

Consiglio Nazionale delle Ricerche  
ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE  
PISA

Pubblicazione n. A 73-20

---

G. ALIA - G. FROSINI - P. MAESTRINI

# A PRINTED BOARD DESIGN ALGORITHM FOR IMPLEMENTATION ON SMALL COMPUTERS

---

Estratto da: 8th Yugoslav International Symposium on Information Processing (Bled, 1973), d/9 1-7

# A PRINTED BOARD DESIGN ALGORITHM FOR IMPLEMENTATION ON SMALL COMPUTERS

G. Alia, G. Frosini, P. Maestrini

Istituto di Elaborazione dell'Informazione  
Consiglio Nazionale delle Ricerche  
Pisa, Italy

The paper deals with the automation of the placement of integrated modules on a board and of the interconnection of the module terminals. The procedures proposed for solving such problems require very limited resources for implementation and are relatively fast, as compared to the approaches discussed in the literature. The reduced complexity originates from the assumption of a structured scheme and the decomposition of the placement problem and of the interconnection problem in simpler cascaded subproblems.

## INTRODUCTION

The problem of automating the realization of a logical circuit consists of four subproblems [1]: covering the given circuit by means of a set of integrated modules; partitioning the modules in subsets (each one corresponding to a board); placing the modules of a subset in proper positions on the board; interconnecting the terminals of the modules on the board.

Automated procedures for solving the two last problems, well suited for implementation on a small computer, are described in this paper.

The aim of the placement problem is to minimize some objective, such as the total conductor path length. As the connections among the modules are not determined when the module placement is carried out, this goal cannot be directly reached; the most algorithms proposed in the literature attempt to minimize some quantity indirectly related to the interconnection length [2,3].

The interconnection problem can be, in turn, divided in two major subproblems: *wire list determination*, that consists in determining how a set of electrically common module ter-

minals should be interconnected; *wire layout problem*, that consists in specifying the conductor path in every wire list. Several procedures for solving single steps of the interconnection problem have been proposed in the literature [4-10]. Specifically, the wire layout problem solutions consist of either a heuristic procedure and or Lee's algorithm [8]. In this algorithm the board is divided into small squares, whose size is comparable to the width of a conductor path. Akers [9] has shown that at least five different symbols are required, in the computation, for characterizing the situation of every square on the board. This technique requires a large memory space for implementation and is incongruous with our aim.

For obtaining an algorithm requiring a limited memory space, a solution is proposed in which the conductor paths are printed according to a fixed connection scheme. Moreover, for reducing the total computation time, the placement problem and the interconnection problem have been decomposed in several simpler subproblems, solved separately. This decomposition is made possible

by the features of the proposed structured scheme.

The procedures we propose attempt to minimize the total area covered by the conductor paths. In every step, an objective related to this quantity is directly minimized.

#### INTERCONNECTION PROBLEM

In the proposed scheme the connections among module terminals are realized by means of horizontal and vertical segments, printed on different sides of the board. An horizontal and a vertical segment electrically common are connected via a hole. The modules are placed in the board on the nodes of an array ( Fig. 1); the exact position of each row and each column will be stated only when the interconnection problem will be completely solved.

A connection among electrically common terminals of modules belonging to the same column is realized by means of a vertical segment, printed in the vertical strip on the right side of the considered column, connecting the horizontal prolongations of the proper terminals (Fig. 2).

A connection among electrically common terminals of modules belonging to different columns is realized by vertical segments connecting module terminals in a same column, as stated in the preceding paragraph, and by a horizontal segment connecting the vertical segments previously obtained and printed in a horizontal strip whose position is to be determined (Fig. 3) .

Some vertical segments can be prolonged to the connector (up to the top side of the board ), for connecting modules belonging to different boards.

By using such a scheme, the wire list problem reduces to determine the horizontal strip where each horizontal segment must be printed. Moreover, the wire layout problem reduces to determine the exact position of a segment in the corresponding strip, i. e. the horizontal (vertical ) line where each horizontal (vertical ) segment must be printed.

Note that in this connection scheme the interconnection problem has always a solution, no matter how the modules are interconnected, while the procedures using the Lee's algorithm are not always able to route all the conductor paths.

#### Wire List Determination

For interconnecting terminals of modules belonging to different columns, a single horizontal segment is used. The position of the horizontal strip selected for printing such a segment determines the length of the conductor path. Each horizontal segment is assigned to the horizontal strip for which the connection length results minimum.

#### Wire Layout Problem

This problem has been solved in two separate steps: a) assignment of the horizontal segments to the horizontal lines in every horizontal strip; b) assignment of the vertical segments to the vertical lines in every vertical strip.

