1. INTRODUCTION

Row style, especially for more complex CMOS circuits, has the advantage of the two dimensional style over the single layer style of the same circuits. This comparison demonstrates how flexible the two dimensional style is when compared to the generalized multilayer row style. We ran from simple CMOS circuits to complex ones. We illustrate our algorithm through a set of examples.

Abstract: Most standard cell generators for CMOS circuits run the internal connections. These wires are placed between and on top of the two diffusion runs horizontally along each side of the row, vertically aligned at the center of their gates. Diffusion runs horizontally along each side of the row, whereas diffusion points belong to linear arrangements (or a row) of transistor pairs. Each pair consists of p- and n-type transistors, which are vertically aligned at the center of their gates. Research focused on the layout style proposed by Uher and Wardeman [18].

2. Dept. of Electrical and Computer Engineering, Univ. of Mass. Amherst, MA 01003
3. IBM Israel Scientific Center, Technion City, Haifa 22000, Israel

Jack A. Feldman and Israel Koren

On Generating Two Dimensional CMOS Cells
The rows are arranged back to back. Power and ground buses are implemented in metal 1 layer by our algorithm, as illustrated in Figure 1. The layout has alternating n and p transistors and the gates with power and ground are separated. The leakage in the large N-channel MOSFETs is a serious problem for each transistor. The type (p or n) of the connecting to the gate, drain and source ports (topological description) are described in terms of their constituent transistors. Our algorithm generates automatically the symbolic layout of a CMOS cell starting from its topological description.

2. IMAGE DEFINITION

In Section 2 we describe the layout image. Section 2 describes the algorithms for the layout generation process. However, no algorithmic details are provided in these papers. Applying a set of appropriate rules, however, no algorithmic details are provided in these papers.

The initial placement and routing are improved by etching of two dimensional layouts. These problems are usually done by the above mentioned algorithms. Our algorithm in one dimension is usually done by the above mentioned algorithms. Our algorithm in two dimensions is usually done by the above mentioned algorithms. Our algorithm in three dimensions is usually done by the above mentioned algorithms. Our algorithm in four dimensions is usually done by the above mentioned algorithms. Our algorithm in five dimensions is usually done by the above mentioned algorithms. Our algorithm in six dimensions is usually done by the above mentioned algorithms. Our algorithm in seven dimensions is usually done by the above mentioned algorithms.

In [9], the authors propose an automated system that generates a multi-row layout from the circuit description at the transistor level. Their experiments show that two of three algorithms generate the given large circuit.

The need for exploring the second dimension led to the development of strategies for generating layouts with multiple rows. Two approaches are employed: (1) create a single row layout for the given large circuit and interconnect the basic circuits laid out in the single row style; their algorithm uses the circuit description at the transistor level. Their experiments show that two of three algorithms generate the given large circuit. The algorithms proposed in [13], [15], [18], and [19] search for the optimal solution.
3.1 Transistor Assignment

Transistor orientation step, respectively.

To determine a partitioning of the transistor assignment step and then the large solution space which is \( O(n^2) \), \( n \) is the number of transistors. By two

The placement problem is solved in two steps. In the first step, transistors are assigned to locations on the placement grid. In the second step, the placement orientation is determined. Partitioning the transistor assignment into two well-defined steps replaces the

3. Placement

These two parameters completely define the placement grid for a given circuit.

Transistors can be placed in the first row are parameters of the placement program. The number of rows and the type of rows: 2. If rows are reserved for N-type transistors, 4 if rows are reserved for P-type transistors, and 8 if rows are reserved for P-N transistor rows. The placement grid corresponds to a back to back arrangement of P-N transistors.

The placement grid is an \( n \times m \) matrix, where \( n \) corresponds to a location where a transistor can be placed, \( m \) is the number of rows, and \( m \) is the number of columns. All locations can be placed, 

in this manner. The matrix is used to accomplish internal routing. The transistors are located along different horizontal channels (see Figure 1).

The layout that meets imposed dimensions, not necessary the minimum ones, is required (this occurs when constructing a library of

Situation:

a layout to generate layouts with different aspect ratios, which is useful in the following

and between consecutive transistor rows of type P and type N, respectively.
The problem of assigning transistors to grid locations can be formulated as the known quadratic assignment problem, which is stated as follows: given a set $E = (1, 2, \ldots, e)$ of elements and a set $L = (1, 2, \ldots, l)$ of locations, find a one-to-one mapping from set $E$ into set $L$.

