

Copyright © 1977, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

INTERLIBRARY LOAN DEPARTMENT
(PHOTODUPLICATION SECTION)
THE GENERAL LIBRARY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720

AN APPROACH TO THE TWO-DIMENSIONAL
PLACEMENT PROBLEM IN CIRCUIT LAYOUT

by

S. Goto and E. S. Kuh

Memorandum No. UCB/ERL M77/16

14 March 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

An Approach to the Two-dimensional
Placement Problem in Circuit Layout

S. Goto* and E. S. Kuh

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

In this paper a new approach to the placement problem is introduced. The main idea is to take advantage of what one can do in linear placement in tackling the two-dimensional placement problem. The method consists of three distinct phases, namely: decomposition, linear placement and iterative improvement. Each is clearly spelled out. Both constructive and iterative algorithms are developed. The complexity of computation is analyzed and the method has been tried with practical examples. Although no general conclusion can be made on the effectiveness of the method, it appears that the method is at least comparable to that described in a recent paper [2].

Research sponsored by the National Science Foundation Grant ENG74-06651-A01 and the Joint Services Electronics Program Contract F44620-76-C-0100.

*Central Research Laboratories, Nippon Electric Co., Ltd., Kawasaki, Japan.

I. INTRODUCTION

In the design of large electronic circuits or systems, one key problem which designers encounter is "layout". Because of its complexity, the layout problem is usually divided into two subproblems, namely: placement and routing. By placement, we mean the selection of locations, for example, on a board for individual modules in such a way to facilitate routing. The modules represent components or subsystems, which in many typical cases, are semiconductor chips with external leads called pins to be connected according to specifications. By routing, we mean the process of making the interconnections while satisfying various physical and economical constraints. Often, there exist conflicting requirements and goals in each subproblem, and there are possible trade-offs; therefore it is difficult to think about what constitutes an optimum layout design. In this paper we will deal with only the placement problem, and we shall adopt the usual goal of minimizing the total routing length defined in an appropriate way.

Our approach differs from those in the literature [1-4] in that throughout the process we bear in mind and depend on what can be done in the much simpler one-dimensional placement problem. Our approach can be separated into three phases, each one is to achieve a specific aim which is easily understood. Wherever possible, we shall comment on the theoretical implication of each problem. Both constructive and iterative algorithms will be developed. The complexity of computation of each proposed algorithm is analyzed and is found to depend on the cubic power of the number of modules.

Phase one deals with the decomposition of all modules into horizontal blocks and vertical blocks. The aim is to minimize the number of

interconnections between vertical blocks and between horizontal blocks, thus more or less reducing the problem to a linear placement problem. Two methods are presented, one constructive and one iterative. From the examples which we have tried, the iterative method can follow the constructive method to yield the best results. At the conclusion of this phase, each module must belong to one horizontal block and one vertical block. The second phase deals with the determination of an optimum ordering of both the horizontal blocks and the vertical blocks. The purpose of this is to minimize the routing length. Here we depend on a recently worked out algorithm of linear placement [5]. Finally, in phase three, we perform the mop-up process to improve still the result. We introduce an iterative improvement algorithm to further reduce successively the total routing length by permuting modules on the same row or on the same column.

Three examples are used to test and evaluate our methods. These are summarized in Section VII along with some comparisons with existing methods.

II. PRELIMINARY REMARKS AND EXAMPLES

Consider a two-dimensional board on which modules are to be placed. The board is characterized in terms of a finite array of slots. A slot is a point in an xy-coordinate system. Modules are the entities which are to be assigned to slots to achieve some form of optimum results. Obviously, one module can occupy one and only one slot, and on each slot not more than one module is allowed.

Modules contain pins for connection by physical wires to form signal nets. In other words, a signal net is a set of pins which must be inter-

connected to have the same potential. In treating the placement problem it is customary to disregard the pins but rather consider modules as the basic entity. In the routing problem the interconnection of pins is then taken up. Thus, each signal net in the placement problem becomes a set of modules and signal nets specification defines the connection of all modules.

Let $B = \{b_1, b_2, \dots, b_n\}$ be the set of all modules and $S = \{s_1, s_2, \dots, s_m\}$ be the set of all signal nets. The specification of interconnection is given by the connection matrix $A = [a_{ij}]$ which is m by n ; $a_{ij} = 1$ if signal net i is associated with module j , and $a_{ij} = 0$ otherwise.

Let the board have p rows and q columns of slots. We may assume that the number of modules is equal to $n = pq$ without loss of generality, because dummy modules which are not connected can always be introduced.

Consider an example of a three by three board with five signal nets and nine modules. The connection matrix is

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \\ \textcircled{5} \end{matrix} & \left[\begin{array}{ccccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \end{matrix} \quad (1)$$

Note in the last two rows it is seen that the two signal nets are associated with the same two modules. This means that they actually connect different pins of the same modules. Assume that the modules are placed as in Fig. 1, one possible interconnection is shown.

We wish first to comment briefly on what we mean by routing length.

First, it is meaningful to define the distance between two adjacent slots, either vertical or horizontal, as one unit length. Next, we define the routing length of a signal net as the half-perimeter of the smallest rectangle which encloses the modules in the signal net. Thus in Fig. 1, the routing length of the five signal nets are found to be 2, 3, 2, 4 and 4, respectively. It should be pointed out that in actuality the physical wire length for signal net ③ is equal to 3 in comparison with the half-perimeter routing length 2. Thus our definition represents only an approximation which is convenient to use for various reasons. First, the routing length as defined is easy to calculate and it fits especially well with our approach which is based on decomposing modules into one-dimensional horizontal and vertical blocks. Second, this measure represents a lower bound of the rectilinear Steiner's minimum tree length. In the case of signal nets which are associated with two or three modules, it is equal to the Steiner's tree length. In most practical circuits, 70 to 80% of the signal nets are associated with only two or three modules. Finally the half-perimeter measure incorporates into consideration to some degree the ability of routing a signal net without being blocked by others.

Consider the given pxq modules and the connection matrix A . The first phase in our method is to form p vertical blocks and q horizontal blocks from the above information. A vertical block contains q modules and a horizontal block contains p modules. Each module must belong to one vertical block and one horizontal block. The goal of the decomposition is to achieve minimum signal net interconnection among the vertical blocks, and, similarly among the horizontal blocks. Let us denote the complete set of vertical blocks and the complete set of horizontal blocks by C^v and C^h , respectively. Thus

$$C^v = \{C_1^v, C_2^v, \dots, C_p^v\}$$

$$C^h = \{C_1^h, C_2^h, \dots, C_q^h\}$$

Each C_i^v contains q modules, and each C_j^h contains p modules.

We shall illustrate the above concept with the same example. For the placement given in Fig. 1, $p = q = 3$ and

$$C_1^h = \{1, 4, 7\}$$

$$C_2^h = \{2, 5, 8\}$$

$$C_3^h = \{3, 6, 9\}$$

$$C_1^v = \{1, 2, 3\}$$

$$C_2^v = \{4, 5, 6\}$$

$$C_3^v = \{7, 8, 9\}$$
(2)

Interconnection between blocks as specified by signal nets can be represented by edges connecting different blocks in an interconnection graph. This is shown in Fig. 2a and Fig. 2b for the horizontal blocks and vertical blocks, respectively. The number of edges which represent interconnection in the horizontal blocks is designated by E^h and is equal to five. The corresponding number for the vertical blocks is designated by E^v and is also equal to five.

Next consider a different grouping as given by

$$C_1^h = \{3, 6, 8\}$$

$$C_2^h = \{5, 7, 9\}$$

$$C_3^h = \{1, 2, 4\}$$

$$C_1^v = \{4, 8, 9\}$$

$$C_2^v = \{1, 3, 7\}$$

$$C_3^v = \{2, 5, 6\}$$
(3)

The interconnection graphs are shown in Fig. 3a and Fig. 3b. It is found that for this decomposition $E^h = 5$ and $E^v = 2$, which is an improvement over the first one.

