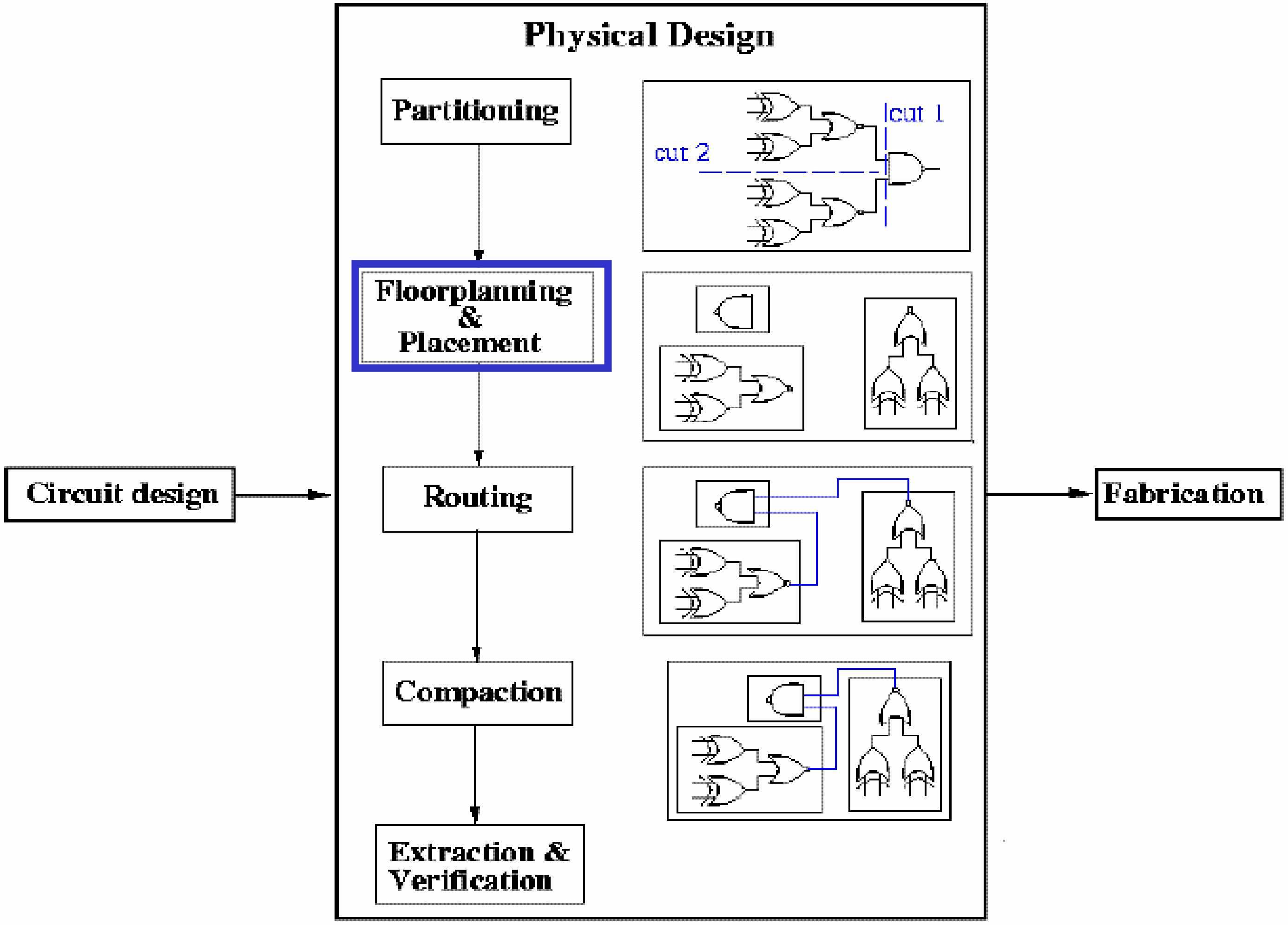
Floorplanning

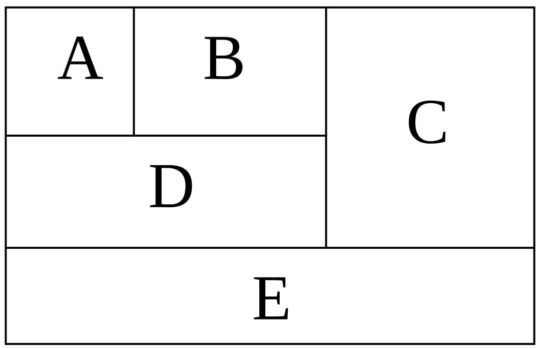
Susmita Sur-Kolay
I. S. I. Kolkata

Physical Design Flow



Floorplans

- A *floorplan* is the placement of (flexible) blocks with fixed area but unknown dimensions
- Blocks are usually assumed to be rectangular in shape
- Some work done with L-shaped blocks
- Usually a lower and an upper bound on the Aspect ratio are given



A Floorplan with 5 rectangular blocks

Floorplanning Problem

Objective :