### 3.1.2. Mathematical Formulation

1. In the vertical direction - maximize the number of polysilicon subunits.
2. In the horizontal direction - maximize the number of diffusion subunits.

Considered for the assignment algorithm are:

- The channel introduced between the adjacent transistor rows. In summary, the objective is to minimize or even eliminate the necessary connections that are accomplished in adjacent cells.
- The space introduced between adjacent transistors. While not shown in Figure 2 illustrating two different permutations of these transistors, with shrinking and non-shrinking diffusion port directions, in case (2), the space between the adjacent transistors forms a straight line. In Figure 3, the connection between adjacent gates is accomplished by a straight line.
- The space introduced between adjacent transistors. While not shown in Figure 2 illustrating two different permutations of these transistors, with shrinking and non-shrinking diffusion port directions, in case (2), the space between the adjacent transistors forms a straight line.

Another factor that arises in the routing and contributes to the area reduction is related to

- Transistor lengths and orientations compared to the configuration illustrated in Figure 2.

- One or two contacts are saved, or if the corresponding net is a multi-port or a two-port.
- The connection between the adjacent drain/source ports is implemented directly in all cases.
- No space has to be introduced between consecutive transistors.
- The reduction due to drain/source abutment results from the following:

- A significant reduction in area results if transistors placed in consecutive locations form a

### 3.1.1. Objectives

The main goal when assigning transistors to locations is to...

- The definition of the objective functions and the solution technique used are described in the following paragraphs.
We label the locations in the placement grid from \( \mathcal{T} \) to \( \mathcal{M} \) with the conventions that the rows of the matrix are scanned from left to right and from top to bottom. Let \( X \) and \( Y \) be the sets of horizontal diffusion sources and horizontal diffusion sinks, respectively. If \( (a, b) \) and \( (c, d) \) are in the same set, then the element \( G_{ab} \) is 1, otherwise 0.

\[
\begin{cases}
G_{ab} = 1 & \text{if } (a, b) \text{ and } (c, d) \text{ have one diffusion port} \\
G_{ab} = 2 & \text{if } (a, b) \text{ and } (c, d) \text{ have both diffusion ports}
\end{cases}
\]

The first objective considered for the transistor assignment problem is to maximize the distance matrix and the distance matrix provides some metric on the placement grid.

The transistors that compose a given circuit compose the set \( \mathcal{E} \) and the \( n \times m \) locations set \( \mathcal{M} \). Then, the affinity matrix reflects the connectivity between the various occupations of the transistors.

Let \( \mathcal{E} \) be the set of all possible mappings of set \( \mathcal{E} \) into set \( \mathcal{M} \). Let \( \mathcal{E} \) be a particular mapping, and let \( (a, b) \) denote the location to which element \( i \) was assigned in the mapping. Let \( d_0 \) be the distance between location \( i \) and location \( j \). Let \( d_{ab} \) be the distance between element \( a \) and element \( b \). Let \( d_{ab} \) be the distance between location \( a \) and location \( b \).

A perpendicular matrix, and let \( d_{ab} \) be the distance between element \( a \) and element \( b \). Let \( d_{ab} \) be the distance between location \( a \) and location \( b \). Let \( d_{ab} \) be the distance between location \( a \) and location \( b \). Let \( d_{ab} \) be the distance between location \( a \) and location \( b \).

Then, the quadratic assignment problem can be stated as follows:

\[
\sum_{(a, b) \in \mathcal{E} \times \mathcal{M}} d_{ab} G_{ab} = \mathfrak{M} \tag{1}
\]

Then, the quadratic assignment problem can be stated as follows:

\[
\sum_{(a, b) \in \mathcal{E} \times \mathcal{M}} d_{ab} G_{ab} = \mathfrak{M} \tag{1}
\]

Consider now the functional:

\[
\mathfrak{F} = \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} \mathcal{P}_{ij}
\]

