MODULE PLACEMENT BASED ON RESISTIVE NETWORK OPTIMIZATION

by

C. K. Cheng and E. S. Kuh

Memorandum No. UCB/ERL M83/35

8 June 1983

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Module Placement Based on Resistive Network Optimization

C.K. Cheng and E.S. Kuh

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

1. Introduction

The problem of module placement is central to automatic layout designer in microelectronics. A good placement is essential to a good layout design. The force-directed method introduced by Breuer and Quinn is a good constructive placement method which leads to initial placement[1]. In their formulation, point modules are assumed, and a force-model is used to determine the state of equilibrium. Hook's Law gives the forces of attraction for modules connected by signal nets, and repulsive forces are used to keep modules apart for those which are not connected. The algorithm amounts to solving a large set of nonlinear equations, which is time consuming. An improvement has been proposed by Antreich, Johnnes and Kirsch using the same force-directed method but with a more systematic formulation of equations[2].

In this paper we propose a more efficient method based on resistive network analogy of the placement problem. The problem can be formulated as an optimization problem with nonlinear constraints. However, if only the linear constraints are considered, the problem amounts to solving a linear sparse resistive network. Thus sparse matrix techniques can be used. Because of its computational efficiency, the procedure is repeated in the overall algorithm of

* Research sponsored by the National Science Foundation grant ECS-8201580 and the Hughes Aircraft Company, Newport Beach, California.

partitioning for the purpose of placing modules on slots. In the formulation, a key idea is that we allow some modules to be fixed in position. Fixed modules could represent I-O pads, but they also play an important role in solving each optimization problem in the overall algorithm.

In section 2 we give a detailed formulation of our approach to the problem. Section 3 is divided into subsections of optimization, scaling, relaxation, and partitioning and assignment. We conclude the paper with a brief discussion of multi-module nets, computation complexity and experimental results.

2. Formulation of the approach

Consider the module placement problem in chip layout. With reference to Fig. 1 where movable modules together with fixed modules represent I-O pads are shown. The movable modules are to be placed on slots where horizontal and vertical lines intersect. The net interconnection specification is given by a net list. We assume that all nets are 2-module nets and multi-module nets have been preprocessed and replaced with 2-module nets[3]. Furthermore, all modules are assumed to have zero dimension, thus their shape, size and pin locations are ignored.

2.1. Objective function

Let the two dimensions on the chip be specified by the x and y coordinates. Let there be a total of n modules located at (x_i, y_i), $i=1,2,....,n$. Let c_{ij} denote the connectivity between module i and module j, i.e., the number of wires between them. Thus $c_{ii}=0$. We choose an objective function which is a measure of the sum of square of wire lengths:

$$\Phi(x,y) = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij} L_{ij}^2 = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij}\left[\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right] \qquad (1)$$

where l_{ij} is the Euclidean distance between module i and module j. It is straight forward to show that Eq. (1) can be written as follows[4]:

$$\Phi(x,y) = \frac{1}{2}\left[x^T Bx + y^T By\right] \tag{2}$$

where

$$B = D - C \tag{3}$$

is an nxn symmetric matrix, $C = [c_{ij}]$ is the connectivity matrix and D is a diagonal matrix whose i-th element d_{ii} is equal to $\sum_{j=1}^{n} c_{ij}$.

With the symmetry between x and y in Eq. (1), we need to consider only the one-dimension problem insofar as optimization is concerned. Thus we dispense with the y coordinate until the end of Sec. 3 where we discuss partitioning and assignment.

2.2. Network analogy

For those who are familiar with circuit theory, B in Eq. (3) is seen to be of the same form as the indefinite admittance matrix of an n-terminal linear passive resistive network. We will model the coordinate of module i, x_i, with a node voltage v_i at node i. The reference coordinate x=0 is thus the datum voltage. The term $-c_{ij}$ in Eq. (3) is then the mutual admittance between node i and node j, and $d_{ii} = -\sum_{j=1}^{n} c_{ij}$ is the self admittance at node i.

The power dissipation in the resistive network is

$$P = \frac{1}{2} v^T Y_n v \tag{4}$$

where v is an n-vector representing the node voltage vector and Y_n is the indefinite admittance matrix which is symmetric. Thus the objective function of the placement problem becomes the power dissipation in the linear passive resistive network. It is well-known that in a passive resistive network the current distributes itself in such a way that the power is minimum. That is, any other current distribution which are not the solution of the network would have a larger power dissipation. In other words the problem of solving network equations is equivalent to that of minimizing a well-selected function which represents power.

2.3. Boundary constraints

Consider the n-terminal resistive network shown in Fig. 2. The first m nodes are floating and their voltages are denoted by an m-vector v_1. The remaining n-m nodes are connected to voltage sources denoted by an (n-m)-vector v_2. Thus the coordinates of the n modules are represented by an n-vector $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ where the coordinates of the fixed modules are specified by v_2 and the coordinates of the movable modules, which are to be determined are represented by v_1.