In performing the first step, the actual position of the extremes of the horizontal segments is unknown, as the position of the vertical segments has not yet be stated. It is assumed that in a horizontal strip two horizontal segments connected to a pair of vertical segments belonging to the same vertical strip can always be printed in the same horizontal line (*compatibility hypothesis* ). For every horizontal strip, the number of horizontal lines necessary for printing the horizontal segments is evaluated, as the maximum number of horizontal segments crossing a column in the considered strip. Every horizontal segment is assigned to a line in the strip, by taking into account two objectives: 1) assigning to different lines two horizontal segments connected to a pair of vertical segments belonging to the same vertical strip, whenever the number of the horizontal lines required does not exceed the maximum previously evaluated; 2) assigning the horizontal segments to the lines in a strip in an ordered way, according to the position of the *centre of gravity* of the extremes of the vertical segments connected to each horizontal segment. The first objective attempts to eliminate the restriction imposed at the relative positions of the vertical segments by the compatibility hypothesis. The second objective attempts to maximize the number of vertical segments that can be printed in the same vertical line.

In the second step, the vertical segments are assigned to the vertical lines with the objective of minimizing the required lines in every vertical strip. This is achieved by filling a line at a time and by assigning each time to the line one of the vertical

segments whose upper bound is the nearest to the lower extreme of the last assigned segment. In performing this assignment algorithm a priority relation must be taken into account between two vertical segments connected to a pair of horizontal segments that have been assigned to the same horizontal line.

#### PLACEMENT PROBLEM

Although the procedure previously described for interconnecting the module terminals leads always to a solution, no matter how the modules are placed on the board, nevertheless the resulting optimization level depends on the actual positions of the modules on the board.

The aim of the procedure described in this section is to find an optimal module position in respect to the application of the interconnection procedure.

The placement problem has been divided into three simpler optimization subproblems: module partitioning into columns; module ordering in every column; column ordering in the board.

#### Module Partitioning

The modules are partitioned in subsets of given cardinality with the final objective of reducing the width of the horizontal and the vertical strips. Note that the width of a vertical strip has a lower bound, determined by the number of interconnections among the modules in the column at the left side of the considered strip and the connector; while does not exist a bound on the width of a horizontal strip. Moreover, in a vertical strip there exist some restrictions on the position of the vertical segments, due to the compatibility hypothesis. As the efficiency of the wire layout algorithm in a given strip increases as the number of segments in the strip increases, it is convenient to concentrate the interconnections in the vertical strips, i.e. to reduce the number of the interconnections among the columns.

In order to reach such objective, first an initial configuration with reduced column interconnections is determined, and then this configuration is improved, thus obtaining an optimal solution in respect to the change of position of any two modules.

The initial configuration is constructed by filling the columns sequentially, through successive steps, each step adding a module

to the current column. The module to be added is selected according to a criterium of "maximum conjunction, minimum disjunction" used in most of similar problems [11]. Precisely, the following figure of merit has been used:

$$f_i = \frac{V_1 + 1}{V_2 + C + 1}$$

where  $V_1$  represents the number of interconnections among the considered module and the modules already assigned to the current column, and  $V_2$  represents the number of connections among the considered module and the modules not yet assigned. A quantity  $C$  has been added to the denominator in order to distribute the connections to the connector with a certain degree of regularity among the vertical strips, and then to increase the efficiency of the algorithm for reducing the vertical strip width. The quantity  $C$  is defined "0" if the number of the connections among the connector and the modules assigned to the current column, including the considered module, is less than the average  $a$  of the number of connections between a column and the connector; otherwise  $C$  is defined equal to  $a$ . The addition of "1" to the denominator prevents the division by zero, and to the numerator draws a distinction between modules for which  $V_1$  is equal to zero.

The module configuration thus obtained is improved by means of a "simple replacement algorithm" [12]. At a given step the module  $M'$  is selected, for which is maximum the difference  $d$  between the number of interconnections among this module and the modules belonging to the other columns, and the number of interconnections among this module and the modules belonging to the same column. The replacement of the selected module yielding the maximum decrement in the number  $n$  of the interconnections among the columns is executed. If no replacement of  $M'$  makes  $n$  to decrease, the module not yet considered for which  $d$  is the greatest is selected. The algorithm stops when there exist no module for which  $d$  is positive, as the number of interconnections among the columns cannot decrease by changing the positions of two modules for which  $d$  is negative.

Note that the degree of optimality of the final configuration obtained by applying the simple replacement algorithm (optimality in respect to the number of interconnections

among the columns) depends on the degree of optimality of the initial configuration. For this reason, an initial configuration with a good optimization level has been constructed before applying this algorithm.