In the second phase of our method we take into consideration the routing length. So far we have decomposed modules into blocks, but we have not decided on the ordering. We shall use a linear placement algorithm to find the optimum ordering for both the horizontal blocks and the vertical blocks. The goal is to minimize the total routing length. The routing length for a particular placement is equal to the sum of the routing length for each signal set. Since we use the half-parameter definition for the routing length of a signal net, it is easy to show that the total routing length for a given placement is equal to the sum of the routing lengths for the horizontal blocks and that of the vertical blocks. Let L be the total routing length, L^h and L^v be the routing length for the horizontal blocks and vertical blocks, respectively, then $L = L^h + L^v$.

For the present example, the routing length for the first decomposition is, according to Fig. 2, $L = L^h + L^v = 8 + 7 = 15$. From Fig. 3, the routing length for the second decomposition is $L = L^h + L^v = 7 + 4 = 11$.

We next consider a different ordering for the second decomposition as shown in Fig. 4. It is found that with the ordering shown, $L = L^h + L^v = 5 + 2 = 7$. For this particular decomposition and ordering, we have the module placement as given in Fig. 5. The total routing length is equal to the sum of the individual routing length for each signal net. They are 1,2,2,1,1 as seen from Fig. 5. Thus we have reduced the total routing length from fifteen for the placement shown in Fig. 1 to seven for that in Fig. 5.

It is to be noted that minimum edge interconnection does not necessarily imply minimum routing length. The examples shown in Fig. 6 illustrate the

point. Therefore, we should be aware of the limitation of our approach discussed so far. Because of this and other reasons, further improvement is possible. We now enter the third phase which is an iterative improvement scheme. By permuting modules on the same row and on the same column we can further reduce the total routing length. This will be discussed in Section V.

III. DECOMPOSITION

The examples of the previous section illustrate the concept of decomposition. In this section we will state the problem in a more formal way and introduce two methods of decomposition. Let us consider the grouping of modules into horizontal blocks first. There are q horizontal blocks and each contains p modules. A signal net is defined as a subset of modules which are associated with the signal net. Thus, depending on the decomposition, a signal net can be represented by the interconnection of a subset of horizontal blocks (or H-blocks) which contain the modules in the net. We may define an H-graph with q nodes representing the q H-blocks and edges specified by the signal net. An edge-sequence is present between nodes representing H-blocks C_j^h and C_k^h if in a signal net there are modules which belong to both C_j^h and C_k^h . If a signal net is common to more than two blocks, say $r = 2$, there exist $(r-1)$ edges in an H-graph, corresponding to the signal net. These edges constitute a linear tree structure among the r blocks. In the same fashion we define the V-graph which has p nodes representing the p V-blocks. Our problem is to find a decomposition in which the total number of edges in the two graphs, $E = E^h + E^v$, is a minimum among all possible decompositions.

This problem is similar to the graph partition problem [6,7] which is an NP-complete problem in the sense of Cook and Karp [8]. Therefore

we need to develop efficient heuristic algorithms. In the following we will first present a constructive algorithm and then an iterative method.

III.1. A constructive method

The constructive method selects modules, one at a time, based on an evaluation function which measures signal net connectivity to modules already selected and then decide which H-block and V-block the selected module belongs to.

STEP 1: Choose a module arbitrarily, say b_1 and let b_1 be the member of C_1^h and C_1^v .

STEP 2: Determine all the members of C_1^v sequentially, one at a time, to have the maximum number of the interconnections with the modules of C_1^v already selected.

STEP 3: Determine all the members of C_1^h from the remaining modules in the same way as in STEP 2. All the members of C_1^v and C_1^h are decided.

STEP 4: $i = 2, j = 2$ and $FLAG = 0$.

STEP 5: Select a module from the remaining modules which has the maximum number of the interconnections with the numbers of both C_i^v and C_i^h which are already selected and assign the module to be a member of C_i^v and C_i^h . If $FLAG = 0$, then to STEP 6; else go to STEP 7.

STEP 6: $i = i+1$.

If $i \leq p$, then go to STEP 5. Otherwise, if $j < q$, then set $FLAG = 1$ and $i = j$ and go to STEP 7; else STOP.*

STEP 7: $j = j+1$.

* It means that all members of all the V-blocks have been selected.

If $j \leq q$, then go to STEP 5.

Otherwise, if $i < p$, then set $FLAG = 0$ and $j = i+1$ and go to STEP 5; else STOP.

The constructive algorithm is a quick and straight-forward method whose solution is not optimum but may be close to it. Obviously, the solution is affected by the module which is chosen arbitrarily in STEP 1. Furthermore, the solution is obtained by checking at local interconnections only.

III.2. An iterative method

The iterative method proposed here is performed more in a global way than the customary strategy of iteratively interchanging modules. The process in which a set of specified modules is reassigned to H-blocks or V-blocks in order to minimize the total number of edges is iterated until no improvement is achieved. The main idea is based on the maximum matching algorithm in a bipartite graph proposed in [9] and can be solved with the Hungarian method. The idea has been adapted in papers [4] and [10], where a minimum matching algorithm is used to minimize the total routing length for unconnected signal nets.

Consider, for example, the first V-block, C_1^V , and let us reassign the modules in C_1^V to different H-blocks. Let $C_1^V = \{b_1, b_2, \dots, b_q\}$ and b_i be a member of the H-block C_i^H . Since elements in the V-blocks are not changed, any reduction in the number of edges in the H-graph by reassignment will not affect E^V . We will introduce a q by q cost matrix denoted by $W = \{w_{ij}\}$ to measure edge interconnection. The i -th row of W corresponds to the module b_i and the j -th column corresponds to the set of modules $C_j^H \triangleq C_j^H - b_j$. The value of w_{ij} is equal to the total number

of interconnection edges between b_i and $C_j^{h'}$, i.e. the number of signal nets common to the module b_i and $C_j^{h'}$.

For the example in Fig. 1, let us consider the grouping given in Fig. 2 with $C_1^v = \{1,2,3\}$. Thus, $C_1^{h'} = \{4,7\}$, $C_2^{h'} = \{5,8\}$ and $C_3^{h'} = \{6,9\}$. The cost matrix is found to be, as seen in Fig. 7

$$W = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \quad (4)$$

The sum of the off-diagonal elements of the matrix W represents the total number of interconnection edges in the H-blocks due to modules in C_1^v only, and it is what we wish to minimize among all possible permutations of the q modules in C_1^v . For the problem at hand it is seen that, by permuting b_1 and b_3 , we obtain

$$W' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \quad (5)$$

Thus we manage to reduce E^h by two without affecting E^v .

The above process is next applied to the first H-block C_1^h . We will be able to decrease E^v while maintaining E^h fixed. The process then continues with C_2^v and C_2^h alternatively. The iteration stops when there is no further improvement.

IV. LINEAR PLACEMENT

All the members of the H-blocks and V-blocks have been decided in the decomposition stage. The next problem to be solved is to find an ordering

of the H-blocks and the V-blocks so as to minimize the total length of the interconnections. The linear ordering problems for the H-blocks and the V-blocks can be solved independently, therefore we will discuss only the linear ordering problem for the H-blocks in this section.

The problem is formulated as follows. Let $C^h = \{C_1^h, C_2^h, \dots, C_q^h\}$ be the family of all the H-blocks and $S = \{s_1, s_2, \dots, s_m\}$ be the set of all the signal nets. Suppose that the H-blocks are arranged in some order, say $O_\alpha^h = \{C_{\alpha_1}^h, C_{\alpha_2}^h, \dots, C_{\alpha_q}^h\}$. The routing length $L^h(O_\alpha^h)$ is to be minimized. This is a linear placement problem [5,11].

First let us introduce a simple way of computing the routing length.