The network equations are:

$$0 = y_{11}v_1 + y_{12}v_2 \tag{5a}$$

$$i_2 = y_{21}v_1 + y_{22}v_2 \tag{5b}$$

where y_{11}, $y_{12}=y_{21}^T$ and y_{22} are the familiar short-circuit admittance submatrices of the indefinite admittance matrix, Y_n. From (4b), we obtain

$$v_1 = -y_{11}^{-1}y_{12}v_2 \tag{6}$$

which gives the solution of the movable modules in terms of the fixed modules and the admittance submatrices.

Remarks

(1) y_{11} is the short circuit driving-point admittance submatrix of a passive resistive network and is thus positive definite.

(2) The solution of Eq. (6) must fall inside the region defined by the smallest and largest voltages of the voltage sources. This is because in a passive resistive network, no node voltage can lie outside the range of voltage source.

(3) The dissipated power obtained from the solution in Eq. (6) is the minimum among all possible v_1. Any deviation from the solution will result in an increase in power.

2.4. Slot constraints

Up to now we have not imposed the constraint that the movable modules must be located on slots. This means that that voltage vector v_1 when finally determined must represent a set of prescribed discrete voltages called the legal values. Let us designate the prescribed slots in terms of the permutation vector $p=[p_1,p_2,\ldots,p_m]^T$ where p_i is the i-th legal value and m is the total number of the movable modules. Thus the permutation of the m legal values must be assigned to the m components of v_2. To express this in terms of our optimization problem, let $v_1=[x_1,x_2,\ldots,x_m]^T$, i.e., x_i denotes the coordinate of module i or the voltage at node i, we claim that the following equations represent the constraints on the modules or voltages which are required to be on slots:

$$\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} p_i$$

$$\sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} p_i^2$$

$$\vdots \qquad (7)$$

$$\sum_{i=1}^{m} x_i^m = \sum_{i=1}^{m} p_i^m$$

The first equation can be written as

$$1^T v_1 = 1^T p \equiv d \qquad (8)$$

where 1 is a unit vector and d is a constant equal to the sum of the m legal values.

Proof:

=> Let $[x_1, x_2, \ldots, x_m]$ equal to any permutation of $[p_1, p_2, \ldots, p_m]$, Eq. (7) is automatically satisfied.

<= Let us define

$$f(x) = \prod_{i=1}^{m} (x + x_i)$$

Then the coefficients of x^i are multi-variable polynomials of $[x_1, x_2, \ldots, x_m]$. Through simple algebraic operations[5] and by using Eq. (7), we can show that

$$f(x) = \prod_{i=1}^{m} (x + p_i)$$

which implies that all modules are on slots.

Q.E.D.

3. Proposed Method

The proposed method can be divided into subproblems of optimization, scaling, relaxation, and partitioning and assignment. The main idea is to solve a simple optimization problem using linear resistive network analogy repeatedly, and in the process the movable modules are assigned to slots. We shall use node voltges and module coordinates interchangeably in the ensuing discussion for sometimes it is more intuitive to make statements in terms of voltages, while in dealing with the actual placement problem it is more convenient to use the coordinates.

3.1. Optimization

From Eqs. (4) and (5), we wish to minimize the power dissipation

$$P = \frac{1}{2} v^T Y_n v = \frac{1}{2} \begin{bmatrix} v_1^T, v_2^T \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} v_1^T y_{11} v_1 + 2 v_1^T y_{12} v_2 + v_2^T y_{22} v_2 \end{bmatrix} \quad (9)$$

subject to the complete set of constraint equations in Eq. (7). This is clearly not feasible. Therefore, we propose to use only the first equation in Eq. (7), which is a linear constraint expressed by Eq. (8).

The solution to the optimization problem of minimizing P in Eq. (9) subject to the linear constraint in Eq. (8) is given by the well-known Kuhn-Tucker conditions:

$$v_1 = y_{11}^{-1} \left[-y_{22} v_2 + i_1 \right] \quad (10a)$$

where

$$i_1 = \frac{d + 1^T y_{11}^{-1} y_{12} v_2}{1^T Y_{11}^{-1} 1} 1 \quad (10b)$$

It is seen that that first term in Eq. (10a) is precisely that given by Eq. (6) for which there is no constraint on slots. The second term of Eq. (10a) can be viewed as a correction term which attempts to put the solution on slots. In terms of electric network, we may use current sources to interpret the effect as shown in Fig. 3. Thus we have a linear resistive network with both voltage and current sources. In addition, we know that the network is sparse because of the inherent nature of the placement problem. Using well-known sparse matrix algorithm, we can greatly reduce the computation time in comparison with those that use attraction and repulsion forces[1,2].

