# A Placement Program for Printed Circuit Boards

Susumu GOSHIMA*, Yasuo ISHIBASHI*, Akira OSAWA**
and Isao YASUDA**

A placement program for printed circuit boards is described. The purpose of the program is to determine the optimal placement of circuit components on a board automatically using DA (design automation) techniques. The outputs of this program are available for the routing program to obtain a printed connection path. The unique features of the program are as follows.

(1) The placement is performed by a step-by-step improvement, applying alternate repetition of gate-assignment and IC-placement so that the total length of the connection path is minimized.

(2) Steinberg's method is extended for determining the location of gates.

(3) Practical methods are developed for determining the location of IC's.

In this paper the placement techniques developed in the program, the computer program, and the results of computation are described.

## 1. Introduction

The placement of components consists of two interrelated procedures; that is, assignment of logic gates to IC's (Integrated Circuits), and placement of IC's on a given board. These two procedures are typically performed in sequential order using two kinds of evaluating measures, that is, "Max. conjunction and min. disjunction" for the assignment and "Minimum-length connection path" for the placement.

One of the unique features of the program is that the placement is performed by a step-by-step improvement, applying alternate repetition of gate-assignment and IC-placement.

In order to perform the placement procedure, several methods are employed, that is, Case's method[1)] is employed for determining the initial gate-assignment, and Steinberg's method[2)] is extended for determining the location of gates Also. practical techniques

This paper first appeared in Japanese in Joho—Shori (Journal of the Information Processing Society of Japan), Vol. 17, No. 6 (1976), pp. 466~471.

* Hitachi Research Laboratory, Hitachi Ltd.
** Omika Works, Hitachi Ltd.

for determining the location of IC's are developed.

## 2. Program description

The program is written in FORTRAN IV, and operates on either a HITAC 8000 series or an IBM 370 series computer with more than 400KBs of main memory.

A general flow-chart of the program is illustrated in Fig. 1.

The program is constructed of 7 steps, and has two nesting loops.

The role of the inner nest is to improve the placement through feed back between gate-assignment and IC-placement; and that of the outer nest is to avoid the local optimization of placement.

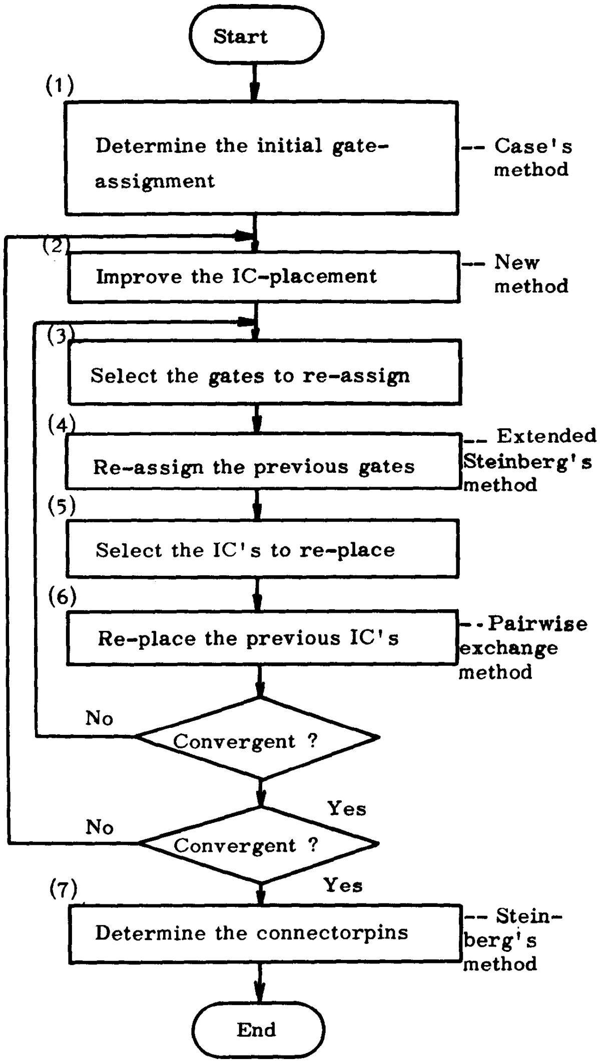
Start
(1)
Determine the initial gate-assignment
-- Case's method
(2)
Improve the IC-placement
-- New method
(3)
Select the gates to re-assign
(4)
Re-assign the previous gates
-- Extended Steinberg's method
(5)
Select the IC's to re-place
(6)
Re-place the previous IC's
-- Pairwise exchange method
No
Convergent ?
Yes
No
Convergent ?
Yes
(7)
Determine the connectorpins
-- Steinberg's method
End

Fig. 1 General Flow Chart of the Program

## 3. Extension of Steinberg's method

In order to apply the assignment problem[3)4)] to placement in electronic circuits, Steinberg has described a method of selecting an unconnected subset U from the component set G, and repeating partial replacement for unconnected set U.

