faces have length ≥ 4.

We can assume that the PTP graph is biconnected. (If the PTP graph is not biconnected then, by adding some edges, we can make it biconnected.)(see Fig.8)

A *rectangular dual* [RD3, RD6] is a dissection of a rectangle into several subrectangles, representing a geometric dual of a PTP graph , where each subrectangle corresponds to a vertex of this PTP graph, and two subrectangles share a boundary if and only if the corresponding two vertices are adjacent in the graph (Fig. 9). It has been widely used in placement algorithms for VLSI design [RD4, RD5, RD8]. See [RD1, RD2, RD7] for finding rectangular duals.

Since the two subrectangles, which correspond to the two vertices of an edge, share a part of boundary, if this edge represents connection requirement then a layout, which assures feasible routing required by a given PTP graph, can be obtained. It has capability of controlling placement: making two rectangles adjacent is done by adding an edge between corresponding vertices in G, and conversely placing two subrectangles apart can be realized by creating a path of appropriate length between corresponding two vertices of G.

Rectangles provided for elements and their terminals, are called *element-rectangles* and *terminal-ones*, respectively. Each of length and width of a element-rectangle has to be larger than those of the corresponding element, and a terminal-one must be wider than the maximum width of the corresponding terminal and connecting wire. Similarly we can get space for wires by providing subrectangles (called *wire-rectangles*) that are wide enough for all required connecting wires to pass through. This implies that the global routing is incorporated. This capability of controlling placement is very useful to our purpose. Actual final sizes of subrectangles are determined by repetition of solving a linear programming (LP) with final constraints on rectangle sizes.

Since the length and width of a printed wiring board are usually specified, this application requires capability of computing or estimating sizes of subrectangles so that the difference between the specified board size and those of the whole rectangle to be constructed may be minimized, under the condition that those of each subrectangle are no less than given lower bounds. This is a quadratic programming problem, for which several optimization techniques are existing [QP1].

Existing algorithms for optimum solutions become very slow as the number of vertices or edges increases. On the other hand, there are a number of choices on which the size of the resulting whole rectangle heavily depends, such as selecting four corners, deciding which side has the lower bound on length (or width), and so on. Furthermore, incorporating routing area may cause increase in the size of the whole rectangle. Consequently we adopt a heuristic algorithm proposed in [MP3, RD9] for this problem, producing sharp approximate solutions very quickly.

D.3. Placement and routing on each outside layer

Actual elements can be put into corresponding element-rectangles determined so far, where we try to mount them so that routing among the actual terminals and corresponding terminal-rectangles can be done as much as possible inside each element-rectangle.

Routing is separated into two stages: interconnection among terminal-rectangles of distinct elements, and connection inside each element-rectangle. The former is easy because wire-rectangles are provided and they are placed adjacent to terminal-rectangles to be connected. On the other hand, the latter may require increasing lower bounds: if any mounting of an actual element into the corresponding element-rectangle fails to obtain a desired routing then this element-rectangle has to be enlarged (Fig.10).

Routability testing inside any element-rectangle is done as follows. Suppose that the terminals are $i = 1, \cdots, n$. The subrectangle inside each element-rectangle, in which the center of the element can be placed, is partitioned into several subrectangles. We assign one transputer or a workstation to each subrectangle so that it may try to find n routes, one for each pair of the terminal i and its terminal-rectangle, by means of exhaustive search for each case where the center of the element is placed at any one grid point inside this subrectangle. In each trial for any one pair, we keep finding routes even if some route may overlap with any other one so that the minimum number of overlapping of those routes for this pair can be computed. Reducing computation time by taking advantages of noncrossing paths is being investigated [T1].