Assume

$$O^h = (C_1^h, C_2^h, \dots, C_q^h) \quad (6)$$

Let $S^h(K^h) \subset S$ be the set of signal nets associated with the set of H-blocks

$$K^h = \{C_1^h, C_2^h, \dots, C_k^h\} \quad (7)$$

then $S^h(C^h - K^h)$ is the set of signal nets associated with the H-blocks which are left over. Let $\ell(C_1^h, C_2^h, \dots, C_k^h)$ denote the number of common signal nets between the two sets of H-blocks, then

$$\ell(C_1^h, C_2^h, \dots, C_k^h) = |S^h(K^h) \cap S^h(C^h - K^h)| \quad (8)$$

measures the density of the interconnection between the two sets of H-blocks as shown in Fig. 8. It is easy to see that for the ordering in Eq. (6), the total routing length for the H-blocks is

$$L^h(O^h) = \sum_{i=1}^{q-1} \ell(C_1^h, C_2^h, \dots, C_i^h) \quad (9)$$

The linear placement algorithm to be used is the ϵ -algorithm

developed in [5]. The ϵ -algorithm produces a feasible solution whose cost is not more than $(1+\epsilon)$ times the cost of the optimum solution. Thus, the smaller the value of ϵ , the better the solution; however the computation time and the required memory space will be larger. We have found with $\epsilon = 1.5$, the computation time is about one tenth of that required for obtaining the optimum solution.

V. ITERATIVE IMPROVEMENT

The wide variety of existing iterative improvement methods can be, basically, divided into three classes: interchange method (such as Pairwise or Neighborhood interchange), relaxation method (such as Force-directed relaxation [1]), and linear assignment method (such as Steinberg's [10] or Rutman's algorithm [4]). Here, we shall consider an iterative improvement method to reduce the routing length by the permutation of modules on the same row or column of a board.

For a given placement of all modules, we will pick up one V-block, say C_k^V , and consider the linear ordering of the modules in C_k^V to minimize the total routing length. The length of a signal net which is not common to any modules in C_k^V remains constant during this manipulation since the modules connected to the signal net remain fixed on the board.

Therefore the problem is reduced to the determination of the linear ordering of the modules in C_k^V so as to minimize the total length of signal nets common to one of the modules in C_k^V . The total length of signal nets with respect to the vertical line is independent of those ordering, since the members of each H-block are not changed and the ordering for H-block is invariant by this manipulation. The ϵ -algorithm

mentioned in Section IV is used to find an ordering of modules.

Consider an example with ten signal nets and six modules, defined by the following connection matrix $A = \{a_{ij}\}$

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \\ \textcircled{5} \\ \textcircled{6} \\ \textcircled{7} \\ \textcircled{8} \\ \textcircled{9} \\ \textcircled{10} \end{matrix} & \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right] \end{matrix} \quad (10)$$

Let a board be composed of three rows and two columns. Suppose that we have the following H-blocks and V-blocks in the decomposition stage:

$$\begin{aligned} C_1^h &= \{1,3,5\} \\ C_2^h &= \{2,4,6\} \\ C_1^v &= \{1,2\} \\ C_2^v &= \{3,4\} \\ C_3^v &= \{5,6\} \end{aligned} \quad (11)$$

We can show that this decomposition has the smallest number of interconnection edges among all decompositions. Interconnections among V-blocks and H-blocks are given in Fig. 9, and the number of edges is given by $E^v = 9$ and $E^h = 6$.

The best linear ordering for this decomposition is given by

$$\begin{aligned} O^h &= (C_1^h, C_2^h) \\ O^v &= (C_1^v, C_3^v, C_2^v) \end{aligned} \tag{12}$$

and is shown in Fig. 10. The total routing length is equal to 18 ($L^h = 12$ and $L^v = 6$). The placement determined by this decomposition and linear ordering is shown in Fig. 11.

We can show that by making permutations within a row or a column, we can further reduce the routing length. In Fig. 11, we have changed the placement of C_2^h from $\{2,6,4\}$ to $\{2,4,6\}$, and the total routing length has been reduced to $L = 17$. This is shown in Fig. 12. It is seen that this new ordering changes the elements in V-blocks as

$$\begin{aligned} C_1^v &= \{1,2\} \\ C_2^v &= \{4,5\} \\ C_3^v &= \{3,6\} \end{aligned} \tag{13}$$

The total number of edges in the V-blocks has actually be increased to ten. Thus, a shorter routine length has been obtained, which corresponds to a decomposition with larger number of edges.

VI. COMPLEXITY OF COMPUTATION

Now, let us analyze the complexity of computation of the algorithms proposed. The following notations are used:

- n: number of modules
- p: number of rows on a board ($\approx \sqrt{n}$)
- q: number of columns on a board ($\approx \sqrt{n}$)
- m: number of signal nets

δ : maximum number of signal nets per module

σ : maximum number of modules per signal net

In our analysis the number of operations is estimated by the number of additions or comparisons. It is shown that the leading term in the computational complexity estimate of the entire process is in the order of $O(\delta\sigma n^3)$.

VI.1. Decomposition phase

(Constructive method)

When a member is decided to belong to C_i^v and C_j^h , we must check the interconnections between $(i-1)+(j-1)$ modules and the other modules. This requires $\delta\sigma \sum_{i=1}^p \sum_{j=1}^q (i-1)+(j-1)$ operations. And we need $(1/2)n^2$ operations to select a module with maximum interconnections. Thus the complexity of computation is in the order of $O(\delta\sigma n^{3/2}, n^2)$.

(Iterative method)

There are q modules in a V-block, thus we need $O(q^3)$ operations to solve the maximum matching algorithm. Suppose we repeat the process for every V-block and every H-block, we need $O(pq^3, qp^3)$ operations or $O(n^2)$ operation.

VI.2. Linear placement phase

The complexity of computation for the ϵ -algorithm is a function of ϵ , δ , σ and the number of modules. The ϵ -algorithm with $\epsilon = 1.5$ (which is used in our program) requires $O(\delta\sigma \times \text{number of modules}^5)$ in our experimental result. There are δp and δq signal nets for H-block and V-block, respectively. Therefore we need $O(\delta\sigma pq^5, \delta\sigma qp^5)$ operations or $O(\delta\sigma n^3)$ operations.

VI.3. Iterative improvement phase

Suppose we apply the ϵ -algorithm with $\epsilon = 1.5$ to every H-block and every V-block, we need $O(\delta\sigma\rho q^5, \delta\sigma q p^5)$ or $O(\delta\sigma n^3)$ operations.

VII. EXPERIMENTAL RESULTS

In order to check and compare the results of our algorithms with others, the algorithms were programmed and tested for three examples in a real problem. The program was written in FORTRAN and run on CDC6400.

Example logic graphs were obtained from Stevens' dissertation [12]. They represent three boards of the ILLIAC IV computer. The smallest, intermediate and largest problems (67, 108 and 151 modules) appearing in [12] were chosen and these examples are referred as Example 1, 2 and 3 in this paper.

Stevens used the actual dimensions of ILLIAC IV boards as the grid size, i.e., there are 10x15 internal card locations and fifteen I/O connector location in one row on the bottom of the board. We reduced the grid size to 5x15 and 8x15 for Example 1 and 2, respectively. The statistics for the example logic graphs are shown in Table 1.

VII.1. Decomposition phase

For each example problem, first the Constructive method was applied and then the Iterative method was used to improve the solution. Number of edges required at the end of the Constructive method and part of the Iterative method are shown in Table 2 and Table 3, respectively. Also included is the computation time for performing each method. In the Iterative method the process is iterated until no further improvement is achieved for any one of the H-blocks or V-blocks.

It has been debated in the literature whether it is better to use

a random start followed by an iterative-improvement or to use a constructive-initial solution followed by an iterative-improvement. The results for five random starts are shown in Table 4. In our experimental results the random start approach is always inferior to the constructive-initial start approach in both the value of the solution and the computation time.

VII.2. Linear placement phase

The ϵ -algorithm with $\epsilon = 1.5$ is applied to find a linear ordering of H-blocks and V-blocks which were obtained by the Decomposition phase. The total routing length of interconnections and computation time for determining a linear ordering are shown in Table 5.

VII.3. Iterative improvement phase

The ϵ -algorithm with $\epsilon = 1.5$ is also applied to find a linear ordering of modules in each H-block and in each V-block. The improvement process is iterated until no further improvement is achieved for any one of the H-blocks or the V-blocks. The results are shown in Table 6.