However, in the case of IC-placement, the element ratio of U to G is not so large that the method can obtain efficient placement of IC's.

Improvements of Steinberg's method in this program are as follows.

(1) In order to enlarge the ratio of U to G, Steinberg's method is applied not to IC-placement, but to gate-assignment.

(2) A sub-optimal procedure is developed for partial replacement of connected set E.

The sub-optimal procedure and some examples of its application will be described.

〔 A sub-optimal procedure for replacement 〕

(1) Define an extended sub-set $E=\{e_1, e_2 \cdots e_m\}$, location set $P=\{p_1, p_2, \cdots p_m\}$ and calculate the evaluating matrix 〔 Aij 〕.

(2) Define the initial assignment of E to P.

(3) Calculate the gain ( $\Delta$ ij), which is a measure of goodness for exchange.

$$\Delta ij = Aipi - Aipj + Ajpj - Ajpi \quad (j \leq m) \ldots\ldots\ldots\ldots\ldots (1)$$

$$= Aipi - Aipj \quad (j > m) \ldots\ldots\ldots\ldots\ldots (2)$$

..... if neither $e_i$ nor $e_j$ are connected to the other elements of E.

$\Delta$ij : the difference between the length of the connection path obtained before exchange and that after exchange .......... otherwise.

(4) Calculate the maximum values of $\Delta$ ij for each row (i)

$$bi = \underset{j}{Max} ( \Delta ij) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

(5) If $\underset{i}{Max}$ (bi) < 0, then terminate the calculation; otherwise, sort the values (bi) in descending order and get a set of pairs for exchange.

(6) Exchange the positions of elements in each of the pairs obtained in the previous step and go to (3).

It is known that the connection of elements in a set of E to each other greatly influences elapsed computer time in this procedure but does not influence the accuracy of replacement.

Fig. 2 shows the results of computation for several assignment problems, obtained by the sub-optimal procedure, compared with solutions by Iri's method.

The X-axis denotes the size of the evaluating matrix for the assignment problems, and the Y-axis denotes the solutions and computer times.

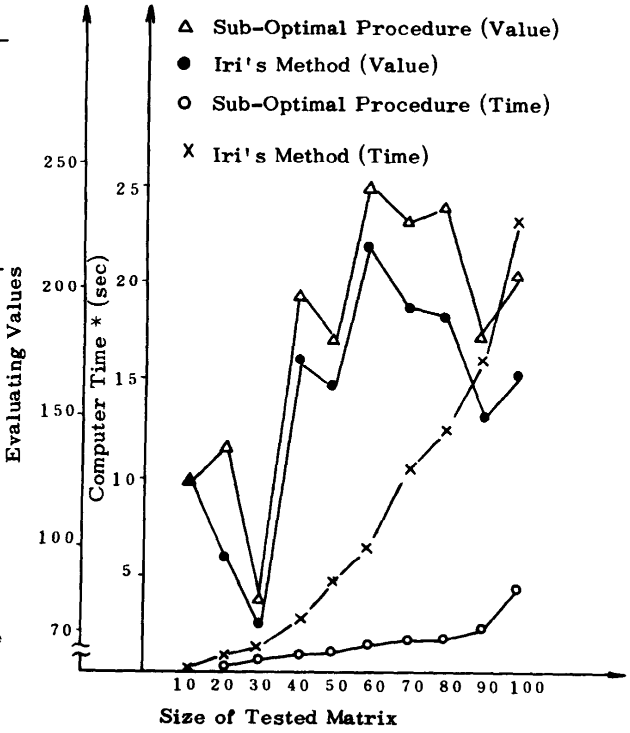

Fig. 2 Results Obtained from The Sub-Optimal Procedure

*) CPU time of HITAC 5020F Computer

The figure shows that the solutions obtained by the sub-optimal procedure approximate closely to the optimal solution of Iri's method, and that 80 percent improvement is obtained in the solutions by random assignment.

## 4. IC-placement method

Two methods are developed for IC-placement. The first method is used for placing mutually connected IC's in neighboring locations, and the second one is used for optimizing the placement of neighboring IC's. In the following, first the methods are described in detail, and second the results of computations obtained with them are shown.

〔The first method〕

(1) Define and calculate the transfer vector Ti (x,y) for each IC numbered i, denoted IC (i).

$$Ti(x, y) = \frac{1}{\sum_j Rij} \left[ \sum_j Rij \cdot Dij \right] \quad \cdots\cdots\cdots \quad (4)$$

where, Rij : electrical affinity between IC (i) and IC (j).

Dij : distance vector composed of X-distance and Y-distance between IC (i) and IC (j).

(2) Define the neighboring IC's, IC (k) and IC ($\ell$)

IC (k) : a neighboring IC (i) in direction X, determined by the x-component of Ti.

IC($\ell$) : a neighboring IC (i) in direction Y, determined by the y-component of Ti.

