# AUTOMATIC FLOORPLAN DESIGN

by

Ralph H.J.M. Otten

Mathematical Sciences Department
IBM Thomas J. Watson Research Center, Yorktown Heights, USA

and

Department of Electrical Engineering
Eindhoven University of Technology, Eindhoven, The Netherlands

**Abstract: The problem of allocating area to modules at the highest level of a top-down decomposition is treated in this paper. A theorem of Schoenberg is applied to obtain a good embedding of the module space into the plane. The dutch metric is introduced to transform netlist information - if available - into a distance matrix. This metric is flexible enough to enable the user to steer the design in an interactive environment, and rigorous enough to yield results satisfying optimality criterions. The embedding is used to derive the topology of the floorplan in the form of the structure tree of a slicing structure. To store the partial structure tree during the construction a concise and convenient data structure, the shorthand tree, is introduced. For any aspect ratio of the chip a minimum area floorplan can be generated.**

**The paper also shows how wiring space predictions can be incorporated, how varying degrees of module flexibility can be accounted for, and how fixing bonding pad macros affects the procedure.**

## 1. INTRODUCTION

At the preceding Design Automation Conference (1981) Heller formulated the macro-placement problem for macros without restrictions on their shape. Elsewhere [4] he translated the initial data for this problem into a graph with weights on the nodes and the edges. The nodes represent the macros realizing high level functions together composing the whole system to be integrated. The weights on the nodes are rough estimates of the silicon area to be used by each macro including its internal wiring. The presence of an edge indicates the existence of global interconnections between the associated macros. The weight is the interconnection count for that pair.

In this paper we describe an efficient solution to this problem. For the sizes Heller had in mind the computation time is less than a second so that early in the design trial floorplans are available that can be used to derive data about delays for critical signals, periphery difficulties, congestion problems, etc. But more general problems with macros of varying degree of flexibility can be handled by the methods presented here. The paper, therefore, addresses the allocation of modules at highest level of a top-down decomposition.

Let us briefly outline the procedure. The task is to assign to each module a rectangular area that can accommodate that module, possibly assess and satisfy wiring space requirements between the module areas, while keeping the total area small and possibly constrained. The initial data contains a module characterization at least having an area estimate and net data from which at least an estimate for the global interconnection count can be extracted. From the data a distance space is derived which is used to position the module centers in the plane. From this configuration a slicing[1] structure (figure 1) is constructed on the basis of deformation calculations. At this point the widths of the channels in this structure might be estimated. The information then obtained can be translated into a set of constraints for an optimization procedure that delivers the dimensions, position and orientation of the modules realizing the best floorplan under these constraints.

## 2. THE MODULE SPACE

Using the word *space* suggests location, proximity or distance, continuity, etc., in short a set with a topology. There are several ways of assigning a topology to a set. Considering the data available it seems advisable to search for a method that establishes a direct connection with the notion of proximity by transforming part of the data into a *distance space*, i.e. associating with each pair of modules a unique non-negative real number and with each one-element subset of the set of modules the number 0. A distance space of m objects is completely specified by a matrix $\mathbf{D}_{mm}$, its *distance matrix*:

$$i \neq j: \quad \delta(\{m_i, m_j\}) = d_{ij} = d_{ji} \in \mathscr{R}^+$$

$$\delta(\{m_i\}) = d_{ii} = 0 \qquad (1)$$

It is a non-negative, symmetric matrix with zeros on the principal diagonal.

In case the only data relating a module with another is the global interconnection count, i.e. the expected number of nets connecting that module with the other, this number can be - at best - interpreted as a measure for the desired relative mutual proximity. So any monotone, non-increasing, positive function of the global interconnection count which vanishes at infinity must be acceptable as a distance function for the procedure to be described.

Due to the complete independence of a distance assigned to one pair from the distances assigned to other pairs it is very easy to form distance spaces, but without any kind of tie-up between mutual distances many undesirable features might occur. A simple relation between the mutual distances of any subset of three elements, such as the triangle inequality, is sufficient to eliminate the anomalies of general distance spaces, and still general enough to include (and define) the very useful class of metric spaces. It is of course not true that any metric distance space can be congruently embedded in a euclidean space of some dimension, let alone in a two-dimensional euclidean space.

Menger [6] used properties of minors of the Cayley-Menger determinant to study the embedding problem. Schoenberg addressed the problem of congruent embeddings of a distance space into a euclidean space of given finite dimension in a paper published in 1935 [9] and gave necessary and sufficient conditions for the existence of such an embedding. In the next section we will rephrase his theorem such that the construction of the embedding - if it exists - is explicitly given.

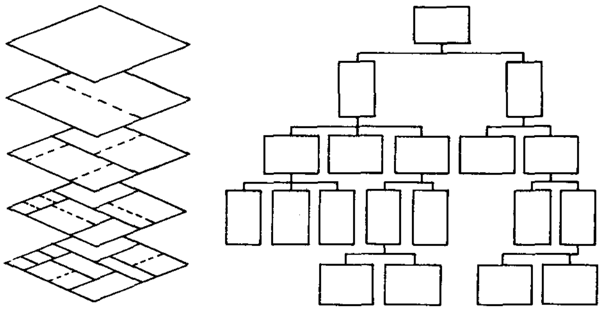


**Figure 1. Slicing is the division of a rectangle into smaller rectangles by parallel lines. The operation can be applied to each of the resulting rectangles (slices) with lines perpendicular to the previous set of dividing lines. Slicing can be repeated to any depth alternating the orientation of the dividing lines. A slicing configuration can be represented by an ordered rooted tree.**

[1] The terminology in this paper is consistent with [8] of which [7] is a concise version.