Where:

\[
\mathcal{P}_{ij} = \begin{cases}
\mathcal{P}_{ij} = 1 & \text{if } \mathcal{P}_{ij} \text{ be a } n \times m \text{ matrix} \\
\mathcal{P}_{ij} = 0 & \text{if } \mathcal{P}_{ij} \text{ be a } n \times m \text{ matrix}
\end{cases}
\]
3.1.4. Solution Technique

\[ \text{Function } h(d) \]

\[
\begin{align*}
\text{otherwise} & \quad 1 \leq |x - y| \quad x \neq y \\
& \quad 0 \\
& \quad \left| y_{dp} \right| = 1
\end{align*}
\]

\[
\begin{align*}
\text{otherwise} & \quad 0 \\
& \quad \left| y_{dp} \right| = 1
\end{align*}
\]

matrix \( y_{dp} \) is as follows:

The second objective considered for the transistor assignment problem is to minimize the number of vertical polylines attributed. We define the affinity matrix \( g_{ij} \) and the distance between diffusion ports of \( T \) and \( T' \). Since the distance between diffusion ports is the same for both diffusion ports \( T \) and \( T' \), and \( \text{min}(g_{ij}) \) we have no common diffusion port. According to these definitions, \( T \) and \( T' \) have one of two common diffusion ports and are denoted the \( x \) and \( y \) coordinates associated with location \( k \) (in the place).
Finally, we combine the two normalized objective values and attempt to minimize a single expression; i.e.,

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{ijkl} \cdot \min\{\psi_{ijkl}, \phi_{ijkl}\} = \psi_N
\]

and

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_{ijkl} \cdot \min\{\psi_{ijkl}, \phi_{ijkl}\} = \phi_N
\]

Normalization factors are given thereafter.

The sign and all local values are free. i.e., \( \psi = \Gamma \) and \( \phi = \Phi \). Symmetry we substitute \( \gamma \) by \( \alpha \) and \( \beta \) by \( \gamma \) in \( \psi \) to obtain \( \Gamma \). In order to normalize, rewrite expression for \( \psi \) by substituting \( \beta \) by \( \gamma \) and \( \gamma \) by \( \beta \) in \( \phi \). The expressions for \( \Gamma \) and \( \Phi \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices.

An important property of the scheme in [5] is that the enumeration of all the possible assignments is not required. For each feasible unassigned element-location pair (u, v) combination (we compute its mean value, and select that element-location pair (u, v) which minimizes the mean value. Then we compute the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices. Let \( \mu^{\psi} \) and \( \mu^{\phi} \) denote the mean value for the horizontal and vertical distance matrices.

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{(1 - \gamma - \mu)(\gamma - \mu)}{\Gamma} +
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{(1 - \mu)(\gamma)}{\mu} +
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{(1 - \gamma)(\mu)}{\mu} +
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{(1 - \gamma)(\mu)}{\mu} = n
\]

where...

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{u_{ijkl} - U}{U} = \psi_N
\]
mean value of the \( \langle x + 1 \rangle \)th partial permutation is given by the following expression:

element-location pair, where \( \langle x \rangle \) and \( \langle y \rangle \) are
then the contribution of the pair \( \langle x \rangle \) to the
mean value of the \( \langle y \rangle \)th partial permutation. Let \( \langle x \rangle \) and \( \langle y \rangle \) denote the currently selected
element-location pair and add it to the mean value of the \( \langle x \rangle \)th partial permutation. In step \( \langle x \rangle \), we update the contribution of the currently selected element-location pair and add it to the mean value of the \( \langle x \rangle \)th partial permutation. The mean value calculation can save \( O(n) \) computations. This is due to the following

3.1.6 Complexity of the Algorithm

location on the placement grid are of the same type.

elements that are feasible in the transistors and the
one having the smallest mean. An element-location pair is feasible if the transistor and the
mean value of the \( \langle x \rangle \)th partial permutation. We compute the corresponding mean value and select the
part of \( \langle y \rangle \) possible locations. We compute the corresponding mean value and select the
unassigned elements and \( \langle y \rangle \) unoccupied locations. For each feasible element-location
order \( \langle x \rangle \)th, in step \( \langle y \rangle \), a new element-location pair is selected out of the remaining \( \langle y \rangle \) unassigned elements and \( \langle y \rangle \) unoccupied locations. For each feasible element-location

3.1.5 Enumeration Scheme

Relative significance.