As mentioned, because only the linear constraint equation is used, the solution will not put modules on slots. As a matter of fact the result will lead to modules more or less confined to the center of the region. Therefore we must introduce ways to bring the modules so obtained to the legal positions. Thus the next step in our overall method is scaling which distributes the solution more evenly over the entire region. However, let us first present the effect of the movement of modules to changes in power dissipation. Let us assume that we deviate away from the solution v_1 of Eq. (10) by δv_1 under the constraint of Eq. (8), i.e.

$$1^T \delta v_1 = 0 \tag{11}$$

Then we claim that the power dissipation is increased by

$$\frac{1}{2} \delta v_1^T y_{11} \delta v_1 .$$

Proof:

From equation (9), we have

$$\Delta P = P(v_1+\delta v_1)-P(v_1) = \frac{1}{2}\left[2\delta v_1^T y_{11} v_1 + \delta v_1^T y_{11}\delta v_1 + 2\delta v_1^T y_{12} v_2\right]$$

From equation (10),

$$y_{12}v_2 = -y_{11}v_1 + i_1$$

and using Eq. (11), we obtain

$$\Delta P = \frac{1}{2}\left[\delta v_1^T y_{11}\delta v_1\right]$$

Q.E.D.

Furthermore, it is possible to derive an upper bound on the increase in power dissipation in terms of y_M, the largest self-admittance in y_{11}. From the Theorem of Gerschgorin[6], we know that the eigenvalues of y_{11} are not larger than $2y_M$, then

$$\Delta P = \frac{1}{2}\delta v_1^T y_{11}\delta v_1 \le \frac{1}{2}\|y_{11}\|\,\|\delta v_1\|^2 \le y_M \sum_{i=1}^{m}\delta v_i^2 \qquad (12)$$

Therefore the increase in power dissipation has an upper bound which is proportional to the norm of the deviation δv_1.

3.2. Scaling

The result of the optimization with linear constraint leads to solutions which have modules concentrated at the center of gravity of all movable modules. The linear constraint dictates the mean position of the modules. The forces of attraction as modelled with linear resistors pull the modules torward the center. The only forces which attempt to scatter the modules are the fixed modules at the boundary. Therefore, in order to be able to partition the modules we will

introduce scaling to redistribute the modules at the expense of increasing the power dissipation. The method used here is to minimize the increase of power ΔP under the constraints which include both the first order and second order equations in Eq. (7). Fortunately, by using the norm of δv_1 in Eq. (12), we again can resort to the well-known Kuhn-Tucker conditions.

Let us assume that in the region or a subregion where there are k modules with legal values given by $[p_1,p_2,\ldots,p_k]$. Let $[x_{o1},x_{o2},\ldots,x_{ok}]$ denote the solution obtained from optimization and let $[x_{n1},x_{n2},\ldots,x_{nk}]$ denote the new solution after scaling. Thus our problem is to minimize

$$\sum_{i=1}^{k}\left(x_{ni}-x_{oi}\right)^2 \tag{13}$$

under the constraints

$$\sum_{i=1}^{k}x_{ni}=\sum_{i=1}^{k}p_i \tag{14}$$

and

$$\sum_{i=1}^{k}x_{ni}^2=\sum_{i=1}^{k}p_i^2 \tag{15}$$

The solution is given by the Kuhn-Tucker conditions, namely: For i=1,2,...,k

$$x_{ni}=\frac{x_{oi}-c_o}{a_o}a_n+c_n \tag{16}$$

where

$$c_n=\frac{1}{k}\sum_{i=1}^{k}p_i \tag{17}$$

$$a_n=\left[\frac{1}{k}\sum_{i=1}^{k}\left(p_i-c_n\right)^2\right]^{\frac{1}{2}} \tag{18}$$

$$c_o = \frac{1}{k}\sum_{i=1}^{k} x_{oi} \tag{19}$$

and

$$a_o = \left[\frac{1}{k}\sum_{i=1}^{k}\left(x_{io} - c_o\right)^2\right]^{\frac{1}{2}} \tag{20}$$

where c_o is the mean position of the computed module positions and a_o is the root mean square amplitude from c_o. If a_o turns out to be very small approaching zero, so is x_{oi}-c_o in Eq. (16), then Eq. (16) must be replaced by

$$x_{ni} = c_n \tag{21}$$

The result of scaling gives an improvement from the result of optimization as for as module location is concerned but at the expense of increasing power dissipation.

3.3. Relaxation

Before partitioning and assigning of modules to slots, we need to perform relaxation to be described below. The method calls for repeated use of scaling and optimization over subregions to be specified by designers. This tends to spread the modules out over the entire region. When a pertinent subregion is considered, modules outside are always kept fixed.