0420-0098/82/0000/0261$00.75 © 1982 IEEE

## 3. SCHOENBERG CONSTRUCTIONS

The m rows of a real matrix $\mathbf{Q}_{mr}$, called a *configuration matrix*, are to be interpreted as the positions of m points in an r-dimensional euclidean space. Since congruent transformations of their configuration in this space are not of interest, we assume the centroid to be at the origin, which means that the column sums of $\mathbf{Q}$ are 0. Let $\hat{d}_{ij}$ be the distance between the i-th and j-th point, and let $\hat{d}_{0i}$ be the distance of the i-th point from the origin. Then, by the cosine law, we have

$$\mathbf{q}_{i.}\mathbf{q}_{j.}^{T} = \frac{1}{2}(\hat{d}_{0i}^{2} + \hat{d}_{0j}^{2} - \hat{d}_{ij}^{2}) \,. \tag{2}$$

In order to eliminate the distances of the individual points from the origin we express these distances in interpoint distances:

$$\hat{d}_{0i}^{2} = \frac{1}{m}\sum_{k=1}^{m}\hat{d}_{ik}^{2} - \frac{1}{2m^{2}}\sum_{h=1}^{m}\sum_{k=1}^{m}\hat{d}_{hk}^{2} \tag{3}$$

using the fact that

$$\sum_{k=1}^{m}\mathbf{q}_{k.} = 0 \,. \tag{4}$$

Substituting (3) into (2) gives

$$\mathbf{q}_{i.}\mathbf{q}_{j.}^{T} = -\frac{1}{2}\left(\hat{d}_{ij}^{2} - \frac{1}{m}\sum_{k=1}^{m}(\hat{d}_{ik}^{2} + \hat{d}_{jk}^{2}) + \frac{1}{m^{2}}\sum_{h=1}^{m}\sum_{k=1}^{m}\hat{d}_{hk}^{2}\right) \tag{5}$$

or in matrix form with $\mathbf{Z}_{mm} = \mathbf{I}(\mathbf{j}_m) - \frac{1}{m}\mathbf{J}_{mm}$:

$$\mathbf{Q}\,\mathbf{Q}^{T} = -\frac{1}{2}\mathbf{Z}\,\hat{\mathbf{D}}^{\bar{2}}\,\mathbf{Z} \tag{6}$$

where $\mathbf{I}$ is a diagonal matrix with the values of the argument on the diagonal and zeros off the diagonal, $\mathbf{J}$ is the matrix, and $\mathbf{j}$ the vector of all ones, and $\hat{\mathbf{D}}$ is the matrix with zeros on the diagonal and $\hat{d}_{ij}$ in the i-th row and the j-th column.[2]

The result obtained is that (6) applied to a distance matrix $\hat{\mathbf{D}}$ of a configuration of m points in the r-dimensional euclidean space yields a positive semidefinite symmetric matrix of which the rank does not exceed r. But also the converse is true: whenever a distance matrix $\mathbf{D}$ substituted in (6) gives a positive semidefinite matrix, there exists a congruent embedding of this distance space into a euclidean space. For there exists an orthonormal matrix $\mathbf{U}$ such that

$$\mathbf{U}\,\mathbf{I}(\Lambda_m)\,\mathbf{U}^{T} = -\frac{1}{2}\,\mathbf{Z}\,\mathbf{D}^{\bar{2}}\,\mathbf{Z} \tag{7}$$

where $\Lambda_m$ is the vector of eigenvalues of the matrix in the right hand side (in decreasing order). By setting (all eigenvalues in $\Lambda$ are nonnegative)

$$\mathbf{Q} = \mathbf{U}\,\mathbf{I}(\overline{\Lambda^{1/2}}) \tag{8}$$

we obtain the configuration matrix of the configuration realizing the interpoint distances given by $\mathbf{D}$. The columns of $\mathbf{U}$ are the normalized eigenvectors of (6), so Schoenbergs theorem states:
*A distance space with distance matrix* $\mathbf{D}$ *is embeddable in an r-dimensional euclidean space fif its Schoenberg matrix*

$$-\frac{1}{2}\,\mathbf{Z}\,\mathbf{D}^{\bar{2}}\,\mathbf{Z}$$

*is positive semidefinite and has rank r or less.*
*The matrix of which the columns are the eigenvectors of the Schoenberg matrix with length equal to the square root of the associated eigenvalue, is the configuration matrix of the configuration realizing the distances of* $\mathbf{D}$.

This theorem is a complete solution to the problem of embedding a given distance space congruently into a given euclidean space. So it enables us to answer the question whether a certain distance space derived from the module and net data available, can be embedded in the euclidean plane. In general this will not be the case. Quite often, however, the structure at the highest level of a top-down design is almost embeddable in the plane, i.e. there is an embedding in the two-dimensional euclidean space realizing distances $d'_{ij}$ instead of $d_{ij}$ such that the function

$$\phi = \sum_{i=1}^{m}\sum_{j=1}^{m}(d_{ij}^{2} - d'^{2}_{ij}) \tag{9}$$

is relatively small. To obtain such an embedding in case there is some euclidean space in which the given distance space can be congruently embedded, we might project the r-dimensional configuration into a suitable two-dimensional subspace. If $\mathbf{T}_{rr}$ is the matrix of an orthonormal transformation of which the first two columns span the projection plane, the value of $\phi$ is

$$\phi(\mathbf{T}) = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=3}^{r}(\mathbf{q}_{i.}{}^{T}\mathbf{t}_{.k} - \mathbf{q}_{j.}{}^{T}\mathbf{t}_{.k})^{2} \tag{10}$$

which is minimal if the other r-2 columns of $\mathbf{T}$ span the same subspace as the eigenvectors associated with the r-2 smallest nonzero eigenvalues of the Schoenberg matrix. Or, equivalently, if the projection plane is spanned by the two eigenvectors associated with the two largest eigenvalues of the Schoenberg matrix. The corresponding two-dimensional configuration is given by the first two columns of the configuration matrix.

So, given a distance matrix, forming the Schoenberg matrix and determining the partial eigensolution containing the two largest eigenvalues seems to be a good embedding procedure, when the Schoenberg matrix is positive semidefinite. In general, with an arbitrary distance matrix, applying that procedure results in a two-dimensional configuration of which the Schoenberg matrix $\mathbf{S}'$ is the best approximation of the Schoenberg matrix $\mathbf{S}$ derived from the original distance space, in the sense that

$$\psi = \mathbf{j}_m{}^{T}(\mathbf{S}-\mathbf{S}')^{\bar{2}}\,\mathbf{j}_m \tag{11}$$

is minimal, i.e. the sum of the squares of the differences between corresponding entries of $\mathbf{S}$ and $\mathbf{S}'$ is minimal.

The proofs of the assertions in the last part of this section are not trivial, and not without elucidating features. Space limitation, however, makes it impossible to include them.