VII.4. Comparisons with other methods

Until now there are only a few formal placement algorithms which have been proposed. In [2], some of the placement algorithms were programmed and tested for some example logic graphs. In order to compare the results of our algorithms with theirs, we used the same example (Example 3) as was tested in [2].

Our final result for Example 3 was that the total routing length of interconnections was equal to 2098 and the computation time for obtaining the solution was 66.93 seconds. In [2] the length of a signal net is estimated by the rectilinear distance and the minimum spanning tree. We have to transform our estimation to the one used in [2] in order to

compare the algorithms on a common basis. With regards to the estimation of the rectilinear distance and the minimum spanning tree, our solution requires 2133 units in total length.

The main results in [2] compared with our result are summarized as follows. By starting from a solution obtained by a constructive initial placement algorithm, the Force-directed pairwise relaxation algorithm found a solution with 2030 units of total length in about 37 seconds of computation time.

The Force-directed pairwise relaxation algorithm operating on the associated quadratic assignment problem yielded a solution of about 2080 units in 38 seconds.

The Pairwise interchange algorithm operating on the associated quadratic assignment problem found a solution of about 2090 units in 29 seconds.

The other three methods tested in [2] could not find a solution with shorter total routing length.

The method proposed here which consists of decomposition, linear placement and iterative improvement found a solution with total routing length which is 5% longer than the best solution [2]. On the other hand, our method which consists of only the first two phases yielded a much better solution than the conventional constructive method [2]. With respect to computation time, we could not compare each algorithm, since the computer used in [2] is not specified. We feel that with one example it is difficult to comment on the efficiency of one method over another.

VIII. CONCLUDING REMARKS

In this paper we have introduced a new approach to the two-dimensional placement problem. The main idea is to take advantage of what we already know in the much simpler one-dimensional placement problem. The method consists of three phases, namely: decomposition, linear placement and iterative improvement. Heuristic algorithms have been proposed for each phase. The computation complexity of each is analyzed. The method has been tried with examples and some comparisons have been made with existing methods. The results indicate that our method yields comparable results in efficiency. However, no meaningful conclusion can be made based on one example.

One advantage of our approach seems to be that the result obtained from the first two phases before final iteration is superior to other constructive algorithms. More work needs to be done in general areas. The decomposition phase should be investigated further. Perhaps more efficient algorithms can be obtained. The effect of linear placement should be studied with respect to whether an optimum decomposition is indeed needed. Finally, the method should be tried on other examples for the purpose of evaluation and possible improvements.

	Example 1	Example 2	Example 3
# Modules	67	108	151
# Internal modules	52	93	136
# External modules	15	15	15
# Signal nets	138	282	419
Av. # signal nets per module	7.0	7.1	6.5
Av. # modules per signal net	3.4	2.6	2.3
Range of signal net size	2-9	2-9	2-9
Placement Grid Size	5x15	8x15	11x15

Table 1. Statistics of example logic graphs.

	# edges			Computation time (sec)
	Total	Horizontal	Vertical	
Example 1	406	263	143	0.48
Example 2	631	319	312	1.06
Example 3	765	371	394	1.80

Table 2. Number of edges after the Constructive method in the Decomposition phase.

	# edges			Computation time (sec)
	Total	Horizontal	Vertical	
Example 1	351	216	135	2.13
Example 2	596	286	310	2.23
Example 3	701	310	391	9.28

Table 3. Number of edges after the Iterative method in the Decomposition phase.

	Random Start			After Iterative method			Computation time (sec)
	Total	Hor.	Vert.	Total	Hor.	Vert.	
1	898	520	378	764	387	377	18.0
2	902	524	378	769	413	356	17.7
3	912	518	394	775	398	377	16.9
4	886	501	383	741	383	358	19.3
5	931	528	403	798	403	395	17.8
Av.	906	518	388	769	397	372	17.9

Table 4. Five random starts followed by the Iterative improvement for Example 3 in the Decomposition phase.

	Total length			Computation time (sec)
	Total	Horizontal	Vertical	
Example 1	730	514	216	6.53
Example 2	1426	839	587	7.67
Example 3	2281	1285	996	9.59

Table 5. Total routing length after linear placement
in the Linear placement phase.

	Total length			Computation time (sec)
	Total	Horizontal	Vertical	
Example 1	618	449	169	13.34
Example 2	1242	739	513	31.34
Example 3	2098	1150	948	46.26

Table 6. Total routing length after Iterative improvement
method in the Iterative improvement phase.

REFERENCES

- [1] M. Hanan and J. M. Kurtzberg, "Placement techniques," Chap. 5 in Design Automation of Digital Systems; Theory and Techniques, Vol. 1, M. A. Breuer, ed., New Jersey: Prentice-Hall, pp. 213-282, 1972.
- [2] M. Hanan, P. K. Wolff and B. J. Anguli, "Some experimental result on placement techniques," Proc. 13th Design Automation Conference, San Francisco, pp. 214-224, 1976.
- [3] D. G. Schweikert, "A two-dimensional placement algorithm for the layout of electrical circuits," ibid. pp. 490-414.
- [4] R. A. Rutman, "An algorithm for placement of interconnected elements based on minimum wire length," Proc. 1964 SJCC, pp. 477-491.
- [5] S. Goto, I. Cederbaum and B. S. Ting, "Sub-optimum solution of the back-board ordering with channel capacity constraint," 10th Annual Asilomar Conference on Circuits, Systems, and Computers, Nov. 1976. Also submitted to IEEE Trans. on CAS.
- [6] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," BSTJ., vol. 49, pp. 291-307, 1970.
- [7] N. Christofides and P. Brooker, "The optimal partitioning of graphs," SIAM J. Appl. Math., vol. 30, no. 1, Jan. 1976.
- [8] R. M. Karp, Reducibility among Combinatorial Problems in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., New York: Plenum Press, 1972.
- [9] H. W. Kuhn, "The Hungarian method for the assignment problem," Nav. Res. Log. Quart., vol. 2, 1955.
- [10] L. Steinberg, "The backboard wiring problem: a placement algorithm," SIAM Review, vol. 3, no. 1, pp. 37-50, Jan. 1961.

- [11] A. Sangiovanni-Vincentelli and M. Santomauro, "A heuristic guided algorithm for optimal ordering," 13th Annual Allerton Conference on Circuit and System Theory, Oct. 1975.
- [12] J. E. Stevens, "Fast heuristic techniques for placing and wiring printed circuit boards," Ph.D. thesis, Comp. Sci., Univ. of Illinois, 1972.