We propose to choose subregions in the following way: First we start from one end of the region, then the other end and, finally, the middle. After the initial optimization over the entire region, three such steps of scaling and optimization over subregions are carried out. The result tends to settle down and is ready for partitioning. Thus we have as

Input: A one-dimensional region with coordinates of movable modules x_i, i=1,2,...,m obtained from initial optimization in the entire region with specified fixed modules x_i, i=m+1,m+2,...,n on the boundary. A parameter β is to be chosen by the designer with $0<\beta<50\%$.

Relaxation:

(1) Order the modules left to right according to coordinates with the smallest first.

(2) Choose $\lceil \beta m \rceil$† modules from the left, setting other modules fixed and do scaling in the left β region.

(3) Fixed the modules so determined in the left β region and release the modules in the right $(1-\beta)$ region. Do optimization.

(4) Choose $\lceil \beta m \rceil$ modules from the right, set other modules fixed and do scaling in the right β region.

(5) Fixed the modules in the right β region and release the modules in the left $(1-\beta)$ region. Do optimization.

(6) Choose $\lceil \beta m \rceil$ modules from the left, set other modules fixed and do scaling in the left β region.

(7) Set modules in both the left β region and the right β region fixed and release the modules in the center subregion. Do optimization.

Output: A one-dimensional region with m modules and new coordinates x_i, i=1,2,...,m.

† $\lceil k \rceil$ means the smallest integer which is larger than k.

3.4. Partitioning and Assignment

We next partition the region into two. The ratio of the left subregion to the right subregion is $\lfloor m/2 \rfloor / \lceil m/2 \rceil$ where $\lfloor k \rfloor$ denotes the largest integer which is smaller than k. We do scaling once more for the left subregion and for the right subregion. As before, in scaling for a subregion we keep those modules outside fixed. The result of this gives two partitioned subregions together with their associated modules.

We next start over again on each subregion, i.e., perform optimization, relaxation, partitioning and scaling independently. However, all modules outside the pertinent subregion under consideration are considered as fixed modules in the ensuing computation.

In the following we will reinstate the y coordinate to consider the 2-dimensional partitioning and assignment problem.

Input: A 2-dimensional region to be partitioned into rectangles each containing a module, a set of m movable modules together with their coordinates and a set of n-m fixed modules.

Assignment:

(1) Do optimization on both the x coordinate and the y coordinate of the movable modules.

(2) While each region contains more than one module

Do

Choose the direction of the cutline.

Cut the longer side of region.

List all current regions.

For each region do partitioning.

(3) For each region, assign the module to the legal value.

4. Discussion

4.1. Multi-module nets

As mentioned in the introduction section, we assume that all nets are 2-module nets in our present treatment. Since multi-module nets are always present, we use the following two models to deal with them:

(1) At the beginning we use a clique to simulate a multi-module net. If there are r modules in a net, the weight of each edge on the clique is 2/r.

(2) After the relative module position is determined, we use a chain to connect the modules. Consider the x direction, we order the modules according to their coordinates; we then link the modules by a chain in this order.

4.2. Computation complexity

The optimization algorithm amounts to a linear resistive network computation. Using sparse matrix technique, we have the computation complexity $O(m^{1.4})$ where m is the number of movable modules. The scaling operation is linear with k where k is the number of modules in a subregion.

As to partitioning and assignment, in each iteration, all current regions are divided into two subregions. It takes $\log_2 n$ iterations to make all the necessary divisions. Thus the total complexity is no more than $O(n^{1.4}\log_2 n)$.

4.3. Experimental results

The 25 module example from [2] is applied to illustrate the procedure of our algorithm. Five external pads are fixed on bottom of the chip and 25 movable modules are to be assigned on the five by five slots.

Figs. 4~7 demonstrate how the module locations evolve from relative positions to slots. On the graph, module positions are indicated by points with module numbers. The connectivity among modules is represented by linking lines.

Fig. 4 is the result of optimization. The module positions are optimal under the constraint that the center of gravity of the modules is at the center of the chip. Relaxation is next carried out and the modules spread over the entire region in the vertical direction. Next partitioning and scaling are used to relocate the modules into two subregions (Fig. 5). Because there are five rows of slots on the chip, the ratio of the sizes of the two subregions is two to three. Fig. 6 is the result of second level relaxation, partitioning and scaling using vertical cutline. Hence module positions are spread out in the horizontal direction. Fig. 7 is the solution of assignment. All modules are located at the centers of grids. The sum of squared length is 324 in comparison with 363 obtained by the force model[1,2].

5. Conclusion

The partitioning and placement problem has been formulated in terms of linear resistive network optimization. The objective function used is the sum of squared wire length which corresponds to power dissipation in the network. Fixed modules become nodes with constant voltage sources. Movable modules then correspond to nodes whose voltages are to be determined. Since modules must be put on slots, a set of constraint equations are imposed on the modules.