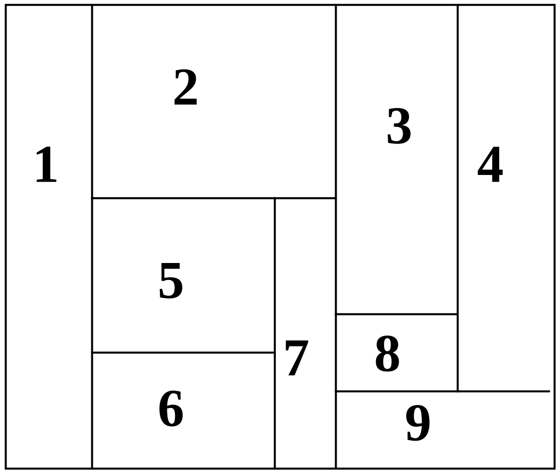
Finding a suitable topology of the blocks and their dimensions such that

– Area minimized

– Critical net delay minimized

Slicible Floorplans or Rectangular Duals

A floorplan is *Slicible* if it is obtained by recursively bipartitioning a rectangle into two slicible floorplans either by a horizontal or a vertical line

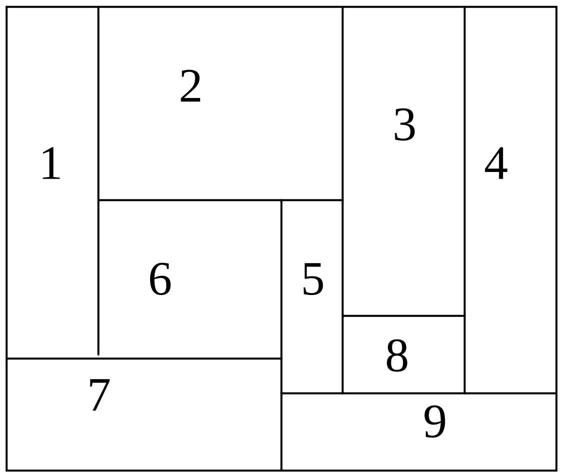


Slicible Floorplan

- easy to represent with binary tree
- Sizing and routing algorithms are polynomial time