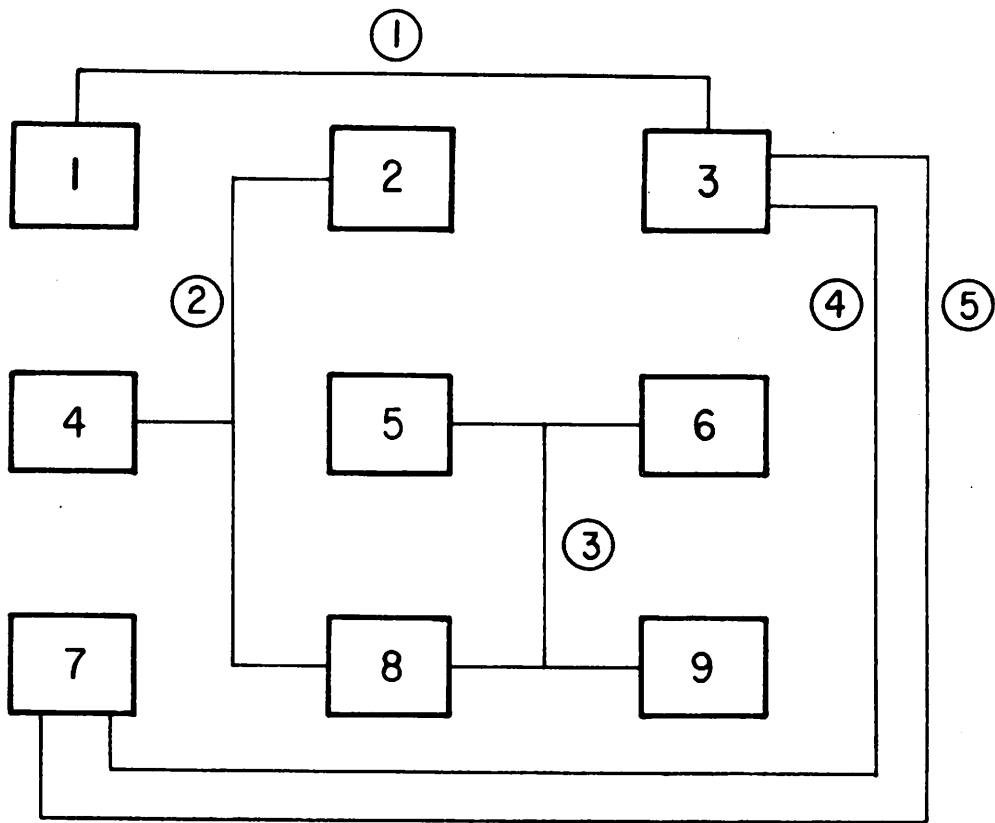
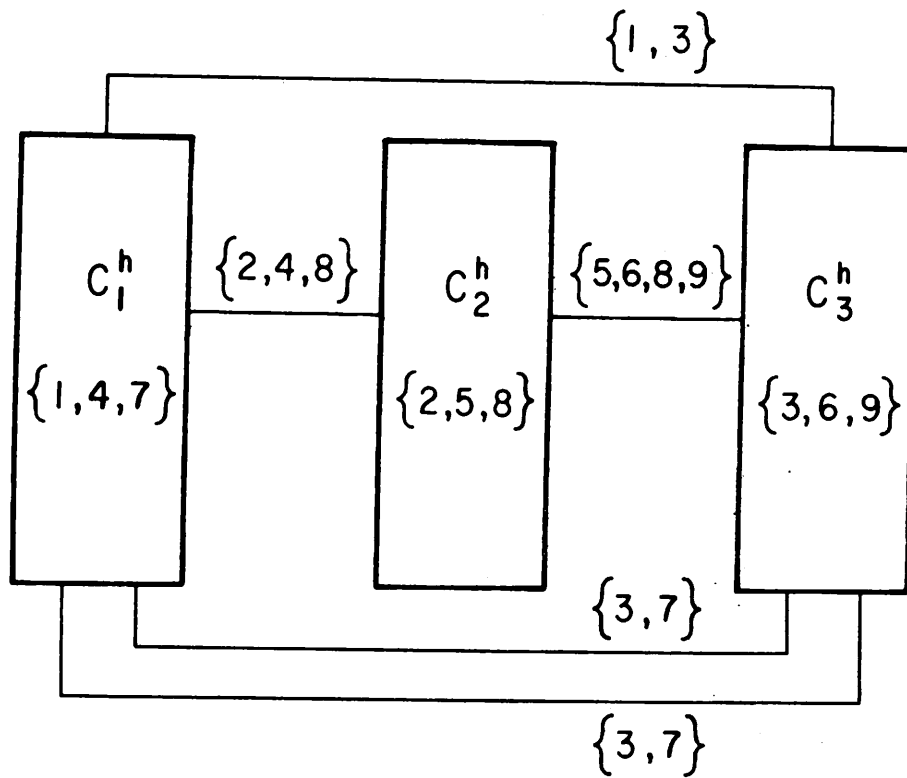
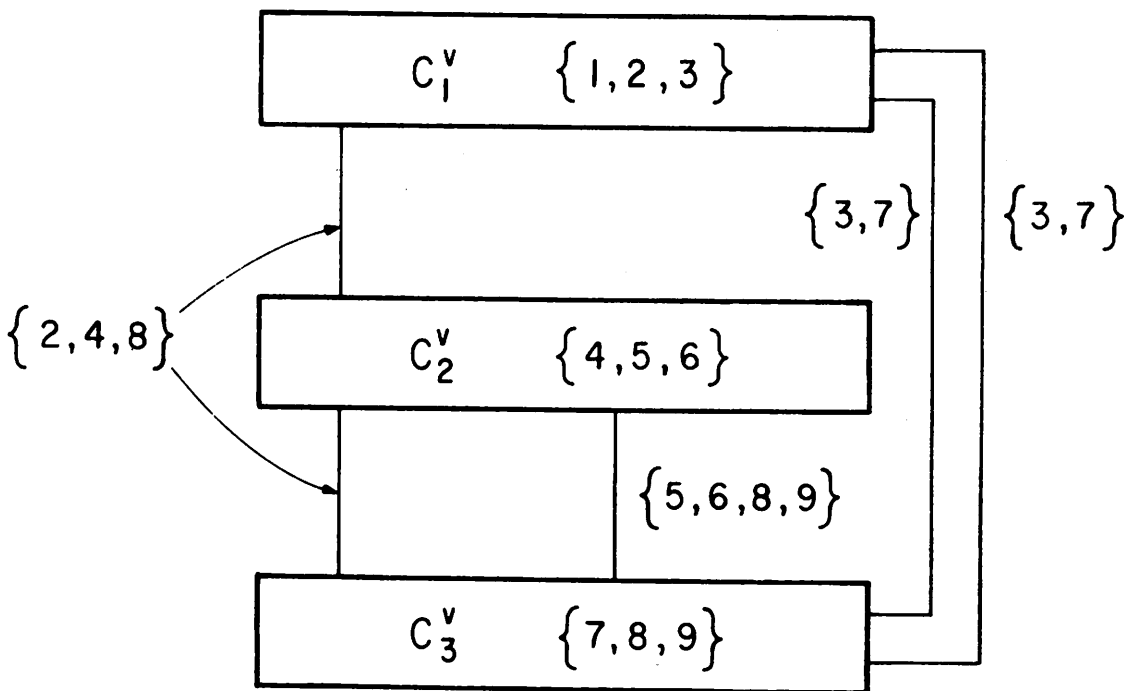


Fig. 1. An example with 5 signal nets and 9 modules placed on a 3 x 3 board.

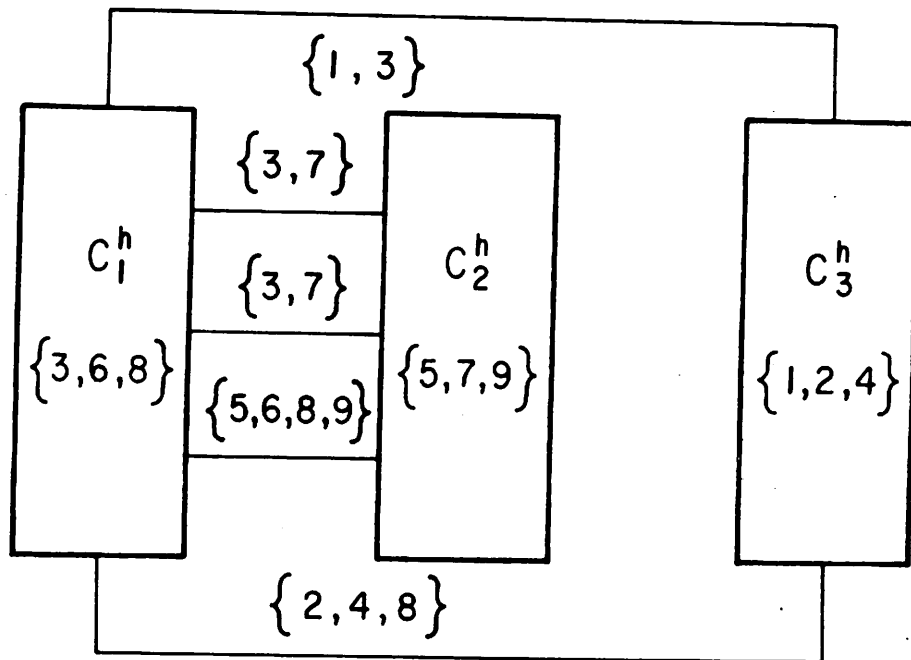


(a)