where \( w \) and \( w' \) are weights associated to the two objectives to allow us to control their

\[
\sum_{x=1}^{n} N_x w_x + \sum_{y=1}^{m} N_y w'_y = W
\]
3.2. Optimal Orientation of Transistors

Two objectives should be assigned a higher weight. The values \((x_1, x')\) and \((x, x')\) can be used as a guideline to determine which one of the two orientations are feasible. Let \(m = \frac{X_1}{x_1} + \frac{x'}{x}\), and \(d = \frac{X}{x_1} + \frac{x'}{x}\) be the number of potential orientations and the number of aligned gates that share the same net, respectively. Given circuit, set \(m = 1\) and \(d = 1\). And find the number \(x\) of aligned gates that share the same net.

Set \(m = 0\) and \(d = 0\) and find the number \(x\) of potential orientations for the given circuit.

An additional optimization for the best \(m, d\) ratio is possible but it does not justify the effort. However, the following heuristic strategy can be employed to select values for \(m, d\). Weights \(w_1, w_2\) are assigned to the above two objectives. Initially, \(w_1, w_2\) are selected to be two objectives. Initially it is clear that the ratio

3.1.7. Strategy for Determining the Weights

which requires \(O\left(\mu n\right)\) steps and consequently \(O\left(\mu^2 n\right)\) is the complexity of our search algo.

Empirically (11) the computation of the mean value requires \(O\left(\mu n\right)\) steps as opposed to (7) where \(\mu, \nu \in \mathbb{R}^+ \), \(\mathcal{S} \subseteq \mathbb{R}^n\) and \(y \in \mathbb{R}^n\):

\[
\left(\sum_{l \in \mathcal{S}} P_l^m p_l^m d_l^m \right) \sum_{s \in \mathcal{S}} S_l^{m_2} + \sum_{l \in \mathcal{S}} \sum_{s \in \mathcal{S}} (l^d d_p^m p_m^m d_{l}^m) + \sum_{l \in \mathcal{S}} \sum_{s \in \mathcal{S}} (l^d d_p^m p_m^m d_{l}^m) \right) + \sum_{l \in \mathcal{S}} \sum_{s \in \mathcal{S}} (l^d d_p^m p_m^m d_{l}^m) = \sum_{l \in \mathcal{S}} \sum_{s \in \mathcal{S}} (l^d d_p^m p_m^m d_{l}^m) + \sum_{l \in \mathcal{S}} \sum_{s \in \mathcal{S}} (l^d d_p^m p_m^m d_{l}^m)
\]
Property 2: Both cost measures are additive functions.

Property 1: The optimal orientation of the \( f \)-transistor column depends only on the \( f \)-transistor column.

This problem: The dynamic programming technique is applicable due to the following two properties of the technique in what follows to satisfy our requirements.

The algorithm was suggested for finding the optimal solution. This solution technique is computationally less expensive in [1], a dynamic-pro- the three and bound algorithm that explicitly enumerates the entire solution space, was pro- posed. This solution technique is computationally less expensive in [1], a dynamic-pro- the same row. The vertical alignment cost is defined as follows:

\[
\begin{align*}
0 & \quad \text{otherwise} \\
\begin{cases}
\alpha & \text{if in one of the adjacent diffusion ports}
\end{cases}
\end{align*}
\]

The horizontal alignment cost is defined as follows:

\[
\begin{align*}
0 & \quad \text{otherwise} \\
\begin{cases}
\beta & \text{if in their adjacent diffusion ports}
\end{cases}
\end{align*}
\]
4. ROUTING

The main difference between these two channel types is that those of horizontal channel, The main difference between these two channel types is that those of horizontal channel, while those of vertical channel are used to interconnect nets that lie in more internal to those channels. Vertical channels are used to interconnect ports which are
illustrated in Figure 1. The horizontal channels are used to interconnect ports which are
interconnected by various transistor ports. These wires will be routed through channels as

once the transistor placement and orientation steps have been completed, we need to

backtracking from \( f = 0 \) to \( f \) by \( \phi \) using the predecessor states.

where there is more than one such state, we break ties by \( \phi \). The final solution is obtained by

\[ \text{max} \{ \phi'(1), \phi'(2), \ldots, \phi'(N) \} \]

the ones for which \( \phi'(i) \) is maximum. For \( \phi = 0 \) we choose among the states \( f = \text{max} \{ \phi'(1), \phi'(2), \ldots, \phi'(N) \} \)

which are computed by equation (13) and (14) above and its predecessor \( \phi \).