## 4. THE DUTCH METRIC

The procedure presented in the previous section delivers a two-dimensional configuration with minimal $\psi$ for any distance space. If the distance space is congruently embeddable into some euclidean space, the resulting configuration is also optimal in the sense that

$$\phi = \mathrm{MIN}\{\ \phi(\mathbf{T}) \mid \mathbf{T}\,\mathbf{T}^{T} = \mathbf{I}(\mathbf{j}_r)\ \}. \tag{12}$$

This cannot be achieved by a nontrivial function of the global interconnection count. This is not the only reason why such a distance function is not satisfactory. The boundedness of the chip and the perimeter requirements of the individual modules are completely neglected, and the proximity preferences are mutually independent and, consequently, mostly not well balanced by such a function. Whenever there is more information available about the interconnection structure of the concerned level, it should be used to diminish these flaws. In case a netlist is available, the *dutch metric*, to be introduced in this section, yields a distance space which is congruently embeddable into a finite-dimensional euclidean space, has no distance exceeding 1, takes all connections of the modules involved into account (not only the nets they have in common), and allows for weighting individual nets (for example to force critical nets to be short).

[2] A bar over a matrix or vector operation indicates that the operation is to be performed elementwise.

With m modules and n nets the netlist can be represented by a (0-1) matrix $\mathbf{P}_{mn}$. There are one-to-one correspondences between the rows and the modules, between the columns and the nets, and between the pins and the nonzero entries of **P**. An entry $p_{ij}$ equals 1 if the corresponding module and net share a pin. Otherwise $p_{ij} = 0$. Adopting the same order for the nets as in **P** the weights assigned to the nets can be represented by a positive vector $\mathbf{w}_n$. The higher the weight of a net, the closer the modules connected by this net should be together. From **P** and **w** we want to derive the distance matrix **D**.

Since we want to base the matrix **D** on the presence of connections, and not on the absence of connections, it is more natural to speak about proximity and a nonnegative proximity matrix $\mathbf{C}_{mm}$, than about distances. The elements on the diagonal of **C** should be the same, and greater than all the off-diagonal elements. If the diagonal elements are chosen to be one, a monotone increasing function of $1 - c_{ij}$, bounded on [0,1] and vanishing at the origin might serve as a distance function. Concavity of this function is to be preferred over convexity, since it discriminates the shorter distances more than the longer ones.

A module having many pins needs a large perimeter and, possibly, claims proximity to many other modules. In the two-dimensional chip environment this cannot be realized by arbitrarily short center-to-center distances. Besides, those other modules do not claim proximity only to that module. A measure for proximity should therefore express a balanced sharing of the neighbourhoods of the modules. Moreover, since we also want to express the fact that certain nets should be kept short the following mutual proximity measure seems to be appropriate:

$$c_{ij} = \frac{\Sigma \{ w_k \mid p_{ik} = 1 \wedge p_{jk} = 1 \}}{\Sigma \{ w_k \mid p_{ik} = 1 \vee p_{jk} = 1 \}} \qquad (13)$$

Clearly, proximity thus defined is a number in the closed interval [0,1]. The *dutch metric*, defined by

$$\mathbf{D}^{\overline{2}} = \mathbf{J}_{mm} - \mathbf{C} =$$

$$\mathbf{J}_{mm} - (\mathbf{P\,I(w)\,P}^T)\ \overline{/}\ ((\mathbf{w}^T\mathbf{j}_n)\mathbf{J}_{mm} - \mathbf{F\,I(w)\,F}^T) \qquad (14)$$

where $\mathbf{F} = \mathbf{J}_{mn} - \mathbf{P}$, the complement of **P**, satisfies the requirements mentioned, and yields a distance space with positive semidefinite Schoenberg matrix. To prove the latter assertion, note that **Z** is a symmetric, positive semidefinite matrix and $\mathbf{ZJ} = \mathbf{JZ} = \mathbf{0}_{mm}$. So it is sufficient to show that **C** is positive semidefinite. Replacing the elements of **C** by geometric series yields

$$\mathbf{C} = \qquad (15)$$

$$\frac{1}{\mathbf{W}}\mathbf{PI(w)P}^T\ \overline{\times}\{\mathbf{J}_{mm} + \frac{1}{\mathbf{W}}\mathbf{FI(w)F}^T + \frac{1}{\mathbf{W}^2}(\mathbf{FI(w)F}^T)^{\overline{2}} + \ldots.\}$$

where $\mathrm{W} = \mathbf{w}^T\mathbf{j}_n$. Since

$$\mathbf{P\,I(w)\,P}^T = (\mathbf{P\,I}(\mathbf{w}^{\overline{1/2}}))\ (\mathbf{P\,I}(\mathbf{w}^{\overline{1/2}}))^T$$

and

$$\mathbf{F\,I(w)\,F}^T = (\mathbf{F\,I}(\mathbf{w}^{\overline{1/2}}))\ (\mathbf{F\,I}(\mathbf{w}^{\overline{1/2}}))^T$$

clearly are positive semidefinite, and elementwise multiplication and summation of positive semidefinite matrices results in positive semidefinite matrices, we may conclude that **C** is positive semidefinite.

## 5. THE STRUCTURE TREE

The fact that slicing simplifies area allocation problems is known for quite some time. Gilmore and Gomory applied it in the multistage cutting stock problem [3] (they characterized it by "guillotine cuts"). Several layout designers have recognized the benefits of that structural restraint [8]. The floorplans generated with the methods described in this paper will be consistent with the slicing principle (figure 1). The methods themselves profit greatly from the features of this structure, which emphasizes the efficacy of the concept.

By the procedure described in section 3 we obtain a two-dimensional configuration of points of which the coordinates are the corresponding components of the two eigenvectors spanning the space. The points represent the modules of the highest level of the system. The directions of slicing lines will be parallel to these vectors, starting with lines parallel to the eigenvector associated with the second largest eigenvalue. For each individual slice its child slices will be determined by a shrinking process weighting each potential slicing line, followed by deformation calculations to obtain a lower bound on the weights of the lines that will be accepted. The results are stored in a concise and convenient data structure, called the *shorthand tree*. We assume that only the minimum amount of information is available for each module, i.e. a rough estimate of the area it will occupy on the chip. Further, we assume that all modules are flexible, but shapes close to square are to be preferred. The procedure described does not differ much from the one for varying degrees of flexibility and more precise module data, provided that the orientation of modules is known. We will come back to these generalizations in section 7.