(3) Calculate the gains Cik (Ci$\ell$) for the exchange of IC (i) and IC (k) (IC ($\ell$)) according to the formulas (5), (6).

$$Cik = Ti(x, y) - Tk(x, y), \quad Ci\ell = Ti(x, y) - T\ell(x, y) \cdots\cdots \quad (5), (6)$$

(4) Repeat steps (2) - (3) for all i.

(5) Sort the values of Cik and Ci$\ell$ in descending order and determine the combination set Q for exchange, which might be effective in improving IC-placement.

$$Q = \{ q_1, q_2, \cdots\cdots, q_n \} \quad \cdots\cdots\cdots \quad (7)$$

(6) Recompute the gain for the exchange $q \in Q$, exchange the pair of elements defined by $q \in Q$, if the gain of $q \in$ is effective for improvement.

〔The second method〕

This method is employed to improve the placement obtained by the first method. In this method, a set of four IC's and a set of four locations are selected. All possible combinations of IC's and locations are examined to find the best possible placement for the four IC's.

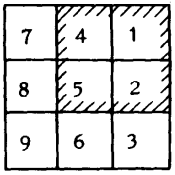

Fig. 3 An Example of Selecting Ic's

A description of this procedure for nine IC's and the locations **are given in Fig. 3.**

(1) Select the following subsets $P_1$ , $P_2$ , $P_3$ , and $P_4$ from the locations set P.

$P = \{1, 2, 3, \ldots\ldots\ldots, 9\}$ , $P_1 = \{1, 2, 4, 5,\}$ , $P_2 = \{2, 3, 5, 6\}$ ,

$P_3 = \{4, 5, 7, 8,\}$ , $P_4 = \{5, 6, 8, 9\}$

(2) Define the IC set $Ci = \{Ci_1, Ci_2, Ci_3, Ci_4\}$ . where, Cij is the IC which corresponds to the j-th element of Pi.

(3) Fixing the locations of all elements of $Ci^*$ (complement set of Ci), calculate the best allocation of Ci to Pi.

(4) Repeat steps (2) ~ (4) for all i= 1, 2, 3, 4.

(5) If the allocation in Ci is not changed in the above step, then terminate the procedure; otherwise go to (1).

〔Results of computation〕

To examine the efficiency of these methods, five examples are tested. Table 1 summarizes the results of computation.

The total length of the connection path in the table illustrates the length of the connection path obtained by the three methods, that is, the new method (a combination of the previous two methods), the random allocation method with 10,000 repetitions and the so called pairwise-exchange-method. Compared with the other methods, the new method shows considerably better placement.

Table 1

Results (Total length of connection path) obtaine from the new algorithm

| Method \ No. of IC | 9 | 12 | 15 | 16 | 20 |
|---|---|---|---|---|---|
| Random allocation | 1884 | 4210 | 4760 | 3130 | 5240 |
| Pairwise exchange | 1916 | 4160 | 4640 | 2890 | 4700 |
| New method | 1892 | 4150 | 4590 | 2930 | 4500 |

## 5. Examples of application

The placement program described above has been applied to various types of printed circuit boards, such as boards with 33 IC's, 55 IC's, 200 IC's, etc..

Five examples are shown in table 2, and compared with manual placement.

The Table shows that a more than 40 percent improvement in the number of unconnected points is achieved over manual placement, and that the computer time in the routing is considerably reduced.

Table 2

Results of applying the program to some printed circuit boards

| | | | | Routing ** | | | |
|---|---|---|---|---|---|---|---|
| Case | No. of gate | No. of IC's | Elapsed computer time in placement * | Manual placement | | Placement by the program | |
| | | | | Unconnected lines | Time* | Unconnected lines | Time* |
| 1 | 106 | 31 | 24 | 34 | 52 | 14 | 37 |
| 2 | 138 | 58 | 54 | 50 | 123 | 38 | 100 |
| 3 | 74 | 25 | 19 | 31 | 52 | 4 | 20 |
| 4 | 65 | 25 | 14 | 11 | 32 | 8 | 27 |
| 5 | 57 | 26 | 26 | 28 | 50 | 23 | 42 |

* CPU time (min.) of HITAC 8400 computer.

** Obtained under the routing conditions ie.
2 layers, one channel, floating through-hall.

## 6. Conclusion

An efficient placement program for IC's on printed circuit boards has been presented. The program can be applied to various types of printed circuit boards and is expected to reduce the time that is required for the design of the boards.

Examples of the application of the program have proved that placement of IC's with this program is considerably better than manual placement in the area of routing the printed path.

## REFERENCES

1) P.W. Case,et al.: Solid logic design automation for IBM system/360, IBM J. Res. & Dev., vol. 8, pp. 127-140 (1964)

2) L. Steinberg : The backboard wiring problem. A placement algorithm, STAM Rev., vol. 3, pp. 37-50 (1961)

3) H.W. Kuhn : Hungarian method for the assignment problem, Naval Res. Logist. Quart., 2 (1955), pp. 83-97.

4) M. Iri : A new method for the transportation network problem, KEIEI KAGAKU, vol. 3, N. 4, pp. 190-206 (1960).