E. Layout Improvement

The assured routability of connection requirements among terminals is a remarkable merit of routing by using rectangular-dualization on outside layers, while it may result in placing some elements apart from the desired locations to be actually mounted and/or in roundabout routing(Fig.11). We use rubber-band equivalent [RB1, RB2, RB3] (or visibility graphs, to be exact) to improve this demerit, and this modification also shortens the total wire length of layout with the original routability preserved. Placement and routing are modified by using a visibility graph whose vertices represent terminals, four corners of elements (that are represented as rectangles) and the four corners of a given board. By using this method, we can obtain a layout with elements placed at specified positions. Subsequent compaction will be done by executing one-dimensional x-coordinate or y-coordinate compaction alternatively (Fig.12). See [MP3] for the details and [CM1, CM2, MP3] for compaction.

F. Routing on Inside Layers

In case of single-layered layout, jumpers are only connection requirements left to be processed. On the other hand, in case of multi-layered layout, there are two kinds of nets that should be processed on inside layers: nonplanar edges (edges deleted in extracting a spanning planar subgraph) and those edges each having a pair of virtual terminals (vertices inserted in circuit partitioning). All of them are two-terminal nets. Hence,we assume that a given routing region is a rectangle having only two-terminal nets inside or on the boundary. As the number of layers and/or the size of a board increase, this routing takes long computation time. Hence a parallel detailed router TRED is developed and is used in order to reduce this computation time [P6, MP2]. See [P1]-[P6] for parallel routers. (References for sequential routers are omitted due to shortage of space.)

We assume that the whole routing area is represented as a two-layered rectangular grid graph. Our router basically consists of the following three phases:

1. Global routing and dividing a given routing region into subregions so that their predictive densities may be almost equal, by executing (a) initial global routing, (b) modifying boundaries, (c) rip-up and rerouting.
2. Locating virtual terminals at the boundaries of subregions.This is the important step for abtaining high routability. In order to prevent resundant crossings, we modify the algorithm of [V1] so that we can obtain a desired placing order of virtual terminals at boundaries.
3. Detailed routing inside each subregions consist of (d) initial detailed routing, and (e) rerouting.

The main advantage of this approach is highly parallelized processing even if only small number of processors are available. As a drawback, however, routability may be reduced because routing heavily depends upon locations of virtual terminals. So we incorporate various improvements in order to attain high routability. Currently the proposed routing method is realized as a ring-network of ten transputers (INMOS/T800), one of which is exclusively used for overall control. Improving routability through constrained or unconstrained via minimization problem is being investigated. See [R2] for the details.

Acknowledgements

The research was and is partly supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan: under (C)07680361, (C)08680371 and (A)07308028.

References

[CP1] Fiduccia, C. M. and Mattheyses, R. M., "A Linear-Time Heuristic for Improving Network Partitions", Proc. 19th DAC, pp. 175–181 (1982).

[CP2] Kahng, A. B., "Fast hypergraph partition", Proc. 26th DAC, pp. 762–766 (1989).

[CP3] Kamidoi, Y., Wakabayashi, S., Miyao, J., and Yoshida, N., "A First Heuristic Algorithm for Hypergraph Bisection", Proc.1991 IEEE ISCAS , pp. 1160–1163 (1991).

[CP4] Kernighan,B.W., Lin,S., "An Efficient Heuristic Procedure for Partitioning Graphs", Bell Sys. Tech. J., No. 49, pp. 291–307 (1970).

[CP5] Yen-Chuen Wei, and Chung-Kuan Cheng , "Ratio Cut Partitioning for Hierarchical Designs", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. 10, pp. 911–921 (1991).

[CP6] Yei Ching-Wei and Cheng Chung-Kuan, "A General Purpose Multipule Way Partitioning Algorithm", Proc. 28th DAC, pp. 421–426 (1991).

[CP7] Une,Y., Mizuguchi,Y., Watanabe,T., "A Circuit-Bipartitioning for Multi-Layered Printed Wiring Boards Design", SIG Notes of IPS of Japan, 94-DA-70 pp. 47–54.(Feb. 1994).

[CM1] Yoshimura, T. "A Consideration for LSI Automatic Compaction", Tech. Rep. of IEICE of Japan, CAS84-118, pp. 21–28 (1984).

[CM2] R. Okuda, T. Sato, H. Onodera and K. Tamaru. "An Efficient Algorithm for Layout Compaction Problem with Symmetry Constraints", Proc. ICCAD'89, pp. 148–151 (1989).