#### Module ordering

The modules assigned to every column are ordered from top to bottom with the aim of minimizing the width of each vertical strip. Precisely, in each strip is minimized the number of vertical lines required for printing the vertical segments connecting terminals of modules belonging to the corresponding column only. The portions of vertical segments necessary for connecting terminals of modules belonging to different columns cannot be taken into account, as the position of the horizontal segments is not known at this stage.

At a given step of the module ordering algorithm, it is evaluated the number  $L_t$  of the lines required for printing the vertical segments connecting the modules already ordered, and the number  $L_n$  of these lines that are not utilized in correspondence to the position in which the new module is to be placed. Let us consider a module  $M''$  not yet ordered. The segments connecting  $M''$  to modules yet ordered have been considered in the previous steps, and the number of lines required for printing these segments has been considered in  $L_t$ . Some of these segments (connecting  $M''$  to modules already ordered only) may stop in a terminal  $t_i$  of  $M''$ , thus making allowable an existing line for a new segment (connecting  $M''$  to modules not yet ordered and starting from a terminal of  $M''$  whose position is lower in respect to the position of  $t_i$ ). Then, the segments connecting  $M''$  to modules not yet ordered only can be printed in existing lines (lines not utilized or lines in which some segments stop), or can require new vertical lines. The module is selected for which is minimum the following figure of merit:

$$f_2 = L_e - L_n$$

where  $L_e$  is the number of the existing lines in which can be printed vertical segments connecting  $M''$ , and  $L_n$  is the number of the new lines required for printing vertical segments connecting  $M''$  to modules not yet ordered only.

An initial value must be assigned to  $L_t$  in order to select the module to be assigned to the first position in the considered column

The width  $W$  of the vertical strip is approximately evaluated assuming that: 1) the length of each vertical line is proportional to the number of modules in the column; 2) the length of each vertical segment is proportional to the number of terminals connected by the segment; 3) the vertical segments can be printed consecutively in the lines independently of their position. The initial value of  $L_t$  is assumed to be the maximum quantity between  $W$  and the number  $I$  of connections among the modules belonging to the considered column and the connector. Of course, the initial value of  $L_n$  is the maximum quantity between "0" and  $W-I$ .

#### Column Ordering

The columns are ordered with the objective of minimizing the total length of the horizontal segments. It should be difficult to minimize directly the width of the horizontal strips, as the position of the horizontal segments is not known at this stage, and the position of a column affects the width of all the horizontal strips, that should be considered simultaneously.

First, the column is considered for which is maximum the number of interconnections with the other columns. In every successive step, the column is selected for which is maximum the number of interconnections with the column previously ordered, and the length  $\ell$  of the horizontal segments necessary for the interconnections is evaluated, supposing the considered column to be assigned both on the left side and on the right side, near to the columns yet ordered. The column is assigned to the position for which  $\ell$  is minimum.

#### CONCLUSIVE REMARKS

The procedures described in this paper have been actually implemented on a IBM-360/67 computer, in order to evaluate the optimization level of the result [13-14]. A program for generating pseudo-random circuits has been written, in the hypothesis that the probability for  $m$  terminals to be electrically common is:

$$p(m) = \frac{k}{m^h}$$

where  $k$  and  $h$  are constant, and  $m \geq 2$ . The number  $N$  of modules on the board, the maximum number  $M$  of electrically common module terminals, and the percent  $F$  of the terminals of a module utilized for the connections can be arbitrarily fixed.

In Fig. 4 a printed board is shown, obtained by applying the proposed procedures to a pseudo-random circuit with 37 12-terminal integrated modules, distributed in 7 columns and 6 rows, generated by supposing  $M=10$  and  $F=60\%$ .

From the examined examples, it results that the quality of the automatically designed boards is comparable with the quality of the boards manually obtained.

This first implementation has also made possible an approximate evaluation of the resources and of the computation time required for an implementation on a small computer. For a circuit like that shown in Fig. 4 and a computer like the Hewlett-Packard 2100, we estimate that a memory space of 10 K-words (the word length being 16 bits) and a computation time in the order of a score of minutes be sufficient.