### 5.1 The shrinking process

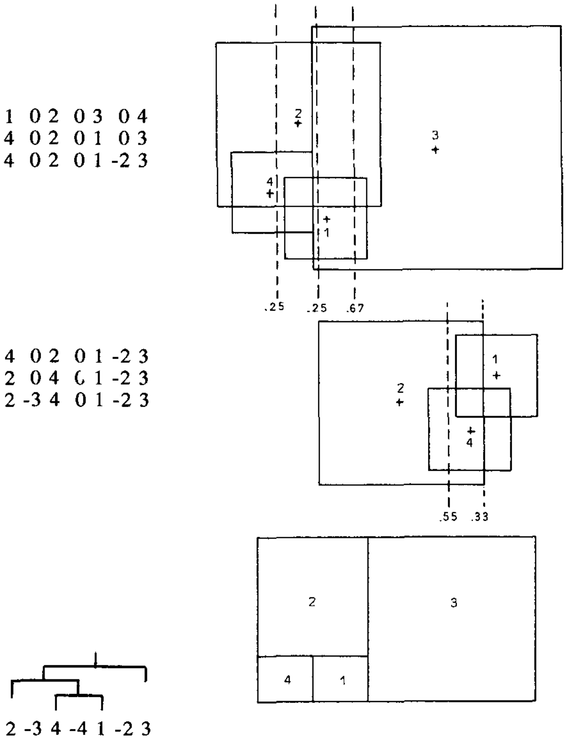


**Figure 2. Determination of the shrink factors.**
**In this small example the squares around the module centers are drawn in their position before the shrinking process. The amounts of shrinking necessary to have the vertical dashed lines intersection free are given for each line. After accepting $\ell_3$ the three modules at the left are treated in the same way. The consequences for the shorthand tree are given for each step. The final tree and configuration complete the picture.**

A given slice $\mathscr{S}$ with $s \geq 2$ modules is to be sliced in a number of child slices. $\mathscr{S}$ is separated from its sibling slices by lines parallel to the X-axis, so slicing must now be tried parallel to Y-axis. We first give - supported by figure 2 - a visualisation of a process determining the *shrink factor* of every possible *slicing line*. Then we will define that factor more precisely and more faithful to the actual computational process.

Think of a module as a square around the point representing it and with its sides parallel to the x- and y-axis. The sizes of these squares are such that each pair of squares has overlap and the ratio of their areas is equal to the ratio of the area estimates given to the corresponding modules. The squares are simultaneously shrunk leaving their centers fixed and preserving the area ratios. At some point during this shrinking process there will be a line that divides the modules in two blocks without intersecting any of the squares. The amount of shrinking necessary to reach that point is the *shrink factor* to be assigned to the corresponding line. Shrinking is continued and another line perpendicular to the x-axis and separating other blocks will be intersection free. Again, the corresponding factor will be assigned to that line. The process is continued until all s-1 factors are assigned. The lines with the highest shrink factors are accepted as slicing lines. The threshold is determined by the deformation (defined in section 5.2) the modules between consecutive slicing lines certainly will incur. The resulting blocks form the child slices, each of which will be treated in the same way if it contains more than 1 module.

More precisely:
The modules in $\mathscr{S}$ are ordered according to their x-coordinate. Let $\pi(M)$ be the rank number of $M \in \mathscr{S}$ in that order. $x_{\pi(M)}$ be its x-coordinate and $\alpha(M)$ be its area estimate. There are in general s-1 possible slicing lines (the only exception occurs when modules have the same coordinate; a slicing line separating those modules gets shrink factor 0). Slicing line $\ell_i$ will separate the modules in $\mathscr{S}_i^- = \{\pi^{-1}(j) \mid j \leq i\}$ from those in $\mathscr{S}_i^+ = \{\pi^{-1}(j) \mid j > i\}$. The shrink factor assigned to $\ell_i$ is

$$F_i = \mathrm{MIN}\left\{ f \mid \forall_{x \in [x_i, x_{i+1}]} \exists_{1 \leq j \leq s} \left[ x_j - \frac{1}{2} f \sqrt{\alpha(\pi^{-1}(j))} < x < x_j + \frac{1}{2} f \sqrt{\alpha(\pi^{-1}(j))} \right] \right\} \tag{16}$$

The computation of $F_i$ is quite simple. We start with an $f^0$ large enough such that

$$\bigcup_{j=1}^{s} \left( x_j - \frac{1}{2} f^0 \sqrt{\alpha(\pi^{-1}(j))}\ ,\ x_j + \frac{1}{2} f^0 \sqrt{\alpha(\pi^{-1}(j))} \right) \tag{17}$$

is an interval. At each iteration step $h_p$ and $k_p$ are determined by

$$x_{h_p} + \frac{1}{2} f^{p-1} \sqrt{\alpha(\pi^{-1}(h_p))} = \mathrm{MAX}\left\{ x_j + \frac{1}{2} f^{p-1} \sqrt{\alpha(\pi^{-1}(j))} \mid \pi^{-1}(j) \in \mathscr{S}_i^- \right\} \tag{18a}$$

$$x_{k_p} - \frac{1}{2} f^{p-1} \sqrt{\alpha(\pi^{-1}(k_p))} = \mathrm{MIN}\left\{ x_j - \frac{1}{2} f^{p-1} \sqrt{\alpha(\pi^{-1}(j))} \mid \pi^{-1}(j) \in \mathscr{S}_i^+ \right\} \tag{18b}$$

The value of $f^p$ is set to

$$f^p := \frac{1}{2} f^{p-1} \frac{\sqrt{\alpha(\pi^{-1}(h_p))} + \sqrt{\alpha(\pi^{-1}(k_p))}}{x_{k_p} - x_{h_p}} \tag{19}$$

The process is continued until $f^p = f^{p-1} =: F_i$.

Note that modules that have points at a relatively short distance will be separated only by lines with a high shrink factor F. This means that they will be together in many nested slices. Modules with points relatively far from each other will be separated after a few slicings. The relative positions of the modules in the final result will be quite accurately the same as the relative positions of the corresponding points in the Schoenberg construction.