[M1] Yasui, T., Tōyama, N., Une, Y., Watanabe, T. and Onaga, K., "MULTI-PRIDE: A Support System for Designing Multi-Layered Printed Wiring Board Layouts of Analog Circuits", Proc. DA Symposium'92, IPS of Japan, pp. 137–140 (1992). (in Japanese).

[MP1] Mizuguchi, Y., "A Study on Placing Elements at Specified Positions in Layout Design of Printed Wiring Boards", Master's thesis of Information Engineering Major, Graduate School of Hiroshima Univ.(1995).

[MP2] Goda, M., "A Study on Routing based on Area Partitioning ", Master's thesis of Information Engineering Major, Graduate School of Hiroshima Univ.(1995).

[MP3] Matsumoto, T., "A Study on Drawing Rectangular Duals within Prescribed Size", Master's thesis of Information Engineering Major, Graduate School of Hiroshima Univ.(1996).

[P1] Spruth, H., Johannes, F., Antreich, K., "PHIroute: A Parallel Hierarchical Sea-of-Gates Router", Proc. 1994 IEEE ISCAS, pp. 487–490 (1994).

[P2] Yamauchi, T., Nakata, T., Koike, N., Ishizuka, A., and Nishiguchi, N., "PROTON: A Parallel Detailed Router on an MIND Parallel Machine", Proc. ICCAD'91, pp. 340–343 (1991).

[P3] Keshk, H., Mori, S., Nakashima, H., Tomita, S., "A Parallel Automated Wire Router Amon", Tech. Rep. of IEICE of Japan, CPSY94-43, pp. 17–24 (1994).

[P4] Shimamoto, T., Shirakawa, I., Hane, H., Yui, N., and Nishiguchi, N., "A Distributed System for Multilayer SOG", IEICE Trans. Fundamentals, Vol. E76-A, No.3, pp. 370–376 (1993).

[P5] Olukotun, O. A., and Mudge, T. N., "A Preliminary Investigation into Parallel Routing on a Hypercube Computer", Proc. 24th DAC, pp. 814–820 (1987).

[P6] Goda, M.,Tyama, N.,Watanabe, T., "A Parallel Detailed Router TRED", TRANSPUTER/OCCAM JAPAN 6 (S.Noguchi, M.Ishizuka and M.Ota(Eds.)), IOS Press, Am sterdam, pp. 156–169 (1994).

[PL1] Booth, K. S. and Lueker, G. S., "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms", J. Comput. & Syst. Sci., 13, pp. 335–379 (1976).

[PL2] Chiba, N., Nishizeki, T., Abe, S. and Ozawa, T., "Embedding Planar Graphs Using PQ-Tree Algorithms", Trans. IEICE Vol. J67-A No. 2, pp. 87–94 (1984). (in Japanese).

[PL3] Jayakumar, R., Thulasiraman, K., and Swamy, M.N.S., "O(n2) Algorithms for Graph Planarization", IEEE Trans. Computer-Aided Design, Vol. 8, No. 3 (1989).

[PL4] Masuda, S., Kashiwabara, T. and Fujisawa, T., "A Wiring Problem on Single Layer Printed Circuit Board without Mounting Modules Upside-Down", IEICE Trans. Vol. J66-A No. 3, pp. 235–242 (1983). (in Japanese).

[PL5] Matsumoto, A., Yamaguchi, K., Kashiwabara, T., Masuda, S., and Taki, M., "A Planarization Algorithm in the Layout Design of Simgle Layer Printed Circuit Board", Tech. Rep. of IEICE of Japan, COMP91-21, pp. 79–88.(1991-05). (in Japanese).

[PL6] Ozawa, T. and Takahashi, H., "A graph-planarization algorithm and its application to random graphs", in Graph Theory and Algorithms, Lecture Notes in Computer Science, Vol. 108, Springer-Verlag, pp. 95–107 (1981).