Non-Slicible Floorplans

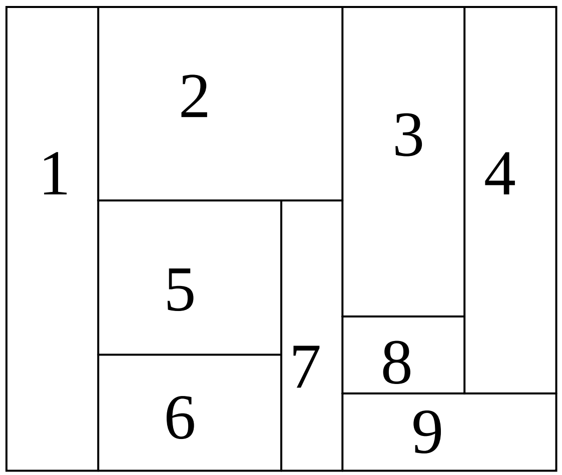
A floorplan that is not *Slicible*

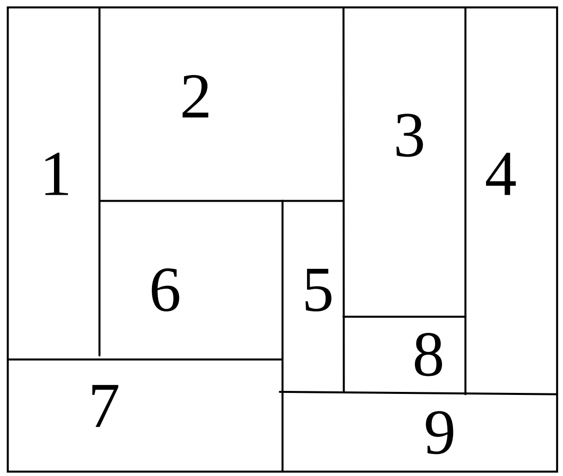


NonSlicible Floorplan

Floorplans

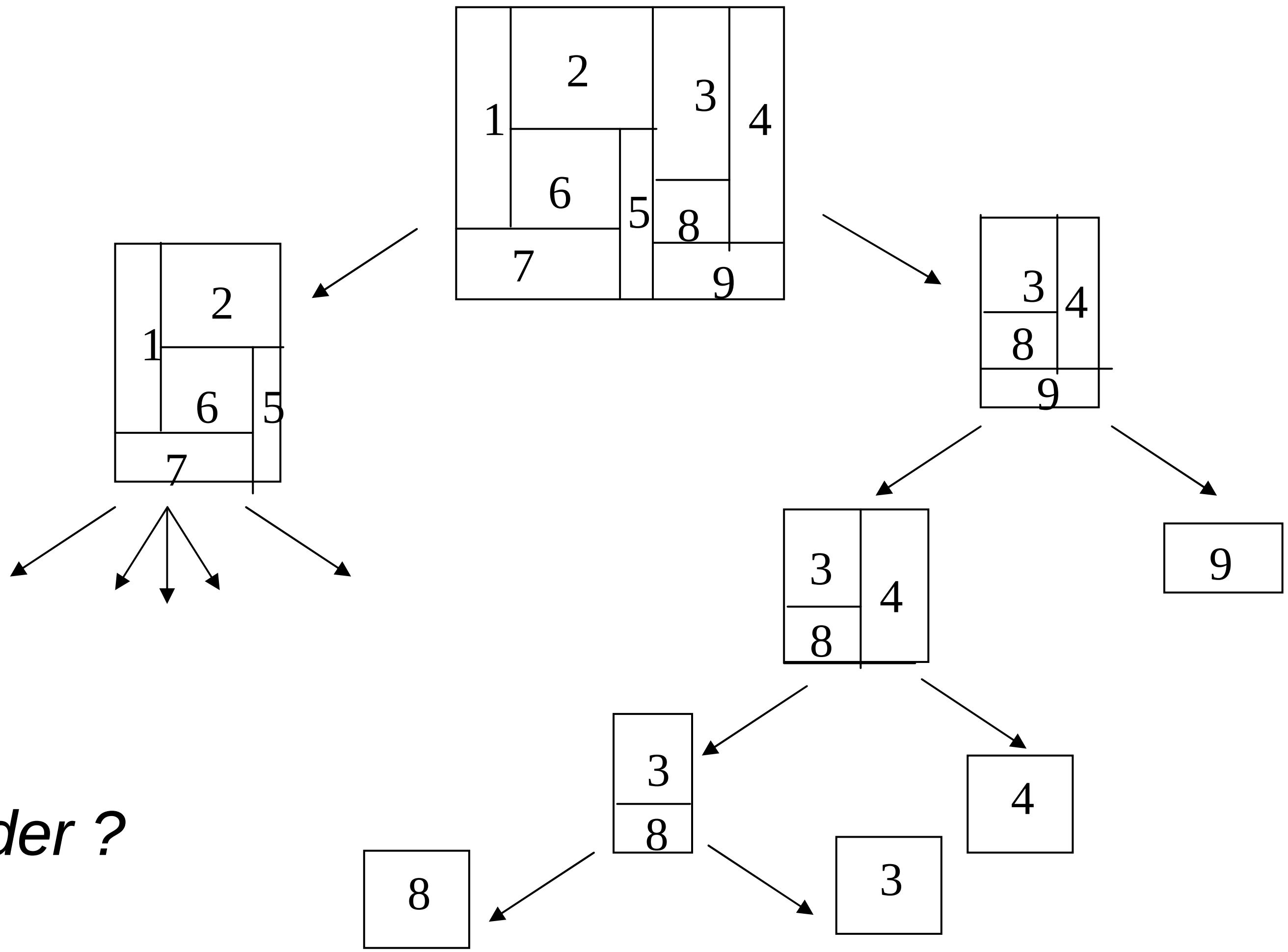
A *floorplan* is said to be hierarchical order of *k*, if it can be obtained by recursively partitioning a rectangle into at most *k* parts





What are the hierarchical orders of the above floorplans ?

Hierarchical Floorplan Trees



Order ?

Floorplanning by Rectangular Dualization

Input :

A set of rectangular blocks
A set of realizations, i.e., (width, height) pairs, for each block
Adjacency graph for the blocks

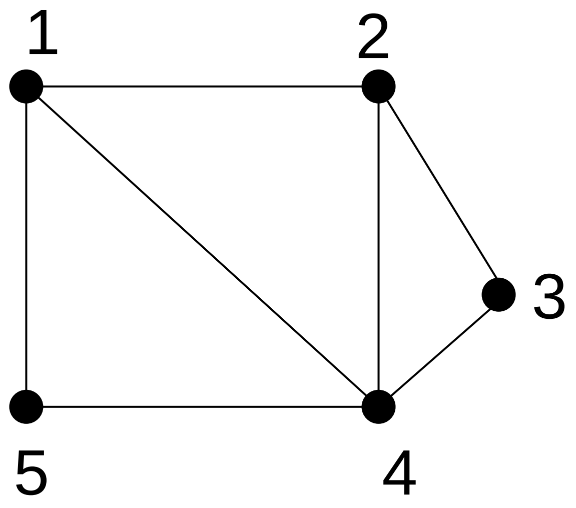
Requirements :

Topology generation --- Location of each block within a rectangular envelope such that no two blocks overlap
Sizing --- An appropriate size, i.e., width and height, of each block

Objectives :

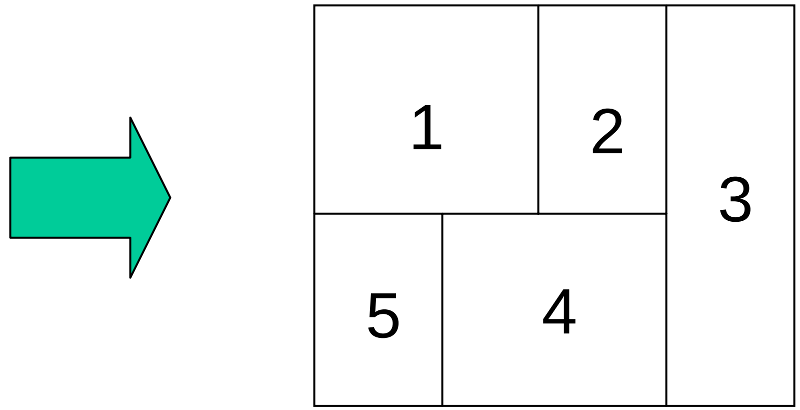
Minimize area of the rectangular envelope
Reduce net-length for critical nets