For each state \( \phi' \) with costs \( C_1 \) and \( C_2 \), let \( \beta = \begin{cases} 0 & \text{if } C_1 < C_2 \\ 1 & \text{if } C_1 > C_2 \end{cases} \)

maximizes expression (12). In case this index is not unique, we choose among all the

indices \( \beta \) which maximize the expression in (12). Initially we have the states \( \phi' \),

let the predecessor to state \( \phi' \) (denoted by \( \phi'(0) \)) be \( \phi \) where \( \phi \) is the index which

where \( C_1 < C_2 \) is the best vertical diffusion alignment cost so far for state \( \phi' - 1 \).

expression is given by:

The second one relates the best vertical diffusion alignment cost for state \( \phi' \) is

where \( C_1 < C_2 \) is the best horizontal diffusion alignment cost so far for state \( \phi' - 1 \).

By:

\[ \text{max} \{ \phi'(1), \phi'(2), \ldots, \phi'(N) \} \]

where \( f \) is the cost of concatenating the transistors in columns \( f \) and \( f' \). This expression is given by:

The first one relates the best horizontal diffusion alignment state \( f \) belongs to column \( f \) and the value \( \phi \) equals the transistors orientation. For each

configuration \( \phi' \) is the cost of a sequence of \( \phi' \) as shown in Table 1. where each entry stands for a four-possible configurations for transistor orientations in a column, let state \( \phi' \) represent the number of columns from left to right, and column \( \phi \) represents the number of rows.

The algorithm scans the \([1 \times m]\) placement matrix where \( m \) represents the number of columns from left to right.
5. Results and Discussion

Table 1 illustrates the results obtained for the Full-Adder. It demonstrates the capability of generating layouts with different aspect ratios.

In this section, we demonstrate the advantages that result from generalizing two-dimensional layouts through a sample of circuits. These include a full adder, multiplier, decoder, 4-to-1 multiplexer, and a 2-to-1 multiplexer. The results reflect a more realistic layout of the design.

1. These steps are as follows:

   a. Determine the nets that can be placed between consecutive channels.

   b. Draw and mark the nets that can be placed in the vertical channels.

   c. Place the routing segments into the nets, which are executed sequentially.

   d. The overall strategy used for the routing algorithm is a left-to-right scan of the routing area.

2. Each net is assigned to the left or right vertical channel. Assignment of wires to tracks is accomplished by assigning the tracks to minimize their lengths.

3. Routing of the least vertical channel. Assignment of wires to tracks is accomplished by assigning the tracks to minimize their lengths.

4. Routing of the greedy channel. The algorithm used to decode the routing requirements is based on a greedy channel decoder.

5. Finally, routing of the right vertical channel is performed. Interchannel coloring is used again to assign wires to tracks in the right vertical channel.

6. Table 1 illustrates the results obtained for the Full-Adder. It demonstrates the capability of generating layouts with different aspect ratios.
6. CONCLUSIONS

cited on a 481 IBM computer. Table 2 shows the computation time required for these crs.

Our system was implemented in Pascal containing about 4000 lines, and has been ex-

represent the ultimate trend, which allow dynamic exchange of layout styles in the process of composing the chip, will impose requirements by the surrounding environment. It is our belief that CAD tools is the most important capability since it allows to select that layout style which best fits the However, the flexibility for generating layouts with different aspect ratios for a given circuit.

the total area for larger circuits.

layout the circuits. The two dimensional layout style becomes more effective in terms of packing the circuits. The two dimensional layout style becomes more effective in terms of

The results demonstrate that the layouts generated by our tool are competitive with the

achieved by the two tools.

The comparison factor from this point of view. Table 2 presents the comparison of the results

layout is used for the design, for the manual arrangement is kept general remain, as long as the winding density for the linear transistor arrangement is kept

whereas for smaller circuits (e.g., AND and OR gates) the IBM tool performed better. As a

For larger and more complex circuits (e.g., full adder, multiplier and decoder) better layouts (in terms of area) were generated by our tool.