[PL7] Iwamoto, K., Watanabe, T., Araki, T., and Onaga, K., "Finding Jumpers in Printed Wiring Board Design for Analog Circuits", Proc. 1991 IEEE ISCAS, pp. 2854–2857 (1991).

[PL8] Mizuguchi, Y., Watanabe, T., "Application of the Path-Addition Planarity Testing Algorithm to Layout Design of Printed Wiring Boards", Tech. Rep. of IEICE of Japan COMP94-87, pp. 103–112 (1995).

[PL9] Mizuno, K., Watanabe, T., "A Method of Reducing Nonplanar Connections in Layout Design of Printed Wiring Boards", SIG Notes of IPS of Japan, 96-DA (Feb. 1997), to appear.

[QP1] Ibaraki, T. and Fukushima, M. , "Saitekika Puroguramingu (Optimizetion Programming)", Iwanami Shoten, Tokyo (1991). (in Japanese).

[R1] Masuda, S., Kashiwabara, T. and Fujisawa, T., "On Single Layer Routing under the Condition that Routing is Allowed in the Are a Occupied by the Module", Tech. Rep. of IEICE of Japan , CAS82-114, of Japan pp. 35–40.(1982). (in Japanese).

[R2] Takahashi, K. and Watanabe, T., "Constrained Via Minimization for Three Layer Routing Problems", SIG Notes of IPS of Japan, 96-DA (Feb. 1997), to appear.

[RB1] Dai, W. W-M., Kong,R. and Jue, J., "Rubber Band Routing and Dynamic Data Representation", Proc. ICCAD'90, pp. 52–55 (1990).

[RB2] Dai. W. W-M., Kong, R. and Sato. M., "Routability of a Rubber-Band Sketch", Proc. 28th DAC , pp. 45–48 (1991).

[RB3] Leiserson, C. E. and Maley, F. M., "Algorithms for Routing and Testing Routability of Planar VLSI Layouts", Proc. 17th STOC, pp. 69–78 (1985).
[RD1] Bhasker, J. and Sahni, S., "A linear algorithm to find a rectangular dual of a planar triangulated graph", Technical Report TR 85-26, Computer Science Dept., University of Minnesota, Minneapolis, 1985. Also appeared in Proc. 23th DAC, pp. 108–114 (1986).
[RD2] Xin. He, "On finding the rectangular duals of planar triangulated graphs", SIAM J. Comput.,22(6),pp. 1218–1226 (1993).
[RD3] Kozminski K. and E.Kinnen, "Rectangular duals of planar graphs", Networks, Vol.15, pp. 145–157 (1985).
[RD4] Tani, K., Tsukiyama, S., Shirakawa, I. "An Algorithm to Enumerate All Rectangular Dual Graphs", Tech. Rep. of IEICE of Japan, CAS86-178, pp. 31–38 (1986). (in Japanese).
[RD5] Matsumoto, T. and Watanabe, T., "Drawing Rectangular Duals within a Rectangle of Prescribed Sizes", Tech. Rep. of IEICE of Japan, CAS95-101, pp. 9–16 (1996). (in Japanese).
[RD6] Kzominski, K. A. and Edwin, K., "Rectangular Dualization and Rectangular Dissections" , IEEE Transactions on Circuits and Systems, Vol. 15, pp. 1401–1416 (1988).
[RD7] Tang. H. and Chen, W., "Generation of Rectangular Duals of a Planar Triangulated Graph by Elementary Transformations", Proc. 1990 IEEE ISCAS pp. 2857–2860 (1990).
[RD8] Tani, K., Tukiyama, S., Maruyama, S., Shirakawa, I. and Ariyoshi, H., "Area-Efficient Drawings of Rectangular Duals for VLSI Floor-Plan", Proc. 1988 IEEE ISCAS, pp. 1545–1548 (1988).
[RD9] Matsumoto, T., Mizuno, K., Watanabe, T., "Drawing a Rectangular dual to Meet Prescribed Constraints", SIG Notes of IPS of Japan, 96–AL (Jan. 1997), to appear.
[T1] Abe, K., Watanabe, T., "Estimating Board Size by means of Noncrossing Paths", in preparation.
[V1] Patrick Geoeneveld, "On Global Wire Ordering for Macro-Cell Routing", Proc. 26th DAC, pp. 155–160 (1989).