Rectangular Dualization - An Example



Adjacency Graph

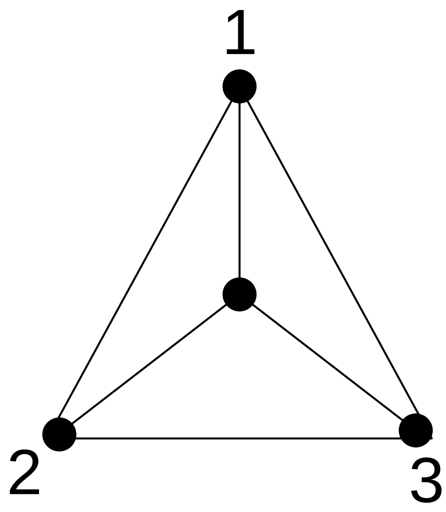
Geometric Dual of Adjacency Graph



Rectangular Graph

A rectangular graph is an adjacency graph

that yields a rectangular dual



A\` Forbidden Rectangular Graph

Characteristics of Rectangular Dualization

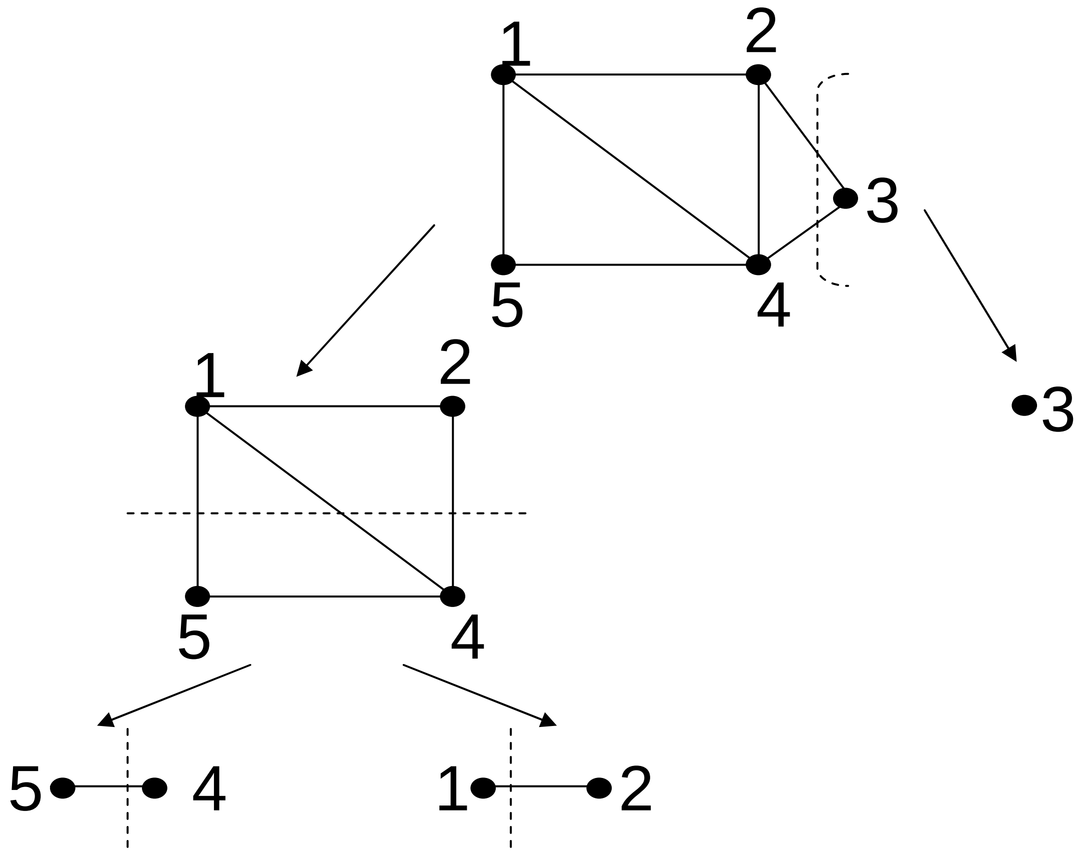
- Every face, except the exterior, is a triangle
- All internal vertices have degree $\geq$ 4
- all cycles that are not faces and the exterior face have length $\geq$ 4

PROPERLY TRIANGULATED PLANAR GRAPH (PTP)

Given an input PTP, any rectangular floorplan
can be generated from it in linear time

- Four corners - NW, NE, SE, SW
- bottom up method
- does not guarantee slicibility even if it exists

Topology Generation : A simple example



Sizing

- selecting a particular (length, width) pair for each block
- objective is to minimize space wastage
- complex for non-slicible floorplans

Sizing Methods - Linear Programming

Based on Mixed Integer Linear Programming

(wi,hi) : width and height of module Mi
(xi,yi) : left and right corners of module Mi
(x,y) : width and height of final floorplan
(ai,bi) : min / max values of aspect ratios of Mi

Non-overlap constraint: xi + w_i *<= xj*
Module size constraint: wihi >= Ai (if module size unknown, assume Ai a lower bound on the area of each module)

Cost function: To minimize total area A= xy

Sizing Methods - Linear Programming

The non-linear objective function can be replaced by fixing the width of the floorplan to W, and attempting to minimize y *only*

The inequalities can be fed into an LP solver LINDO, and a solution obtained.