We consider only the first order constraint which, in essence, fixes the center of gravity of the movable modules. The optimization calculation can thus take advantage of the sparse matrix technique.

To assign modules to slots, we introduced scaling, relaxation, partitioning and assignment. Experiment results indicate that our method not only is computationally efficient but also leads to excellent placement in terms of our objective function. It appears the method can be extended to modules of irregular shape and size. Thus potentially it will be useful for gate-array and building-block placement.

References

[1]. N. R. Quinn, Jr. and M. A. Breuer, "A Force Directed Component Placement Procedure for Printed Circuit Boards," IEEE Trans. on Circuits and Systems, vol. CAS-26, No. 6, pp. 377-388, June 1979.

[2]. K. J. Antreich, F. M. Johnnes and F. H. Kirsch, "A New Approach for Solving the Placement Problem Using Force Models," Proc. IEEE Int. Symp. on Circuits and Systems, pp. 481-486, 1982.

[3]. D. G. Schweikert and B. W. Kernighan, "A Proper Model for the Partitioning of Electrical Circuits," Proc. 9th Annu. Design Automation Workshop, pp. 56-62, June 1972.

[4]. K. M. Hall, "An r-Dimensional-Quadratic Placement Algorithm," Management Sci., vol. 17, no. 3, pp. 219-229, November 1970.

[5] N. Jacobson, "Basic Algebra," San Francisco: W.H. Freeman and Company, pp. 133-135, 1974.

[6] R. S. Varga "Matrix Iterative Analysis," New Jersey: Prentice-Hall, Inc., 1962.

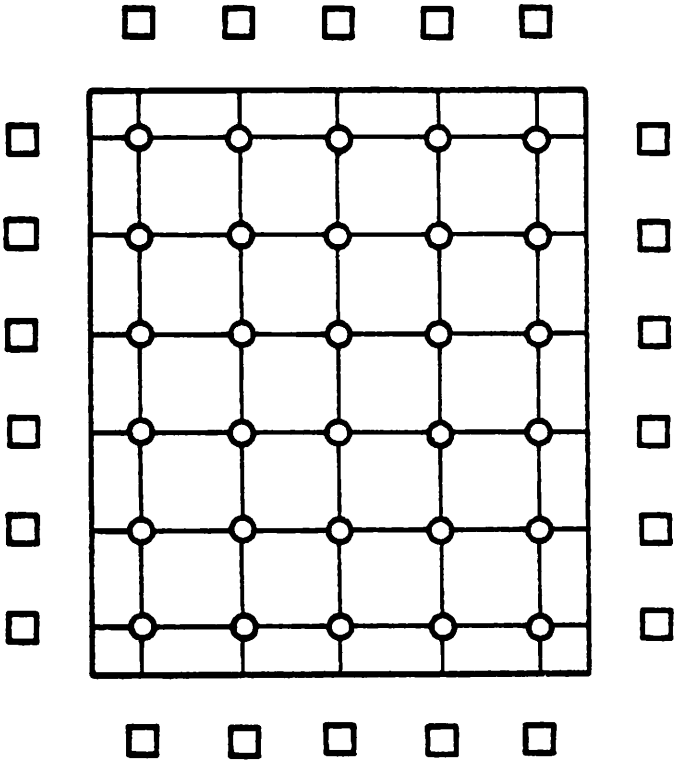

Fig. 1 Chip with movable modules to be placed on slots and fixed modules on the boundary representing I-O pads.

