In the present the layout design for integrated circuits is in search of new ways of describing and constructing. Many systems of automatic design are in use or under development, since the manual design seems to be slow and ineffective for the requested large scale integration in the future.

The RELACS-System developed at the Humboldt University of Berlin has an other approach. It is not an automatic layout system at all, but it allows an adequate symbolic description of regular, hierarchical designs. Such a RELACS-description reflects the structural (Kolmogorov-) complexity of a layout design and is independent of the recursion depth and the description of the atomic elements. Nevertheless the description is short, our system can generate quickly the whole pattern or even arbitrary windows on it. On the other hand it is impossible to refine a placement by RELACS. This job remains to the designer.

While other automatic layout systems like CADIC [1] are based on logical-topological design level, RELACS is more geometrical oriented. The data base for RELACS are patterns of fixed size, i.e. rectangular blocks, length and widths of which are multiples of an elementary length \( \lambda \). It can be assumed that each pattern is filled with squares of size \( \lambda^2 \), which come from an external library and have some functional description or a description as a stick-diagram or something else. If the pattern is regular then the number of elementary squares of the pattern should be small.

Beyond this RELACS is dealing with configurations of patterns. Such a configuration represents the topological connections between the patterns composed together like in a picture of Piet Mondrian.

1 Akademie der Wissenschaften der DDR, Institut für Mathematik DDR - 1086 Berlin, Postfach 1304
2 Humboldt-Universität zu Berlin, Sektion Mathematik DDR - 1005 Berlin, Postfach 1297
Definition. A configuration is a spherical graph \((V,E)\), i.e. embedded in \(\mathbb{R}^2 \cup \{\infty\}\) without crossings, together with a compass \(c\). A compass \(c\) assigns to each edge an an incident vertex a direction from \(D=(e,s,w,n)\), such that for all edges \(k=(v_1,v_2) \in E\) the set \(\{c(k,v_1),c(k,v_2)\}\) is either \((e,w)\) or \((n,s)\).

In addition to this it is expected that the clockwise ordering of the edges around a vertex induced by the embedding is the usual one, i.e. if \(D\) is ordered by \(e>s>w>n\), and \(k_1,k_2,...,k_r\) is the clockwise ordering of all edges incident with \(v\) starting with "east" then \(c(k_1,v) \leq c(k_2,v) \leq ... \leq c(k_r,v)\).

Of course a single pattern is a 2-vertex configuration with 4 edges.

Configurations can be composed by cutting off the edges around \(\infty\) and gluing them with respect to their directions. Compositions of configurations can be described by sequences of some elementary operations. Operations like the dihedral ones or appending and breaking.

For example appending

\[
\begin{array}{c}
1 \\
\end{array}
\quad
\text{and}
\quad
\begin{array}{c}
2 \\
\end{array}
\quad
\text{gives}
\quad
\begin{array}{c}
1 \\
\end{array}
\quad
\begin{array}{c}
2 \\
\end{array}
\]

Note that there are many possibilities for gluing the east edges of 1 with the west edges of 2. They arise from the different realisations of the configuration as patterns. For example

\[
\begin{array}{c}
\begin{array}{c}
\text{corresponds to}
\end{array}
\end{array}
\quad
\begin{array}{c}
\begin{array}{c}
\text{corresponds to}
\end{array}
\end{array}
\]

while

\[
\begin{array}{c}
\begin{array}{c}
\text{corresponds to}
\end{array}
\end{array}
\quad
\begin{array}{c}
\begin{array}{c}
\text{corresponds to}
\end{array}
\end{array}
\]

The description of the append operation keeps the indefiniteness.

Another example for an operation is the breaking off. This operation you need for describing a configuration like

\[
\begin{array}{c}
1 \\
\end{array}
\quad
\begin{array}{c}
2 \\
\end{array}
\quad
\begin{array}{c}
3 \\
\end{array}
\]