The procedure may be repeated for several values of *W*, and the desirable one picked up

Wire length and routing area can also be estimated and incorporated to form a set of complex *LP* equations

Sizing Methods - Simulated Annealing

Guiding Factors -
Solution space
movement from one solution to another
cost evaluation function

A binary tree representation of a floorplan is converted into a normalized Polish expression

Movements
Exchange two operands when there are no other operands in between
Complement a series of operators between two operands
Exchange adj operand and operator if the resulting expression is a normalized Polish expression

Sizing Methods - Simulated Annealing

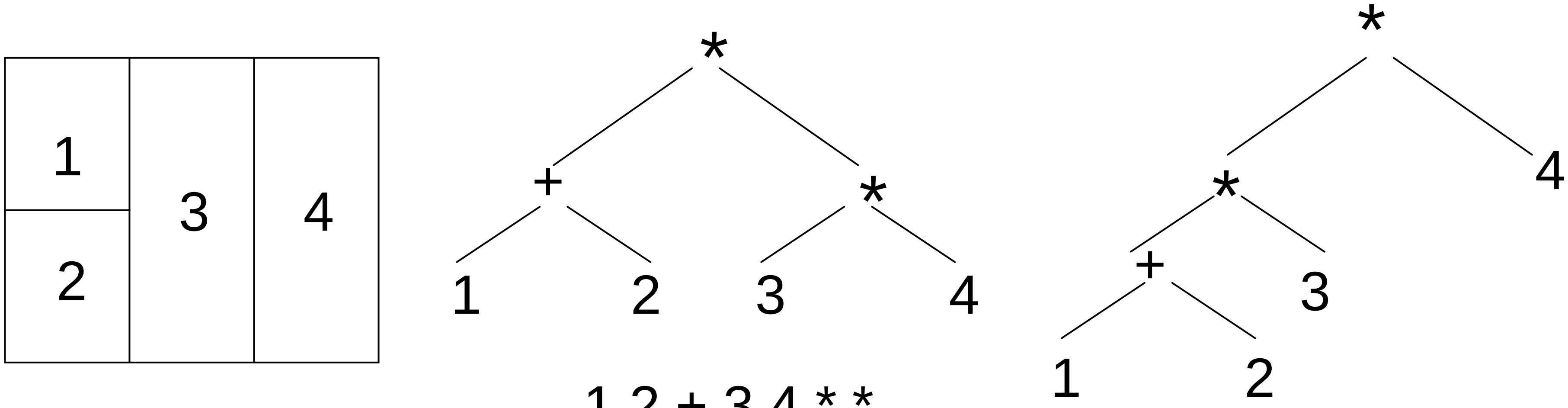
- solution is restricted to slicing floorplans only

- for easy representation of slicing floorplans, *Polish* notation is used

- Polish expression is a string of symbols obtained by traversing the binary floorplan tree in post-order

Sizing Methods - Simulated Annealing

Generating polish expressions from floorplan trees -



1 2 + 3 4 * *

Normalized Polish expression -

*if there is no consecutive *'s and +'s in the expression*

Sizing Methods - Simulated Annealing

Theorem. There is a 1-1 correspondence between the set of normalized Polish expressions of length 2n - 1, and the set of slicing structures with n leaves

Three types of movements used to move from one floorplan to another -

1. Exchange two operands when there are no other operands in between
2. Complement a series of operators between two operands
3. Exchange adjacent operand and operator if the resulting expression is in normalized Polish form

Sizing Methods - Simulated Annealing

Annealing Schedule

Temperature changes in the manner $T_i = r.\ T_{i-1}$
Typical value of r is 0.85
Algorithm starts with an initial normalized Polish expression

Default initial Polish expression is
12 * 3 * 4 * …. * n *, which corresponds to placing n modules horizontally next to each other

At each temperature, enough moves are attempted until there are N downhill moves or the total number of moves exceeds 2N, where $N = \gamma.n$, where γ is user specified symbol

Sizing Methods - Genetic Algorithm

GALLO Proposed by M Rebedaudengo and M S Reorda in 1996

uses the usual GA operators

each individual in the population is represented by a string of n genes

the ith gene is the index of the implementation chosen for the ith basic rectangle

Sizing Methods - Procedure GALLO

```
P0 = create_initial_population();
compute_initial_fitness();
i=0;
while (stopping_condition <> TRUE)
{
   A = P0 ;
   for j = 0 to i
   {
      /* generation of new elements */
      op = select_an_operator();
      Sj^i  = apply_operator(op, Pi);
      A = A U Sj^i;  j = j + 1;
    }
    compute_fitness(A);
    Pi+1 = { the P best individuals of A};
    i = i + 1;
}
```