#### REFERENCES

- [1] Breuer, M. A. "Recent developments in the automated design and analysis of a digital system", *Proc. Inst. of Elect. and Elect. Eng.*, Vol. 60, No 1 (January 1972), pp. 12-27.
- [2] Hanan, M and Kurtzberg, J. M. "Force-vector placement techniques" IBM Report RC 2843 (April 1970).
- [3] Hillier, F. S. and Connors, M. M. "Quadratic assignment problem algorithms and the location of indivisible location", *Management Sci.*, Vol. 13, (September 1966) pp. 42-57.
- [4] Lin, S. "Computer solutions of the traveling salesman problem", *Bell System Tech. J.*, Vol. 44 (1965), pp. 2245-2269.
- [5] Loberman, H. and Weinberger, A. "Formal procedures for connecting terminals with a minimum total wire length", *J. ACM*, Vol. 4, (October 1957), pp. 428-437.
- [6] Gilbert, E. N. and Pollark, H. O. "Steiner minimal trees", *Soc. for Ind. and App. Math.*, Vol. 16 (1968) pp. 1-29.
- [7] Hanan, M. "On Steiner's problem with rectilinear distance", *Soc. for Ind. and App. Math. J.*, Vol. 14 (March 1966) pp. 255-265
- [8] Lee, C. Y. "An algorithm for path connections and its applications", *IRE Trans. on Elect. Comput.*, Vol. EC10, (September 1971) pp. 346-365
- [9] Akers, S. B. "A modification of Lee's path connection algorithm", *IEEE Trans. on Elect. Comput.*, Vol. EC16 (February 1967), pp. 97-98.
- [10] Fisk, C. J., Caskey, D. L. and West, L. E., "Accel: automatic circuit card etching layout", *Proc. IEEE*, Vol. 55 (November 1967), pp. 1971-1982.
- [11] Breuer, M. A., *Design automation of digital systems: theory and techniques*, Vol. 1, Prentice-Hall, (1971).
- [12] Nicholson, T. A. J. "Permutation procedure for minimizing the number of crossing in a network", *Proc. IEEE*, Vol. 115 No 1 (January 1968), pp. 21-26.
- [13] Alia, G., Di Giacomo, V., Frosini, G. and Maestrini, P. "Tracciamento automatico di circuiti stampati secondo uno schema predeterminato", *Internal Report B 72-7, Consiglio Nazionale delle Ricerche - Istituto di Elaborazione dell'Informazione-*, (May 1972).
- [14] Alia, G., Frosini, G. and Maestrini, P. "Automated module placement and wire routing according to a structured bipartite planar scheme in printed boards", *Computer Aided Design*, Vol. 5, No. 3 (July 1973), pp. 152-159.