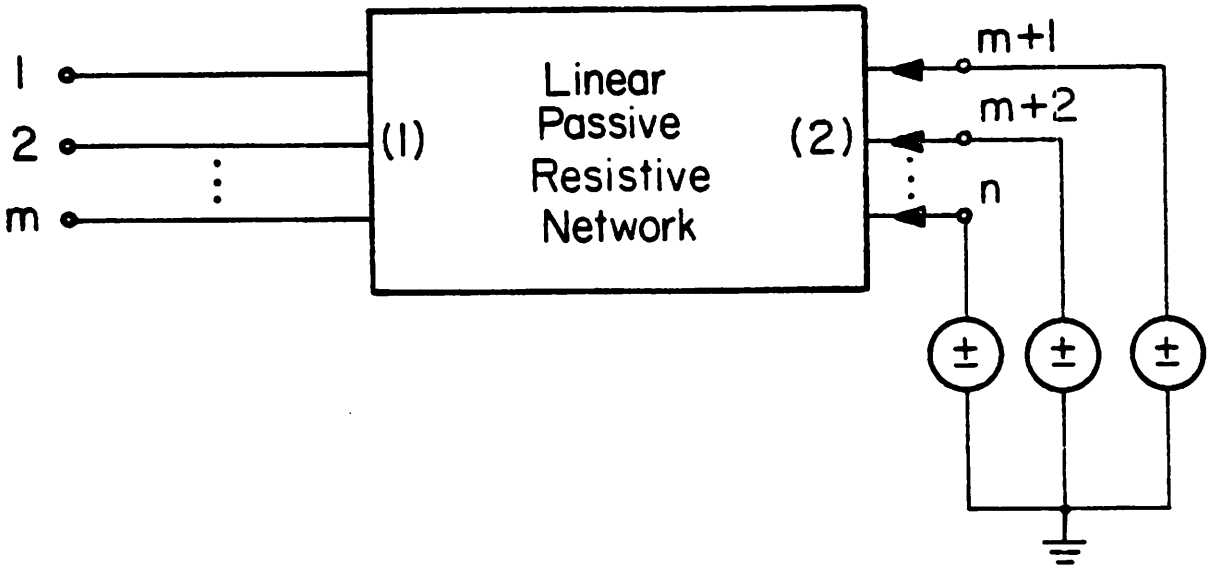

Fig. 2 An n-terminal linear, passive resistive network whose first m nodes are floating and the remaining n-m nodes are connected to voltage sources.

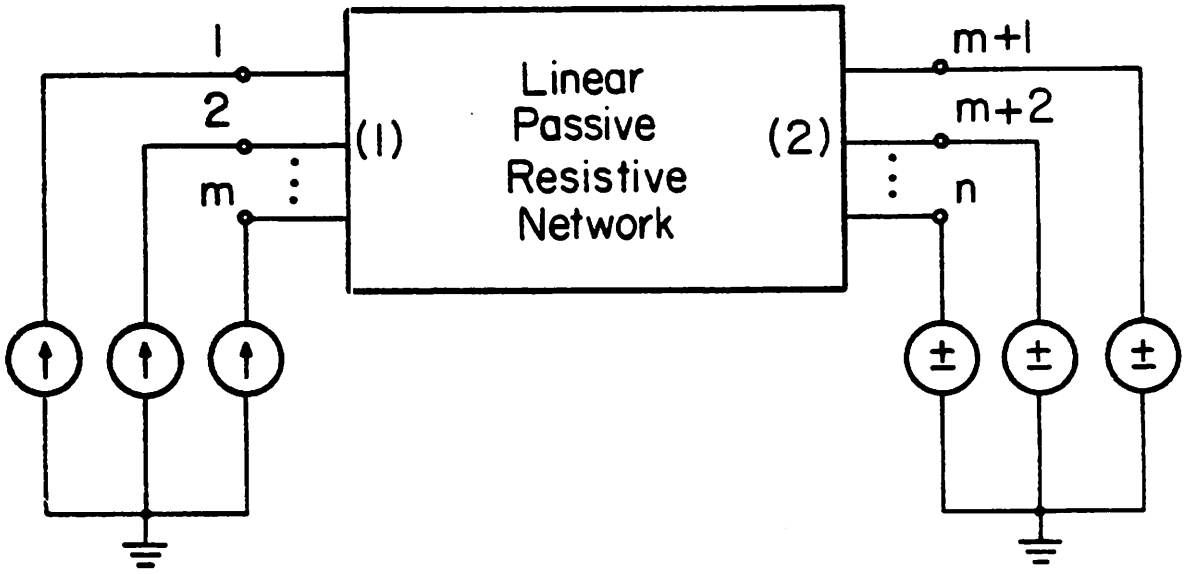

Fig. 3 Network interpretation of the optimization problem with linear constraints.