Sizing Methods - GALLO

An array representation of an implementation of a Rectangular Floorplan

1	2	3	4	5	6	7	8	9	10	11
4	1	2	3	5	6	1	8	2	3	5

Chromosome / Individual

Gene

Sizing Methods - GALLO

Applying Mutation operator

A gene is randomly selected, and its value changed

1	2	3	4	5	6	7	8	9	10	11
4	1	2	3	5	6	1	8	2	3	5

4	1	2	3	**3**	6	1	8	2	3	5

Sizing Methods - GALLO

Applying 2-cut crossover

Two parents of different genetic patterns

4	1	2	3	5	6	1	8	2	3	5

$\otimes$

5	4	2	1	2	4	1	7	6	3	8

$\Downarrow$

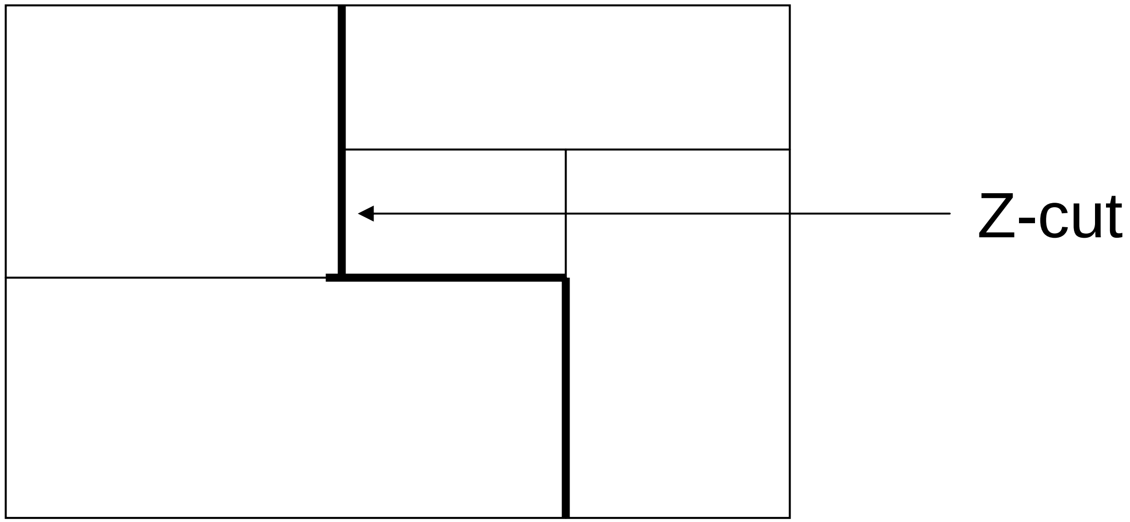
4	1	2	1	2	4	1	7	2	3	5

5	4	2	3	5	6	1	8	6	3	8

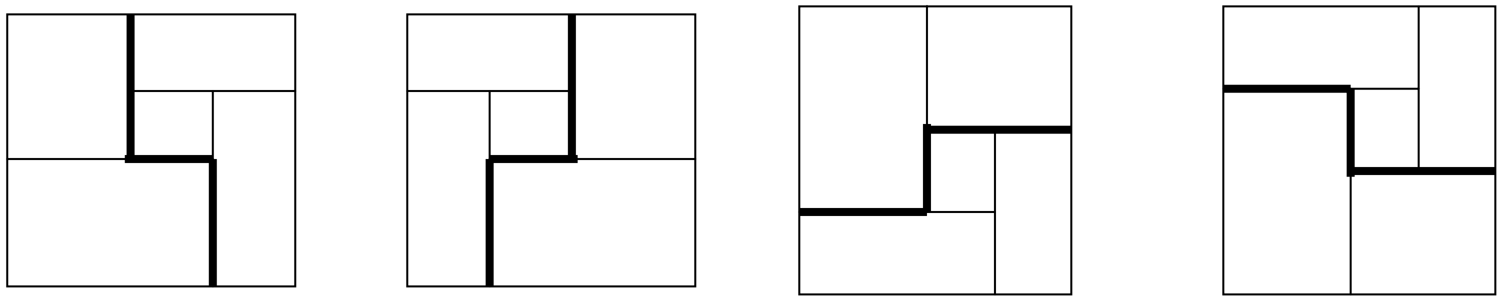
Unified Floorplan Generation based on Rectangular Dualization

- Integration of two phases
- Phase I - Topology generation by using sequence of slices in a planar triangulated graph
  - consider a sequence of cuts (slices or Z-cuts) as an AND-OR graph
  - use algorithm AI_FP*. Which is a variant of AO*
- Phase II - Sizing