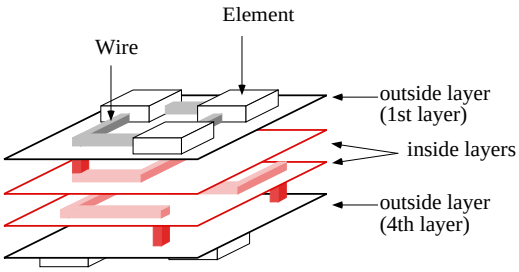


Fig. 1. A schematic explanation of layout design of an n-layered printed wiring board with n=4.

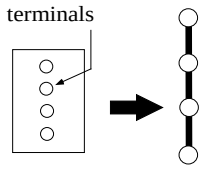


Fig. 2. Representation of a linear element.

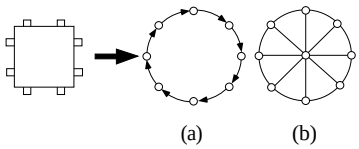
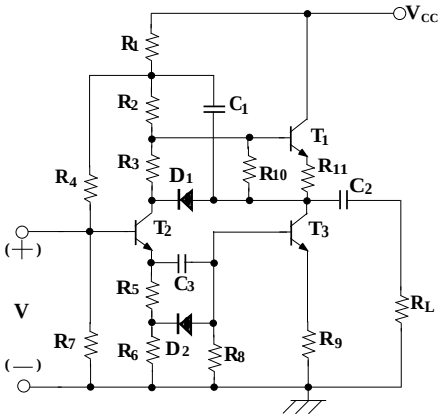


Fig. 3. Representation of elements: (a)an up-sided element as a clockwise directed cycle; (b)a free element as a wheel.




```
NET_LIST
R1  2 R      R2  2 R      R3  2 R      R4  2 R
    1 2          2 3          3 4          2 7
R5  2 R      R6  2 R      R7  2 R      R8  2 R
    9 11         11 13        7 13         10 13
R9  2 R      R10 2 R      R11 2 R      RL  2 R
    12 13        3 5          8 5          6 13
C1  2 C      C2  2 C      C3  2 C      D1  2 D
    2 5          5 6          9 10         4 5
D2  2 D      T1  3 T      T2  3 T      T3  3 T
    11 10        1 8 3        4 9 7        5 12 10
VCC 1 V      V   2 V
    1            7 13
```

Fig. 4. An example of a circuit and the set of net lists

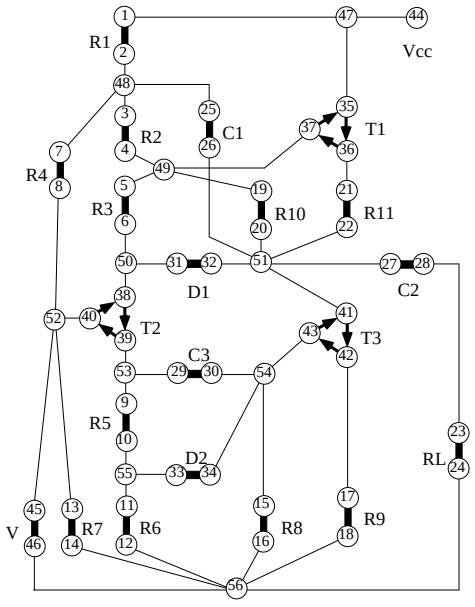


Fig. 5. The graph $G_T = (V_T, E_T)$ for the circuit in Fig.4

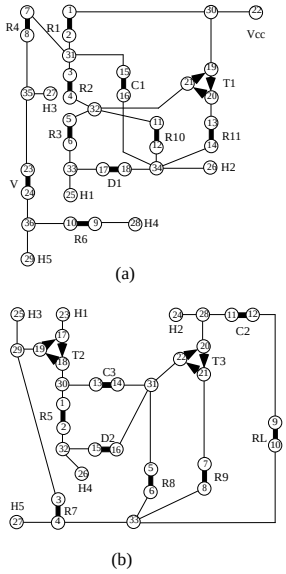


Fig. 6. The terminal-vertex graphs constructed by bipartitioning the graph of Fig.5.

