Floorplanning: taking layout information into account at early stages of the design process.

Illustration of the floorplan-based design strategy on Gajski’s Y-chart
THE FLOORPLAN-BASED DESIGN METHODOLOGY

Structure is immediately translated into layout, by making estimates of the area, I/O, etc.
THE FLOORPLAN-BASED DESIGN METHODOLOGY (Continued)

* A chip is essentially a two-dimensional medium; taking this aspect into account in early stages of the design helps in creating designs of good quality.

* Floorplanning gives early feedback: thinking of layout at early stages may suggest valuable architectural modifications.

* Floorplanning fits very well in a top-down design strategy, the stepwise refinement strategy also propagated in software design.

* Floorplanning assumes however flexibility in layout design, the existence of cells that can adapt their shapes and terminal locations to the environment.
FLOORPLANNING CONCEPTS

* **Abutment**: establishing connections between cells by putting them directly next to each other, without the necessity of routing.

* **Leaf cell**: a cell at the lowest level of the hierarchy; it does not contain any other cell.

* **Composite cell**: a cell that is composed of either leaf cells or composite cells. The whole chip is the highest-level composite cell.

* **Restriction**: all leaf cells and composite cells are supposed to be rectangular.
Slicing floorplans: a floorplan with the property that a composite cell’s subcells are obtained by a horizontal or vertical bissection of the composite cell. Slicing floorplans can be represented by a slicing tree.
FLOORPLANNING CONCEPTS (Continued)

* In a slicing tree, all cells (except for the top-level cell) have a *parent*, and all composite cells have *children*.

* Not all floorplans are slicing.

Composite cell without the slicing property

* Limiting floorplans to those that have the slicing property is reasonable: it certainly facilitates floorplanning algorithms.
SLICING TREE GENERATION

The automatic generation of a slicing tree is similar to the automatic placement of cells. One could e.g. use a *min-cut partitioning* algorithm.
SHAPE FUNCTIONS

Flexible cells imply that cells can assume different aspect ratios.

* One can e.g. assume that all implementations of the cell have the same area $A$. The relation between the width $w$ and the height $h$ is: $hw = A$, or $h = \frac{A}{w}$. The shape function is a hyperbola.
SHAPE FUNCTIONS  
(Continued)

* Very thin cells are not interesting and often not feasible to design. The shape function is a combination of a hyperbola and two straight lines.
SHAPE FUNCTIONS (Continued)

* Leaf cells are built from discrete transistors: it is not realistic to assume that the shape function follows the hyperbola continuously.

* In an extreme case, a cell is rigid: it can only be rotated and mirrored during floorplanning or placement. This type of cell is called an *inset cell*.

The shape function of a $2 \times 4$ inset cell.
SHAPE FUNCTIONS
(Continued)

* In general a piecewise linear function can be used to approximate any shape function.

* The points where the function changes its coefficient are called the break points of the piecewise linear function.
THE SHAPE FUNCTION OF COMPOSITE CELLS

The shape function of a composite cell obtained by vertical abutment is the \textit{sum} of the shape functions of its children cells.

\[
f_3 = f_1 + f_2
\]
THE SHAPE FUNCTION OF COMPOSITE CELLS (Continued)

* In case of horizontal abutment, the computation of the new shape function is somewhat more complicated: $f_3^{-1} = f_1^{-1} + f_2^{-1}$.

* A choice for the minimal shape of composite cell fixes the shapes of the shapes of its children cells.
SIZING ALGORITHM FOR SLICING FLOORPLANS

* The shape functions of all leaf cells are given as piecewise linear functions.

* Traverse the slicing tree in order to compute the shape functions of all composite cells (bottom-up composition).

* Choose the desired shape of the top-level cell; as the shape function is piecewise linear only the break points of the function need to be evaluated, when looking e.g. for the minimal area.

* Propagate the consequences of the choice down to the leaf cells (top-down propagation).

Note: the algorithm operates in polynomial time!