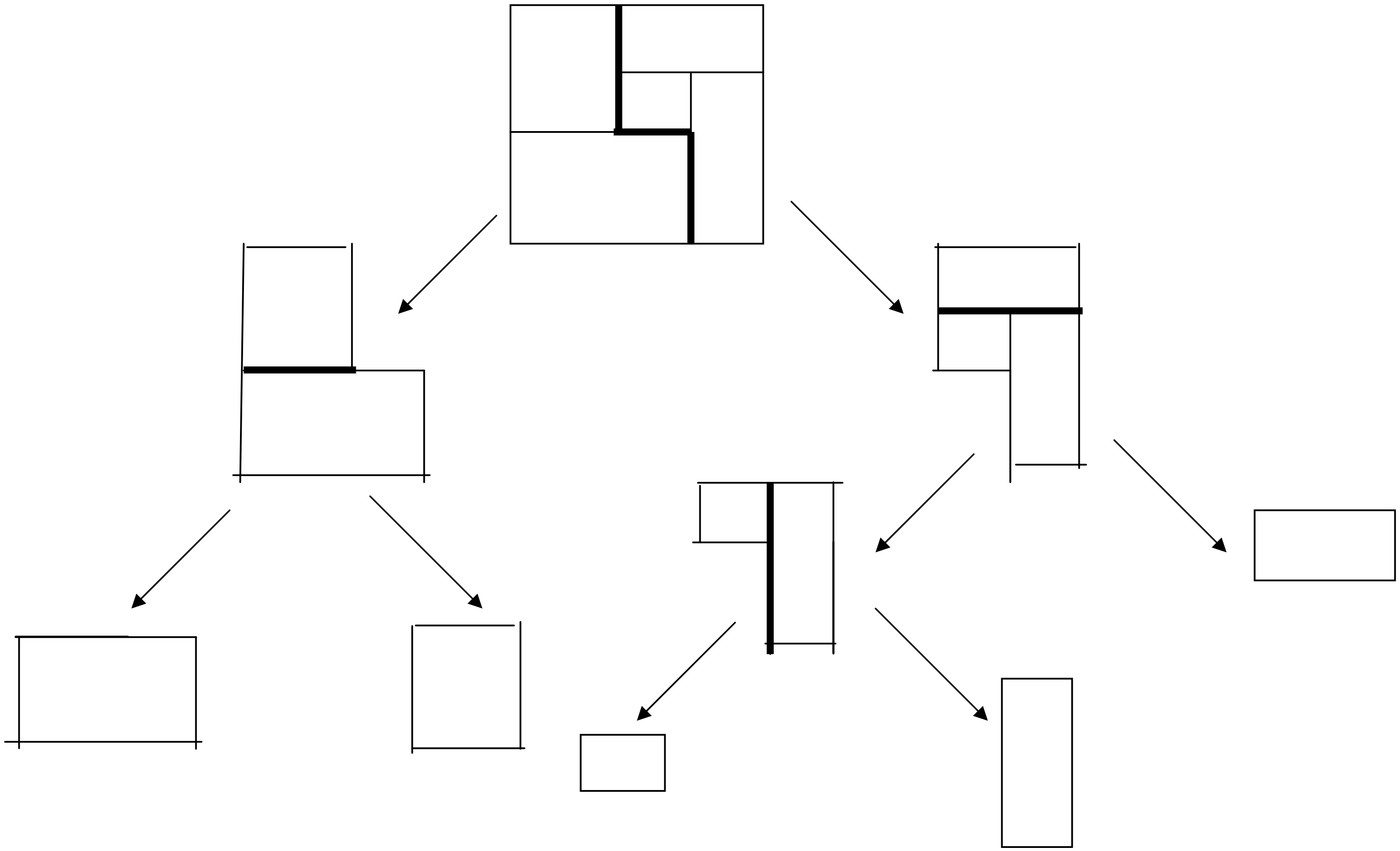
Unified Floorplan Generation based on Rectangular Dualization