### 5.2 Deformation

The preferred shape for flexible modules is a square. It is, however, in general not possible to accommodate the modules of a set $\mathscr{B}$ as squares in a slice with area

$$\Sigma(\mathscr{B}) = \sum_{M \in \mathscr{B}} \alpha(M)\ ,$$

to say nothing of such a slice with given dimensions, $l_1$ and $l_2$. To assess the deviation from square form of the modules in $\mathscr{B}$ we define the deformation as follows:

$$\mathrm{def}(\mathscr{B}, l_1, l_2) = \sum_{M \in \mathscr{A}} \left( \frac{\sqrt{\alpha(M)}}{\mathrm{MIN}\{l_1, l_2\}} + \frac{\mathrm{MIN}\{l_1, l_2\}}{\sqrt{\alpha(M)}} - 2 \right) + \tag{20}$$
$$+\ \mathrm{def}\left(\mathscr{B} \setminus \mathscr{A}, \mathrm{MAX}\{l_1, l_2\} - \frac{\Sigma(\mathscr{A})}{\mathrm{MIN}\{l_1, l_2\}}, \mathrm{MIN}\{l_1, l_2\}\right)$$

where $\mathscr{A} = \{M \in \mathscr{B} \mid \alpha(M) > (\mathrm{MIN}\{l_1, l_2\})^2\}$. When $\mathscr{A} = \emptyset$ the deformation will be zero. This does not mean that all modules of $\mathscr{B}$ fit into an $l_1 \times l_2$ slice as non-overlapping squares. It only is an indication that the modules are small in comparison with that slice, and a reasonable packing can be expected.

To use this definition of deformation we must be able to calculate the dimensions of the slices in a partial structure tree. With the areas for the leaves given, this is possible if we know, for example, the aspect ratio of the chip. We therefore either accept a user defined value, or set it by default to the ratio between the two largest eigenvalues of the Schoenberg matrix. The longest side is always chosen to be parallel with the eigenvector of the largest eigenvalue.

The calculation of the dimensions of all slices starts with deriving the outer dimensions from the total area and the aspect ratio. One of these dimensions, the shortest, is inherited by the child slices of the ancestor. The other dimensions of these child slices are obtained by dividing the sum of the areas of the modules in each of them by the inherited dimension. Proceeding in this way, each slice inheriting the dimension calculated for its father slice, will yield the dimensions of all slices.

Since the computation is performed in a top-down fashion, it can also be applied to partial structure trees. So, after the shrinking process we know the dimensions $l_1^{\spadesuit}$ and $l_2^{\spadesuit}$ of the slice accommodating the modules in $\mathscr{S}$ and the shrink factors of all possible slicing lines. Accepting the lines with the c-1 highest shrink factors would yield c child slices $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_c$. The deformation associated with this selection of lines will be

$$\mathrm{DEF}(c) = \sum_{i=1}^{c} \mathrm{def}\left(\mathscr{C}_i, l_1^{\spadesuit}, \frac{\Sigma(\mathscr{C}_i)}{\Sigma(\mathscr{S})} l_2^{\spadesuit}\right) \tag{21}$$

when $l_1^{\spadesuit}$ is the dimension to be inherited.

Though the deformation threshold for acceptance can be chosen freely, experience up to now only exists for a threshold of 0, which corresponds with a very cautious slicing process. Whether the resulting deformation equals 0, can be determined without performing the calculations of (20) and (21). However, when even for c=1 there is nonzero deformation, the deformation of such a binary slicing must be calculated in order to be compared with the deformation of a binary slicing in the perpendicular direction. Whenever the former value is higher, no line is accepted at the current stage, otherwise the line with the highest shrink factor is accepted in spite of the deformation caused.

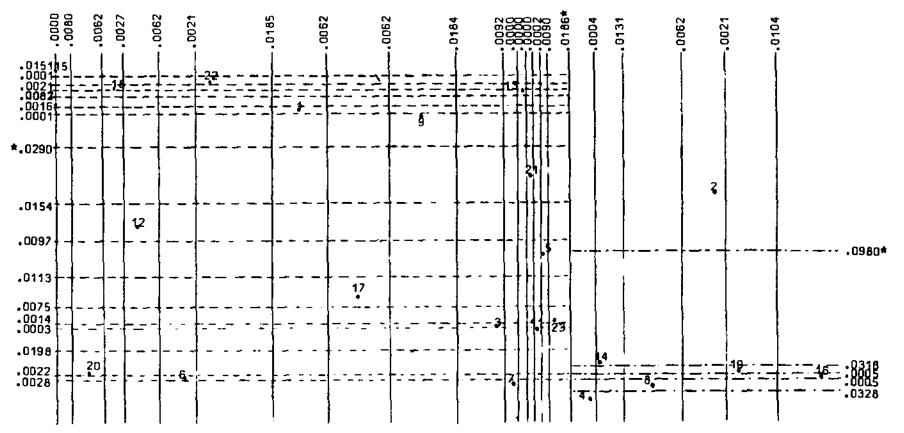
### 5.3 The shorthand tree

The shorthand tree is introduced for storing partial structure trees. It is a one-dimensional array with 2m-1 integers. The modules are represented as positive integers. After completion of the slicing structure these positive integers will be separated from each other by m-1 nonpositive integers representing the slicing lines. The values of these nonpositive integers indicate the level of slicing.

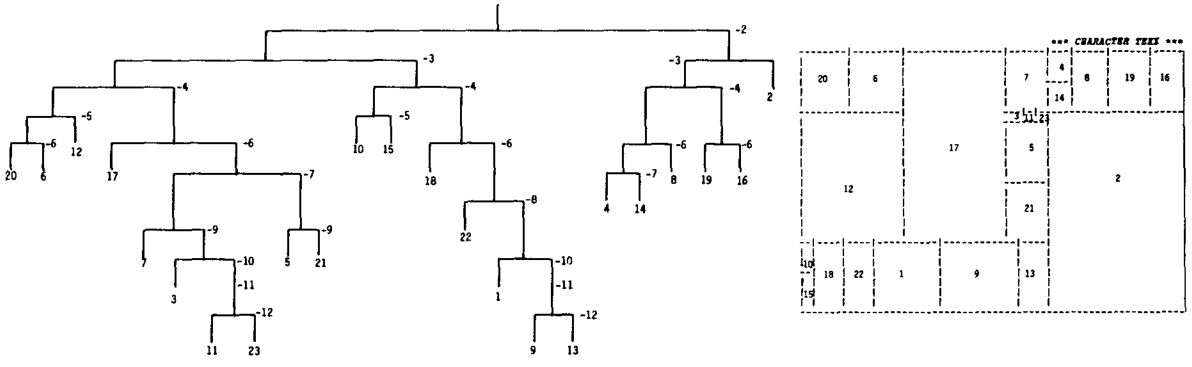
Initially all m-1 positions for the slicing indicators are filled with zeros. The positive integers are ordered according to the components of the eigenvector associated with the largest eigenvalue of the Schoenberg matrix. The procedures described in section 5.1 and 5.2 determine the first set of slicing lines to be accepted. At the corresponding positions in the array the zeros are replaced by a -2 (0 and -1 are reserved for separating eventual bonding pad macros).