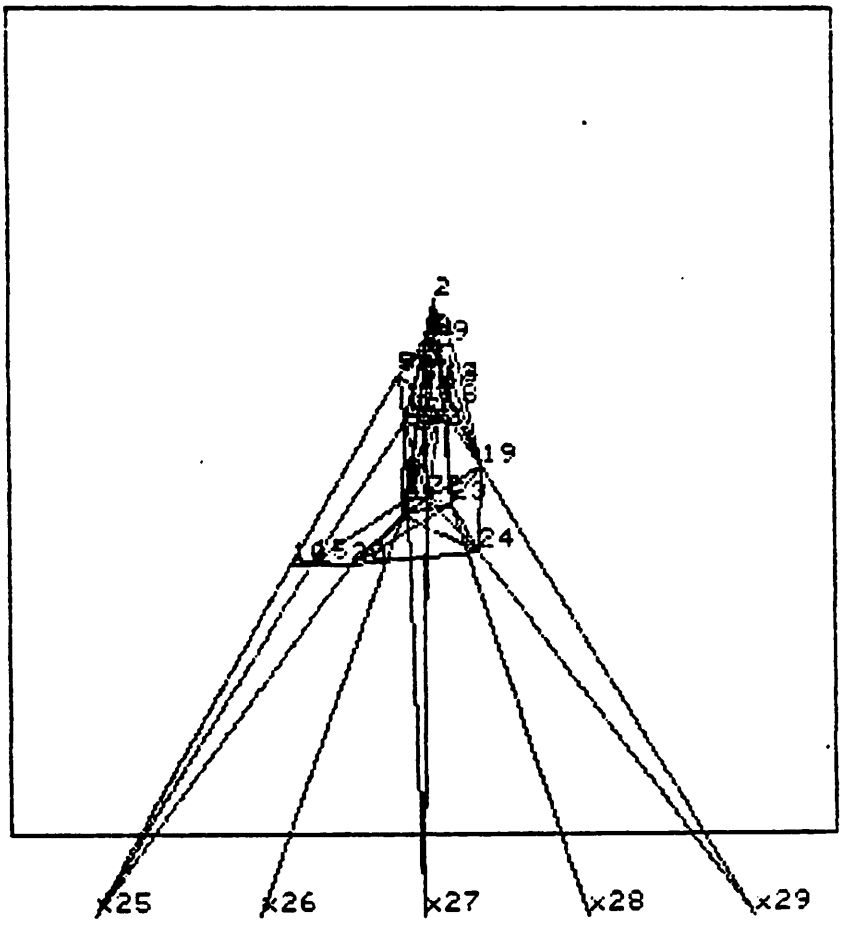

Fig. 4 Result of assignment step (1) on 25 module example. The module positions are optimal under the constraint that the center of gravity of the modules is at the center of the chip.

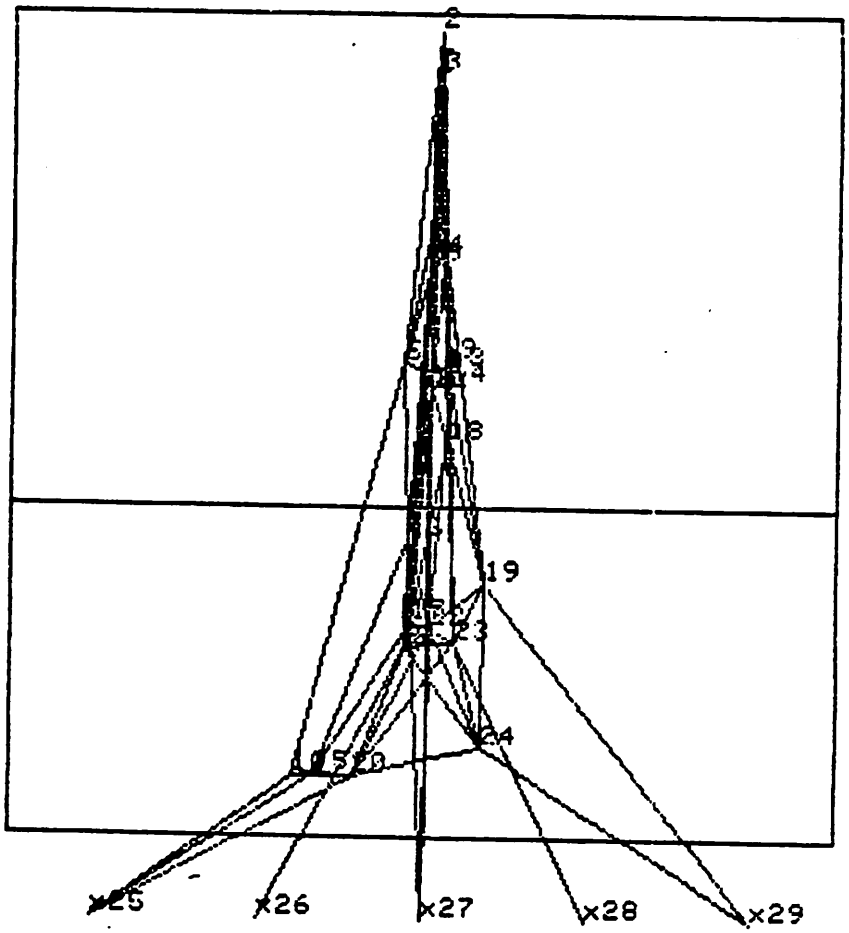

Fig. 5 Result of first level partitioning and scaling. The ratio of the sizes of the two subregions is two to three.