The results obtained with layouts generated by our tool are competitive in terms of

improved obtained when 5 rows were used. By varying W' the layout area might be further

the available space on the chip. Note that for this example the minimal area layout was

which cells are used. In this case layouts must meet the imposed dimensions according to

why it is also important if area constraints are imposed by the surrounding environment in
REFERENCES

Acknowledgments

To thank Alltech Brouck, for helpful discussions and S. B. Shukla for proofreading.

The authors wish to express their gratitude to the following institutions for their financial support:

- Department of Electrical Engineering, Technion, Haifa, Israel.
- The Research Council, Technion, Haifa, Israel.
- The Office of the President, Technion, Haifa, Israel.

Their contributions were invaluable in making this work possible.

Exploring water, soil, and air quality monitoring, this paper is based on the work of the authors and their colleagues.

A preliminary version of this work was presented at the 1995 Conference on Electrical Engineering, Haifa, Israel.

The present results demonstrate that optimization in two dimensions can be efficiently explored for reducing the cost of layout.

The authors thank the reviewers for their valuable comments and suggestions.

One of the most important capabilities of the system is that it can deal with different aspect-oriented layouts.

Their useful comments are gratefully acknowledged.
### Table 2: Comparison Table

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>5216</th>
<th>7410</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3717</td>
<td>6554.7</td>
<td>4716</td>
</tr>
<tr>
<td>4</td>
<td>3726</td>
<td>3426.8</td>
<td>3792</td>
</tr>
<tr>
<td>4</td>
<td>3721</td>
<td>3792.7</td>
<td>3792</td>
</tr>
<tr>
<td>6</td>
<td>8904</td>
<td>9702.4</td>
<td>9073</td>
</tr>
</tbody>
</table>

### Table 1: Results for Full-Adder

<table>
<thead>
<tr>
<th></th>
<th>34</th>
<th>11</th>
<th>1468</th>
<th>238</th>
<th>61</th>
<th>1</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2</td>
<td>67</td>
<td>124.9</td>
<td>68.9</td>
<td>128.6</td>
<td>188.4</td>
<td>128.6</td>
</tr>
<tr>
<td>42</td>
<td>26</td>
<td>39.2</td>
<td>154.4</td>
<td>15.4</td>
<td>152.4</td>
<td>152.4</td>
<td>152.4</td>
</tr>
<tr>
<td>53</td>
<td>36</td>
<td>56.6</td>
<td>125.6</td>
<td>125.6</td>
<td>125.6</td>
<td>125.6</td>
<td>125.6</td>
</tr>
<tr>
<td>69</td>
<td>26</td>
<td>85.3</td>
<td>154.4</td>
<td>154.4</td>
<td>154.4</td>
<td>154.4</td>
<td>154.4</td>
</tr>
<tr>
<td>39</td>
<td>18</td>
<td>74.9</td>
<td>128.6</td>
<td>128.6</td>
<td>128.6</td>
<td>128.6</td>
<td>128.6</td>
</tr>
<tr>
<td>24</td>
<td>23</td>
<td>101.8</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
</tr>
<tr>
<td>28</td>
<td>16</td>
<td>101.8</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
<td>15.4</td>
</tr>
</tbody>
</table>

### References

Table 3: Computation time

<table>
<thead>
<tr>
<th></th>
<th>CPU U time (sec)</th>
<th>CPU U time (sec)</th>
<th>Assignment</th>
<th>Elements</th>
<th>No. of rows</th>
<th>No. of columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>13</td>
<td>27</td>
<td>32</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>0.6</td>
<td>21</td>
<td>29</td>
<td>30</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2.8</td>
<td>42.7</td>
<td>77.3</td>
<td>96</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4.3</td>
<td>96</td>
<td>445</td>
<td>62</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7.0</td>
<td>13</td>
<td>27</td>
<td>32</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Routing</td>
<td>CPU U time (sec)</td>
<td>Orientation</td>
<td>Assignment</td>
<td>Elements</td>
<td>No. of rows</td>
<td>No. of columns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Layout in the single row style.
Figure 6. Virtual search tree employed for the enumeration scheme.

Figure 5. Alligned transistor gates i share the same net versus alligned transistors gates that belong to different nets.

(a)  
(b)  
(c)