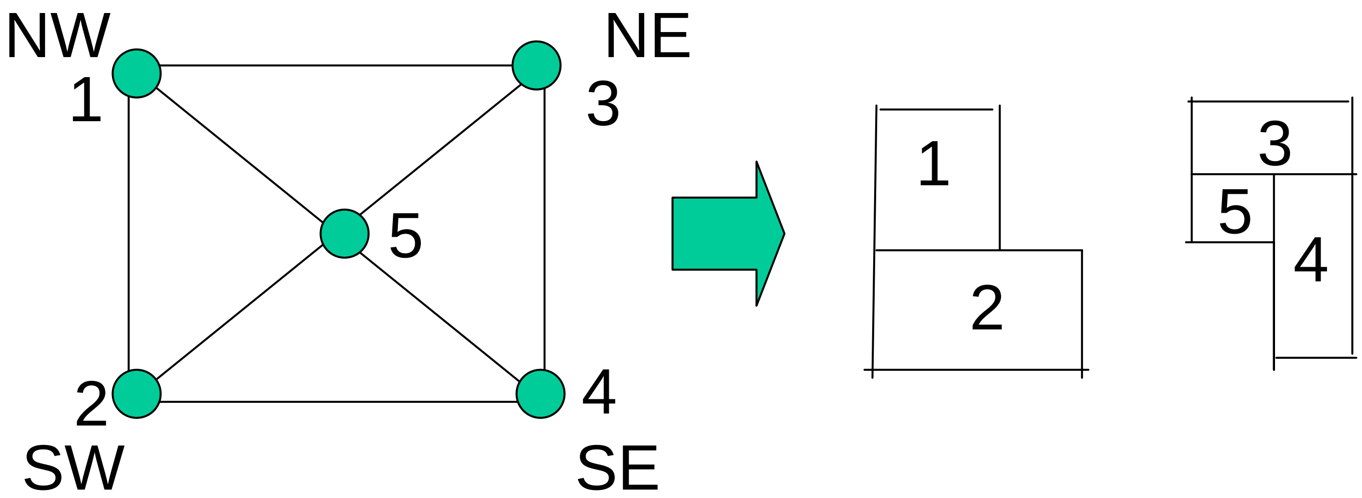
Four Types of Z-cuts



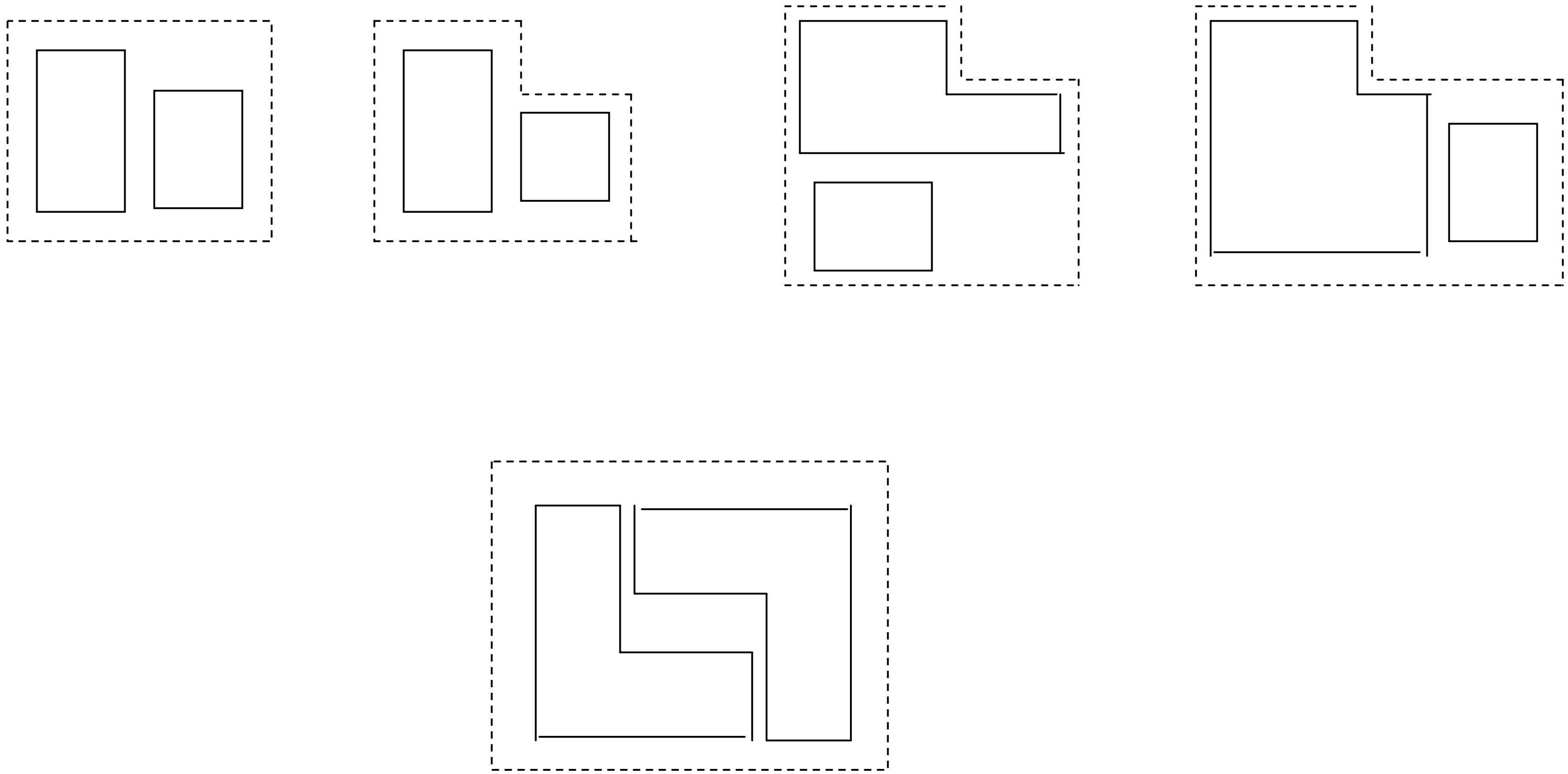
Floorplan Tree with Z-cuts



Application of Z-cuts



Sizing in AO_FP*



References

- Algorithms for VLSI Physical Design Automation
  Naveed Sherwani
  Kluwer Academic Publishers

- An Introduction to VLSI Physical Design
M. Sarrafzadeh and C. K. Wong
McGrawHill International Edition