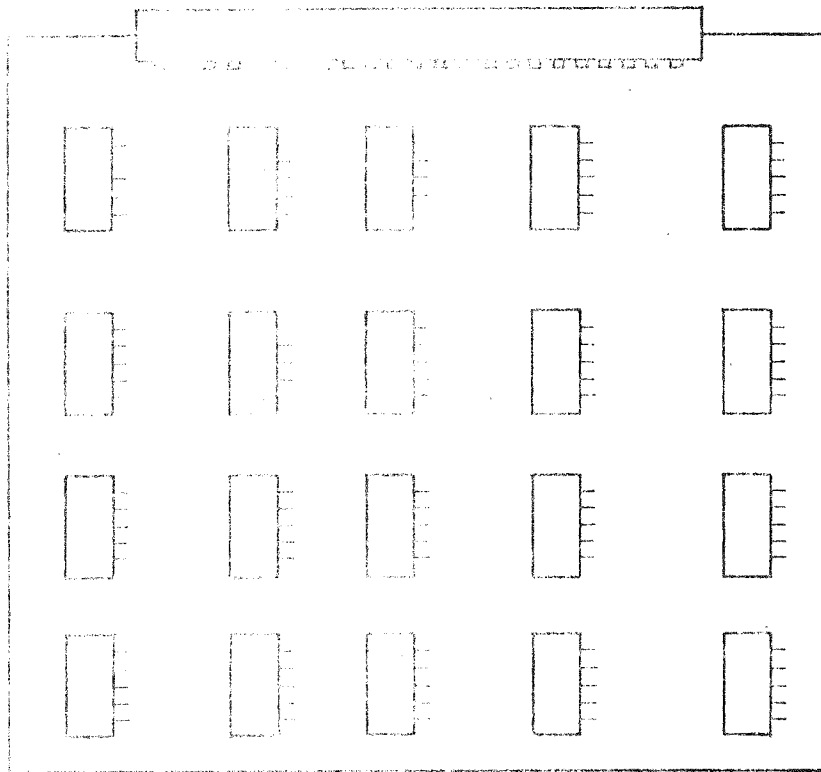


Fig. 1 Module positions on the board

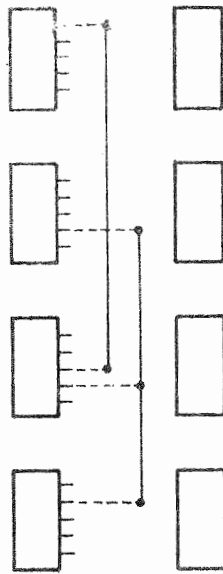


Fig. 2 Connections among terminals of modules belonging to the same column

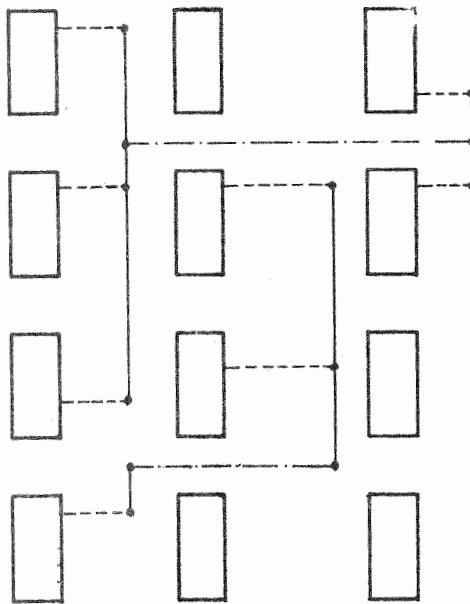


Fig. 3 Connections among terminals of modules belonging to different columns

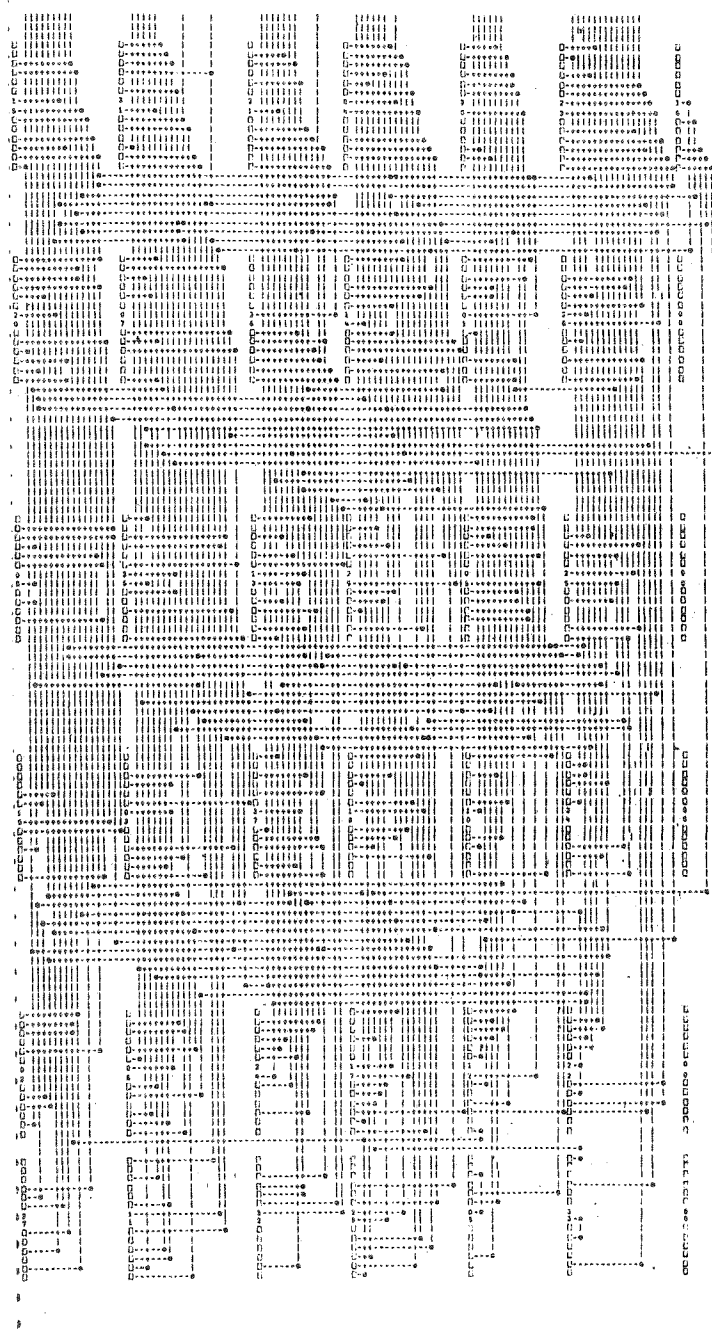


Fig. 4 An automatically designed board