In each of the subsequent steps the groups of (two or more) positive integers separated by negative integers are treated in a similar way. They are ordered according to the coordinates not used in the preceding step, and the positions corresponding to the accepted slicing lines are filled with an integer 1 lower than the indicator of the preceding step. The algorithm continues until all zeros in the shorthand tree are replaced by negative integers.

| | positions | | area |
|---|---|---|---|
| 1 | -8.41 | 7.40 | 53.04 |
| 2 | 16.47 | 2.44 | 298.05 |
| 3 | 3.55 | -5.15 | 3.04 |
| 4 | 9.51 | -9.92 | 6.00 |
| 5 | 5.31 | -1.24 | 26.50 |
| 6 | -15.04 | -8.25 | 33.50 |
| 7 | 5.24 | -8.58 | 25.24 |
| 8 | 12.77 | -8.70 | 23.74 |
| 9 | -1.00 | 7.38 | 55.90 |
| 10 | -22.65 | 7.57 | 5.68 |
| 11 | 5.26 | -5.16 | 2.85 |
| 12 | -17.97 | .55 | 138.26 |
| 13 | 5.25 | 8.52 | 25.87 |
| 14 | 9.56 | -7.44 | 5.93 |
| 15 | -22.65 | 9.91 | 5.21 |
| 16 | 22.65 | -8.68 | 24.50 |
| 17 | -4.71 | -3.74 | 222.71 |
| 18 | -19.14 | 8.74 | 20.85 |
| 19 | 17.71 | -8.65 | 25.45 |
| 20 | -20.89 | -8.22 | 33.20 |
| 21 | 5.26 | 3.39 | 25.74 |
| 22 | -14.46 | 8.75 | 22.43 |
| 23 | 6.97 | -5.10 | 3.06 |



```
1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0 11 0 12 0 13 0 14 0 15 0 16 0 17 0 18 0 19 0 20 0 21 0 22 0 23
10 0 15 0 20 0 18 0 12 0 6 0 22 0 1 0 17 0 9 0 3 0 7 0 13 0 21 0 11 0 5 0 23 -2 4 0 14 0 8 0 2 0 19 0 16
7 0 6 0 20 0 11 0 3 0 23 0 17 0 5 0 12 0 21 -3 9 0 1 0 10 0 13 0 18 0 22 0 15 -2 4 0 14 0 8 0 2 0 19 0 16
7 0 6 0 20 0 11 0 3 0 23 0 17 0 5 0 12 0 21 -3 9 0 1 0 10 0 13 0 18 0 22 0 15 -2 4 0 8 0 16 0 19 0 14 -3 2
```



```
20 -6 6 -5 12 -4 17 -6 7 -9 3 -10 11 -12 23 -7 5 -9 21 -3 10 -5 15 -4 18 -6 22 -8 1 -10 9 -12 13 -2 4 -7 14 -6 8 -4 19 -6 16 -3 2

20 -6 6 -5 12 -4 17 -4 7 -5 3 -6 11 - 6 23 -5 5 -5 21 -3 10 -5 15 -4 18 -4 22 -4 1 - 4 9 -4 13 -2 4 -5 14 -4 8 -4 19 -4 16 -3 2
```

**Figure 3. An illustration of the slicing process. In the table the position of the module points and module areas are given. With that information the shrink factors have to be determined. The first three steps (the first two levels) of the slicing process are given. The accepted lines are indicated by an asterisk. In the shorthand tree they are represented by -2 and -3. (Note the reordering of the positive integers.) The final result is given in the lower part of the figure. The fact that the outdegree of the vertices never exceeds 2 is a consequence of the very cautious way of allowing deformation. Whenever no slicing occurs at an intermediate level the tree can be reduced by having a multiway slicing at the previous level. This reduces the number of levels from 12 to 6. The shorthand of the tree depicted and its reduced form are given.**

## 6. THE WIRING SPACE

After establishing the structure tree the floorplan is topologically fixed. The vertices of the tree are the slices in the structure. The leaves represent the modules. The slicing lines represent channels which must accommodate the global nets. Estimating the size of these channels is a highly technology and style dependent task, and therefore not within the scope of this paper. With the information obtained by the procedures of the previous section fast algorithms with several degrees of accuracy are feasible in many cases. (The author's experience is with two-layer channel routing.) Because of the nature of the problem most of these algorithms will deliver estimates for the channel widths, not for the channel areas. As a consequence the simple top-down area distribution method, described in section 5.2, is not applicable. We therefore have to derive the geometrical constraints imposed by the structure, represented by the structure tree, and the minimum areas of the modules.

Obviously, the shorthand tree is an ideal data structure for storing the partial structure tree during the slicing process. For tree traversal, however, it is not adequate at all. Therefore, before establishing a set of equations expressing the relations between slice dimensions, the shorthand tree is translated into a traditional triply linked tree in which each vertex has a pointer to its father, its primogenitive and the next sibling. (Simultaneously, the junction cells [8], which are the channels, can be added to the tree. There is a junction cell between each pair of siblings, or, equivalently, for each nonpositive integer in the shorthand tree, or, equivalently, for each slicing line in the structure with zero channel width.)