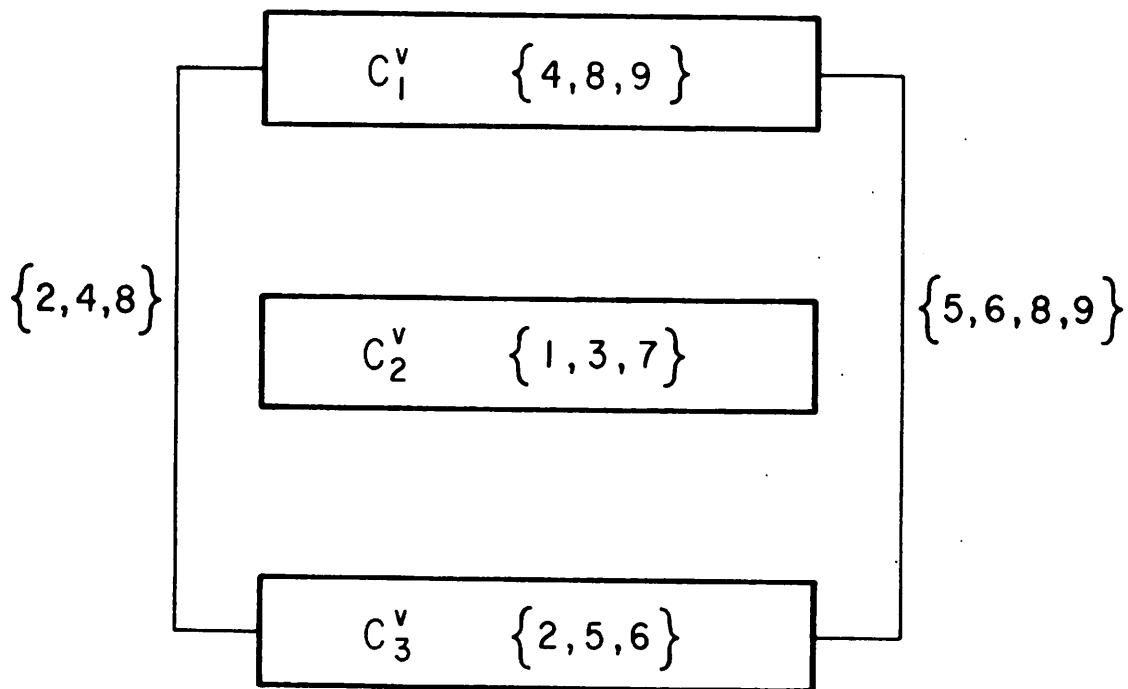


(b)

Fig. 2. (a) Three horizontal blocks with $E^h = 5$. (b) Three vertical blocks with $E^v = 5$.



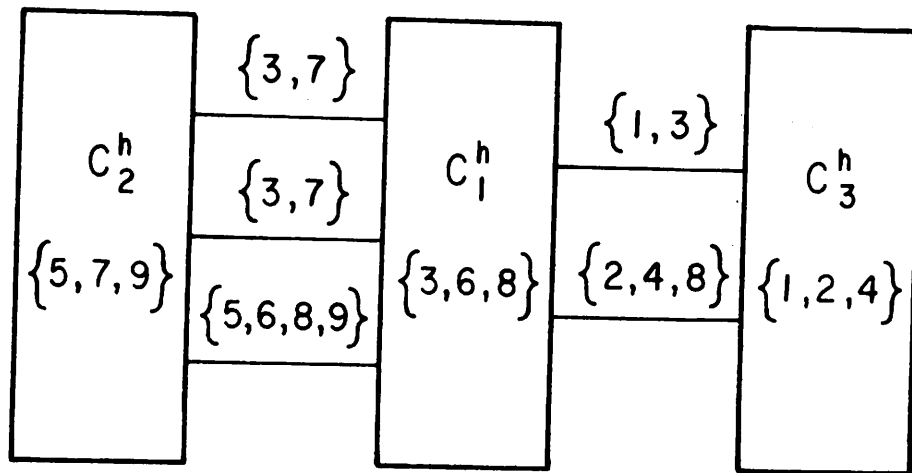
(a)



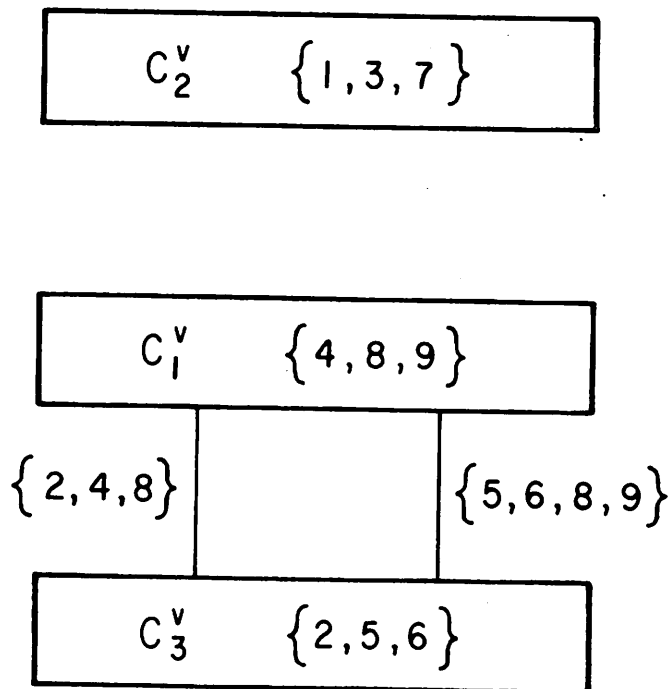
(b)

Fig. 3. (a) A different set of horizontal blocks with $E^h = 5$ and $L^h = 7$.

(b) A different set of vertical blocks with $E^v = 2$ and $L^v = 4$.



(a)



(b)

Fig. 4. (a) A different ordering for the same horizontal blocks as in Fig. 3a with $L^h = 5$. (b) A different ordering for the same vertical blocks as in Fig. 3b with $L^v = 2$.

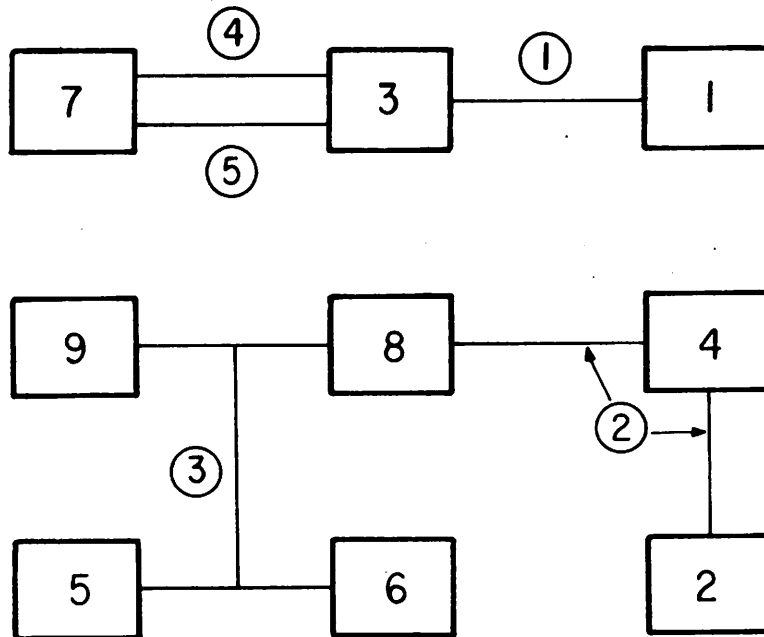
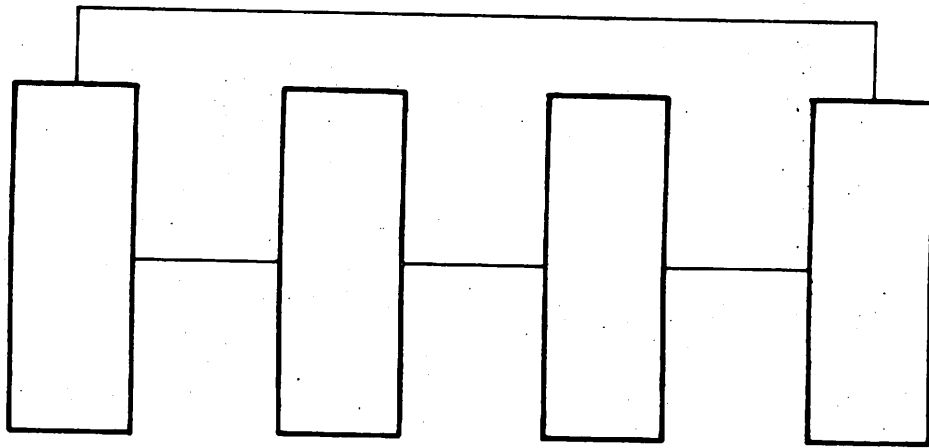
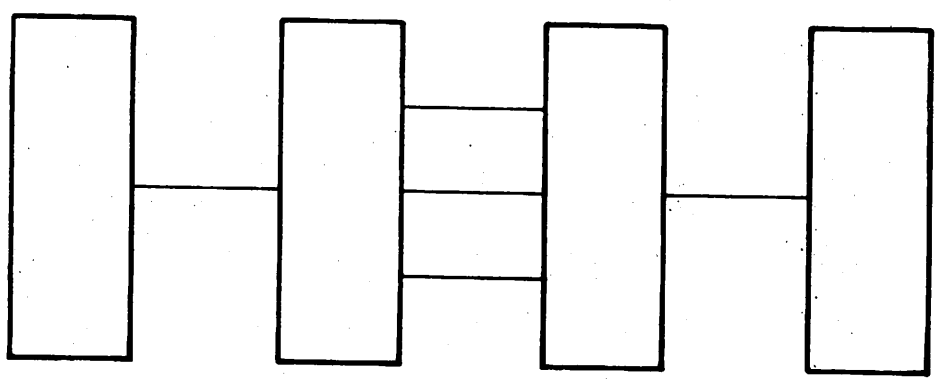