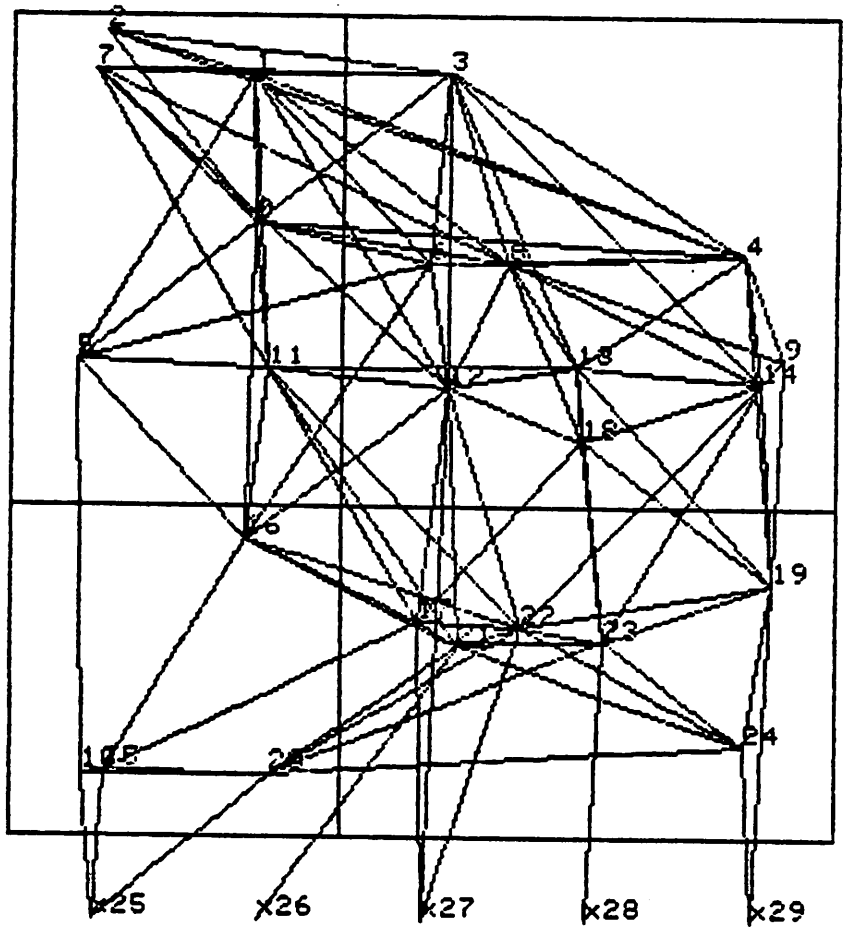

Fig. 6 Result of second level partitioning and scaling.

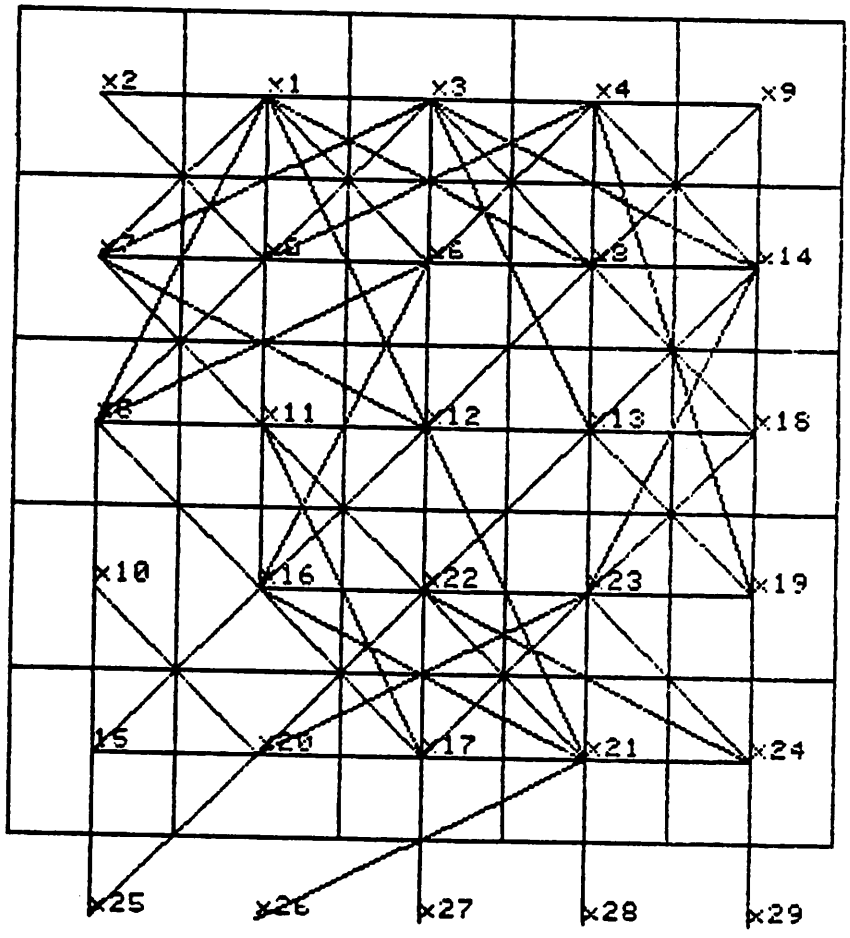

Fig. 7 Solution of assignment.