For each compound slice a set of linear equations can be written (figure 4):

$$\begin{array}{llllllll} X & -y_1 & -y_2 & \dots & \dots & -y_s & & = \sum_{i=1}^{s-1} b_{i,i+1} \\ & -x_1 & & & & & +Y = & b_1 \\ & & -x_2 & & & & +Y = & b_2 \\ & & & \dots & & & \dots & \\ & & & & \dots & & \dots & \\ & & & & & -x_s & +Y = & b_s \end{array} \qquad (22)$$

Each vertex gives rise to as many equations as its degree in the tree. The inner vertices, representing the compound slices, lead to the linear equations of (22). For each leaf, representing a module, a nonlinear equation, fixing the area of the module, enters the set. Together these equations are the constraints the final geometrical structure has to satisfy. To find a solution that is optimal in some sense, for example the smallest total area, a suitable objective function to be optimized under these constraints, has to be defined. The dimensions resulting from the simple area distribution procedure of section 5.2 with zero channel widths can be used as a starting point. It is the solution of the system with one extra constraint, e.g. the aspect ratio, where the compound equations have a zero right hand side. The left hand side is unchanged. By ordering equations and variables in a special way the zero-nonzero entry structure of the jacobian becomes almost upper triangular (figure 5). However, at least as many subdiagonals remain as the number of slicing levels exceeds 2.

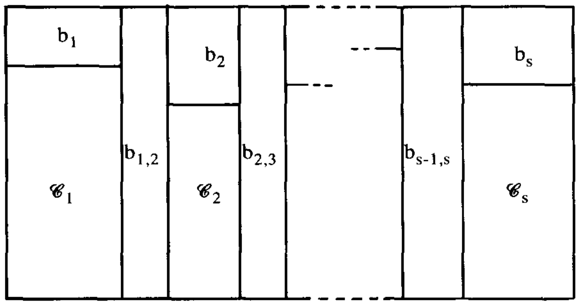


**Figure 4. Definition of the variables in the compound equations (22). $b_{i,i+1}$ is the channel width; $b_i$ accounts for the nets that cross $\mathscr{C}_i$.**

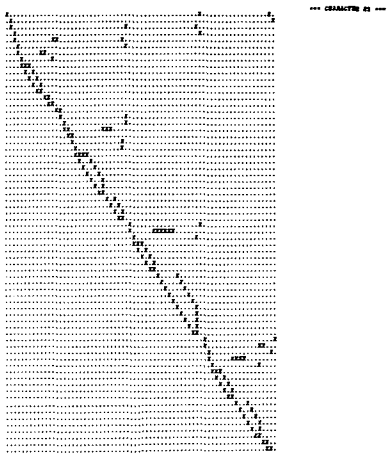


**Figure 5. The zero-nonzero entry structure of the jacobian for the example of figure 3.**

Yet, given the Y-dimension of the ancestor, all slice dimensions can be calculated in that special order by calling the following recursive procedure with the ancestor $\mathscr{M}$ and the given dimension as arguments.

```
detx($\mathscr{S}$,Y);
begin if $\mathscr{S}$ is not a leaf then
    begin detx:=0; for i:=1 to s do
        begin  $x_i$:=Y-$b_i$;
               if $x_i \leq 0$ then no solution;
               $y_i$:=detx($\mathscr{C}_i$,$x_i$);
               detx:=detx+$y_i$+$b_{i,i+1}$    {$b_{s,s+1}$=0}
        end;
    end
    else detx:=$\alpha(\mathscr{S})$/Y
end.
```

Either the other dimension of the ancestor will be returned, and all descendant slice dimensions assigned, or there exists no solution for the given Y-dimension. So, a simple one-dimensional direct-search for the minimum of $Y_{\mathscr{m}} \times \text{detx}(\mathscr{M}, Y_{\mathscr{m}})$ is probably the most efficient method, in case the total area is to be minimized.

## 7. VARYING DEGREES OF FLEXIBILITY

In the preceding sections we presented a complete solution to the macro placement problem as described in the introduction. We also showed that the algorithms can be combined with channel width prediction routines. In [5] a similar problem is treated, but the dimensions of the modules are assumed to be fixed. Yet, in the initial placement stage the shape of the modules is ignored, and only module areas are taken into account. In fact, this stage is equivalent to a sequence of binary slicings, each slicing chosen such that the number of nets connecting modules from different blocks is small, and the difference between the total areas of the two blocks is not too big. Because of their tendency to shift complexity to lower level blocks, these socalled min-cut algorithms that try to keep the number of nets connecting high-level blocks small, are declining in many vlsi-design environments. The approach based on Schoenberg constructions is more global in the sense that it considers the whole connectivity structure at once. Since it is also considerably faster[3], one might prefer to generate the configuration in the plane by the Schoenberg construction, and to build a two-dimensional tree [1] with the coordinates as keys, each time

[3] Even an APL implementation took for the Schoenberg construction (accuracy $10^{-6}$) only 22.7 seconds for a circuit with 162 modules, 237 nets and 627 pins. The calculation of **D** with (14) with full outer product operation (i.e. no sparsity savings) took 52.7 seconds.

separating on the basis of approximately equal area division. This tree, a binary slicing tree, can be easily translated into a (necessarily series-parallel) polar graph. Then, placement improvement as in [5] can reduce the amount of dead area. Binarity is not essential for these procedures, so a structure derived by the methods of section 5 is also suitable as an initial placement routine preceding the improvement stage. This is, however, not a satisfactory approach. Firstly, because the true shapes of the modules are not taken into account when deriving the initial structure. Secondly, a placement improvement as in [5] destroys the desirable slicing structure [8]. And thirdly, that improvement cannot handle a mixture of modules with flexible and less flexible shapes.