Fig. 5. An optimum placement with $L = 7$.



(a)



(b)

Fig. 6. An example of two interconnections with $E = 4$ and $L = 6$ in Fig. 6a and $E = 5$ and $L = 5$ in Fig. 6b.

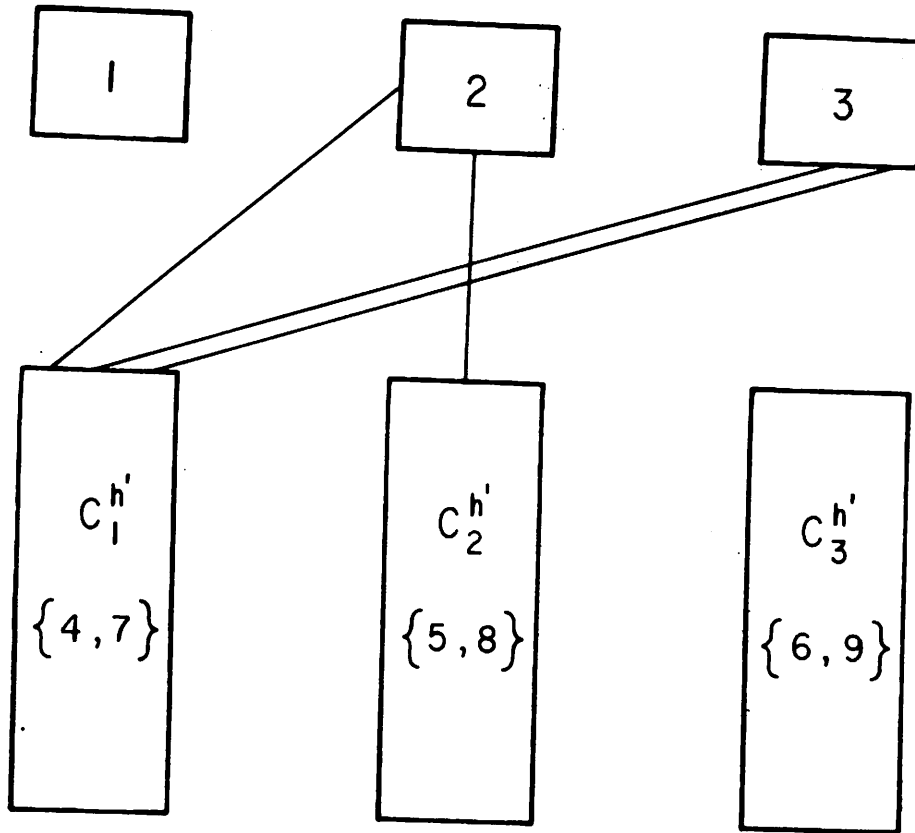


Fig. 7. Example to illustrate the meaning of the cost matrix W .

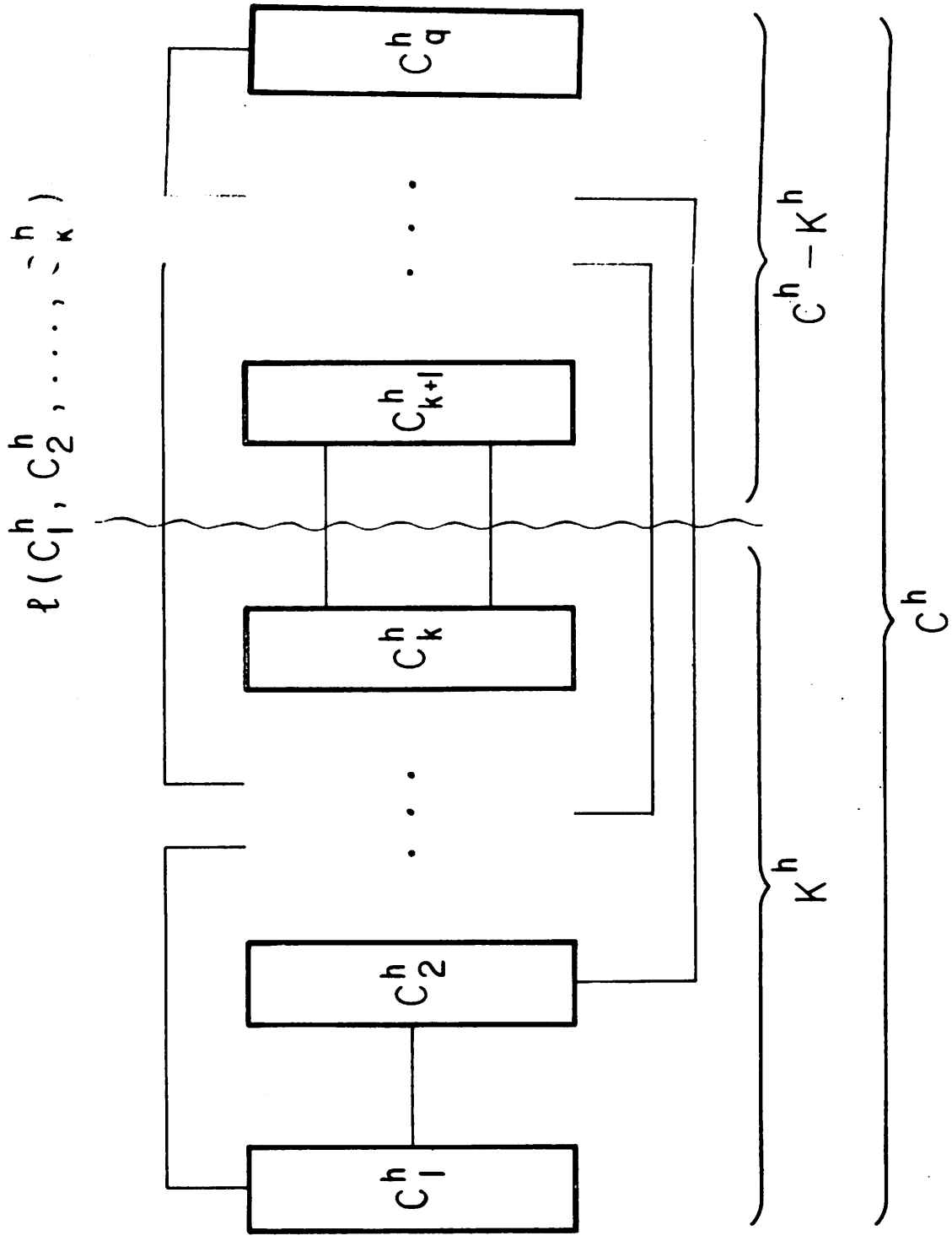
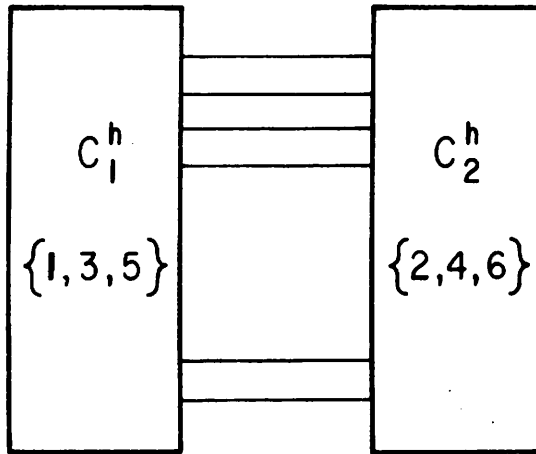
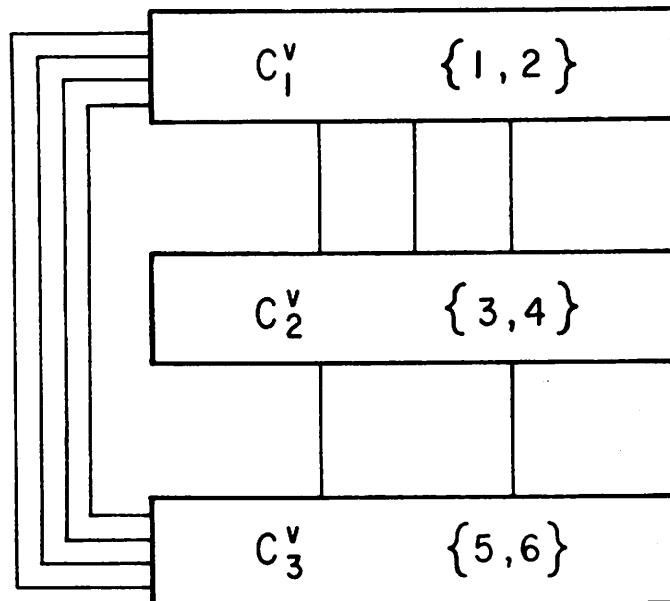