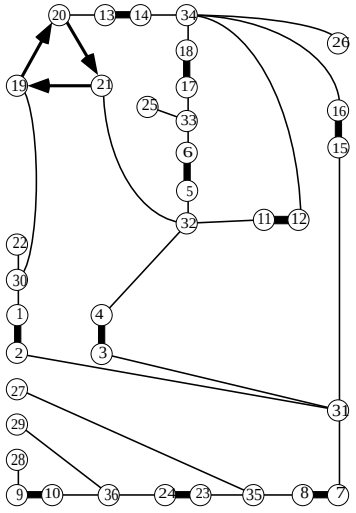


Fig. 7. A planar graph extracted from the graph in Fig.6(a).

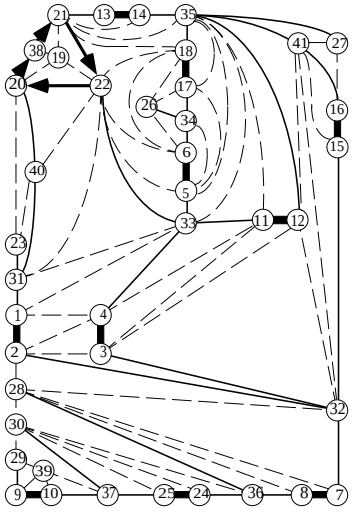


Fig. 8. A PTP graph of the planar graph in Fig.7.

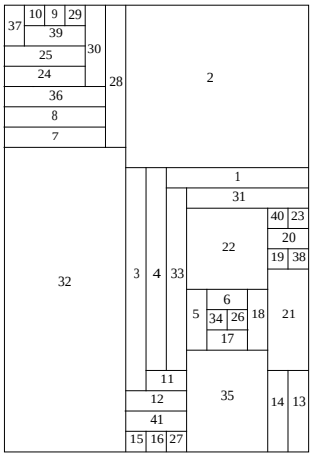


Fig. 9. A rectangular dual of the PTP graph in Fig.8.

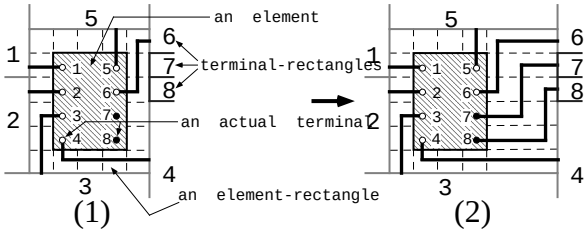


Fig. 10. An example requiring enlarging an element-rectangle (1)Connecting actual terminals 7, 8 to their terminal-rectangles is unsuccessful; (2)Enlarging the rectangle enables routing that satisfies all connection requirements

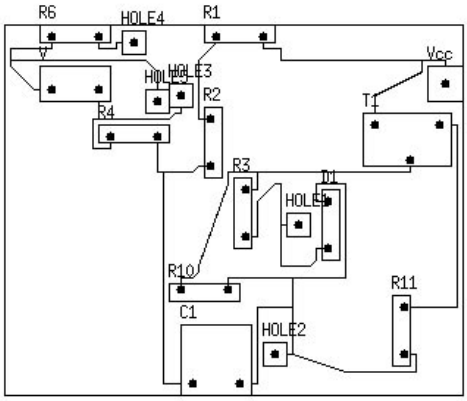


Fig. 11. An initial layout obtained for the circuit of Fig.6(a)

.

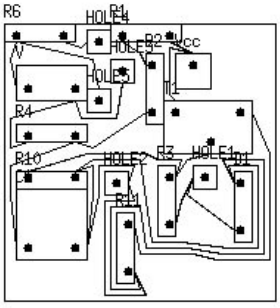


Fig. 12. A final layout obtained from that of Fig.11