Of course, the shrinking process of section 5.1 can be performed with aspect ratios different from 1, if the orientations of the modules with respect to the two eigenvectors spanning the plane are known. So we have to deal with the problem of assigning good orientations to some modules after the Schoenberg configuration has been generated. Modules with one or both dimensions fixed usually are predefined cells of which the layout has to be inserted into the floorplan of the chip. Completely specified cells are therefore called *inset cells* [8]. Sometimes stretching in one direction is allowed to provide for pitch matching. In that case we speak of *stretch cells*. They have one dimension fixed and a minimum area. For both classes of cells the distribution of the pins over the sides is known. Of course, for each such a module side we can derive a characteristic vector $\mathbf{v}_n$ indicating by a component 1 whether it has a pin connected to the corresponding net, and by a component 0 the absence of such a pin. We also can calculate the dutch distance of $\mathbf{v}$ to all rows of $\mathbf{P}$. Now, suppose we add a point to the r-dimensional Schoenberg configuration having exactly those distances to the corresponding module points (at most one extra dimension is needed for this addition), which position would the projected point have in the two-dimensional configuration? To save space we omit the derivation of the answer which is

$$\left( \frac{\sum_{i=1}^{m} q_{i1}(s_{ii}-d_i^2)}{2\lambda_1} , \frac{\sum_{i=1}^{m} q_{i2}(s_{ii}-d_i^2)}{2\lambda_2} \right) \tag{23}$$

where $s_{ii}$ is the i-th diagonal element of the Schoenberg matrix $\mathbf{S}$, $\lambda_1$ and $\lambda_2$ the largest eigenvalues of $\mathbf{S}$, and $d_i$ is the dutch distance of $\mathbf{v}$ to $\mathbf{p}_{i.}$. Thus, two simple calculations give the position at which the module side "would like to be" if it were separable from its module. The vector obtained by subtracting the module point coordinates from the result of (23), can be interpreted as a force vector. Such a vector can be determined for each side. If we think of the module as a rectangle with minimum area, fixed position of its center, and the four "forces" applied to the centers of the respective sides, an amount of potential energy is associated with each of the eight possible orientations. The one with minimum energy is selected.

## 8. BONDING PAD MACROS

The positions of the bonding pads along the perifery of the chip are often prescribed. Ignoring these constraints might lead to lower quality layouts. In order to take them into account we consider the bonding pads to be placed on the same side of the chip as a *bonding pad macro* which is treated as a module during the Schoenberg construction. Consequently, the netlist is extended with four rows representing the bonding pad macros. The nonzero entries in these rows indicate the nets connected to a bonding pad in the corresponding macro. These four macros will be represented by four distinct points, $\mathbf{q}_{n.}$, $\mathbf{q}_{s.}$, $\mathbf{q}_{e.}$ and $\mathbf{q}_{w.}$, in the higher-dimensional configuration realizing the distances obtained by using the dutch metric formula (14). Now, instead of projecting the configuration into the plane spanned by the eigenvectors associated with the largest eigenvalues of the Schoenberg matrix, we might project it into the plane spanned by the two vectors $\mathbf{q}_{n.}$-$\mathbf{q}_{s.}$ and $\mathbf{q}_{e.}$-$\mathbf{q}_{w.}$, which are the difference vectors of the points representing opposite bonding pad macros.

From experience we should learn when to use the plane spanned by the difference vectors, and when to stick to the eigenvectors. In the decision the following pairs of values should be considered: the two largest eigenvalues $\lambda_1$ and $\lambda_2$ of the Schoenberg matrix derived from (14), the two eigenvalues $\mu_1$ and $\mu_2$ of the Schoenberg matrix derived from the distances after projection into the plane of the two difference vectors, the lengths of the two difference vectors, and the angle between the two difference vectors. When $\lambda_1\lambda_2$ is considerably larger than $\mu_1\mu_2$ and/or $(\mathbf{q}_{n.}$-$\mathbf{q}_{s.})(\mathbf{q}_{e.}$-$\mathbf{q}_{w.})$, the original "eigenvector" plane should be chosen, since off-chip connections should not be critical in the first place, and apparently there is no clearly better projection plane in that case as far as the positions of the bonding pad macros are concerned.

## 9. CONCLUSION

The algorithms presented in this paper are very fast which make them suitable tools in early interactive floorplan design. The vector w consists of a convenient set of parameters for trying to eliminate net length violations. Global routers can be used to estimate these net lengths, but also to locate congested channels which is important for technologies with fixed channel capacities. (The author uses a breadth-first algorithm on the dual of the graph isomorphic to the line configuration resulting from the procedures of section 3, 4 and 5). Also pin allocation problems at the periphery of a module can be detected and avoided by adding constraints.

The methods of this paper are also consistent with the principles of structured layout design as laid down by the author in [7] and [8]. They are therefore also useful in design systems with a higher degree of automation. With a suitable functional hierarchy and an adequate initial characterization of the modules represented in that hierarchy required as input, these methods combined with technology and style dependent algorithms for filling blank cells and junction cells are the ingredients of a Structured Layout Implementation for Custom Environment (SLICE).

To complete the picture some limitations and drawbacks. The Schoenberg construction tries to embed a distance space by interpreting its distances as euclidean distances. In layout design the minkowski-1-metric is a more adequate metric. An equivalent of the Schoenberg theorem for this metric, however, is not known, and much more difficult to obtain. Of course, we can specify an objective function based on the minkowski-1-metric, and optimize that function starting from the obtained projection of the Schoenberg configuration, but computation time and storage needed would be considerably increased.

Some analysis preceding application of the Schoenberg construction seems recommendable, when there is a big range in module areas. The dutch metric compensates in a sense for heavily connected, and thus relatively big modules, but whenever a module will have dimensions close to the chip dimensions to be expected, the modules should not be treated on an equal footing.

**References**

[1] J.L. Bentley, *Multidimensional binary search trees used for associative searching* , Comm. of the ACM, vol. 18 - nr. 9, pp. 509-517, 1975.

[2] C. Eckart & G. Young, *The approximation of one matrix by another of lower rank* , Psychometrika, vol. 1, pp. 211-218, 1936.

[3] P.C. Gilmore & R.E. Gomory, *Multistage cutting stock problems of two and more dimensions* , Oper.Res., vol. 13, pp. 94-120, 1963.

[4] W.R. Heller, *An algorithm for chip planning* , Caltech Silicon Structures Project Memo 2806, 1979.

[5] U. Lauther, *A min-cut placement algorithm for general cell assemblies based on a graph representation* ,J. of Dig.Syst., vol. 4 - nr. 1, pp. 21-34, 1980.

[6] K. Menger, *Untersuchungen über allgemeine Metrik* , Math.Ann., vol. 100, pp. 75-163, 1928.

[7] R.H.J.M. Otten, *Complexity and diversity in ic layout design* , Proc. of ICCC, pp. 764-767, 1980.

[8] R.H.J.M. Otten, *On vlsi layout design automation* , **Hardware and software concepts in vlsi** (ed. G. Rabbat), Van Nostrand Reinhold Co., 1982.

[9] I.J. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert"* , Ann. of Math., vol. 36, pp. 724-732, 1935.