Fig. 8. Routing length computation with H-blocks.

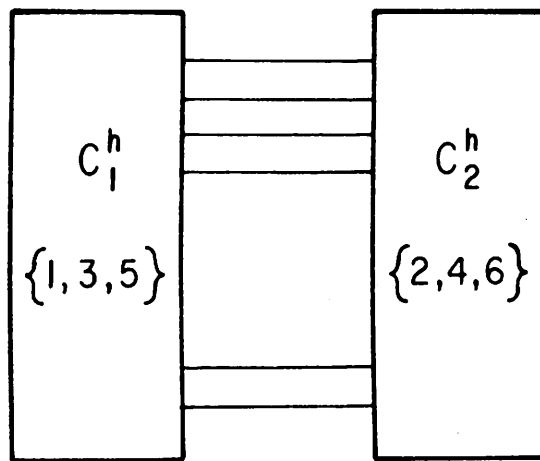


(a)

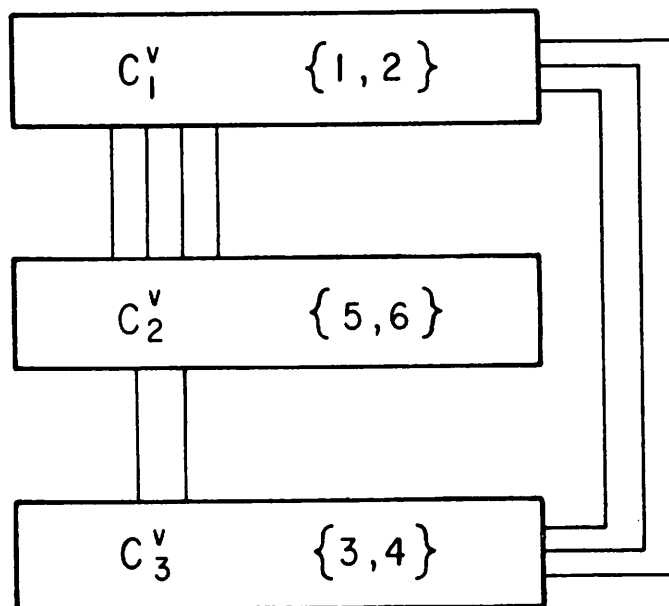


(b)

Fig. 9. (a) Interconnections among H-blocks with $E^h = 6$. (b) Interconnections among V-blocks with $E^v = 9$.



(a)



(b)

Fig.10. (a) With linear ordering for the same horizontal blocks as in Fig. 9a, we have $L^h = 6$. (b) With linear ordering for the same vertical blocks as in Fig. 9b, we have $L^v = 12$.

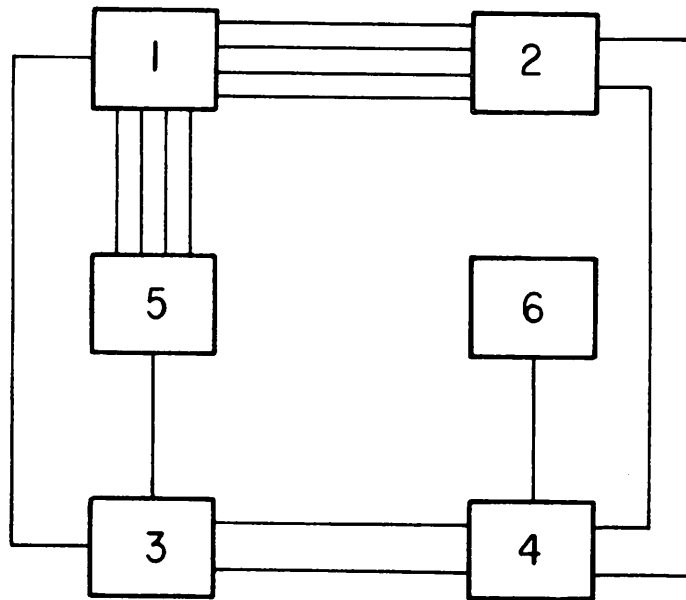


Fig.11. A placement with $E = 15$ and $L = 18$.

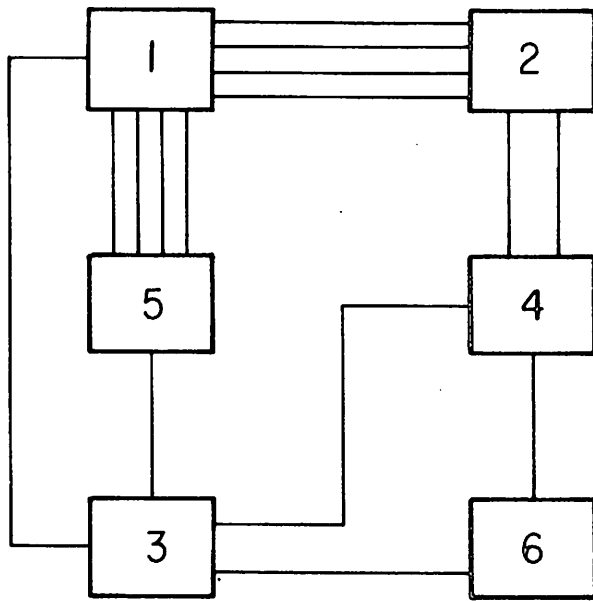


Fig.12. A placement with $E = 15$ and $L = 17$.