Representing Topological Structures for 3-D Floorplanning

Renshen Wang[1]
Evangeline Young[2]
Chung-Kuan Cheng[1]

[1] University of California, San Diego

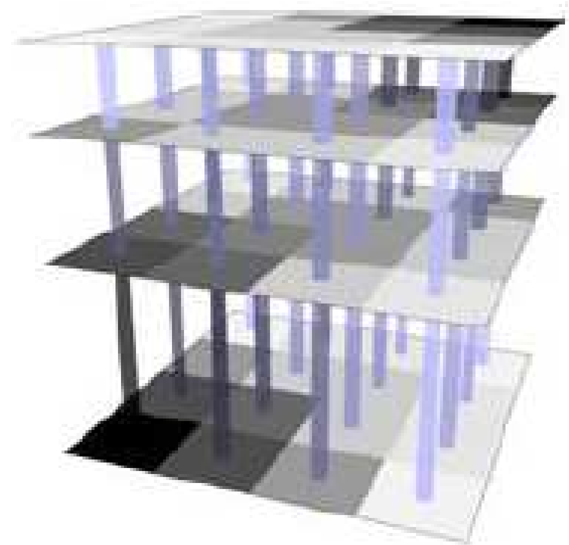
[2] The Chinese University of Hong Kong

Goal of Today's Talk

- Overview on 2-D and 3-D Floorplanning Representations
  - Encoding schemes, floorplan classes
  - Difficult extending of 2-D → 3-D
- 3-D Mosaic Floorplans
  - Mosaic: Definition, Pros and Cons
- Introducing Twin Tree Representations
  - TBT [Yao 2001] → TQT
- Future Directions

Introductions

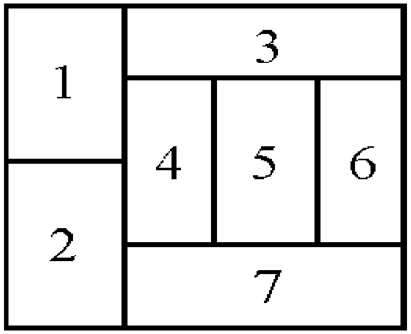
- 3-D circuits
  - Reducing interconnect length
  - Ideally $\sqrt{n} \rightarrow \sqrt[3]{n}$
  - Interconnect bottleneck on performance
  - Moore's law…



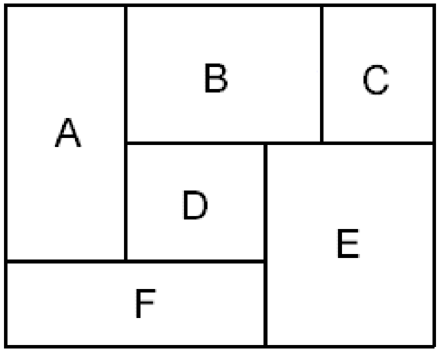
- Challenges on 3-D design
  - Thermal behavior, cooling method
  - Physical design complexity: 1 more dimension

Floorplan Representations

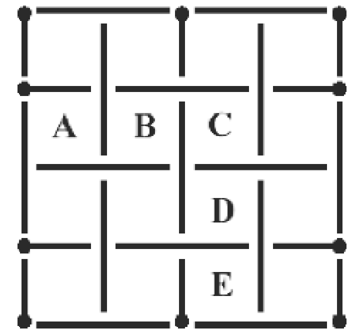
- On slicing floorplans
  - [Otten 1982]
  - Simple but incomplete



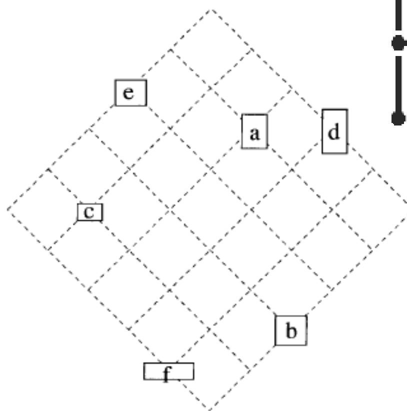
- On mosaic floorplans
  - CBL [Hong 2000]
  - TBT [Yao 2001]
  - Q-seq [Zhuang 2002]
  - …
  - Simple, and complete (with dummy modules)



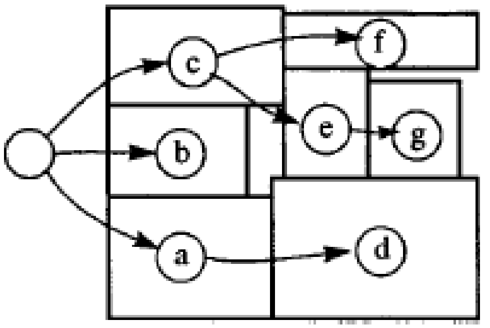
Floorplan Representations



- Representing topological orders
  - BSG [Nakatake 1994]
  - SP [Murata 1995]
  - TCG [Lin 2001]
  - Usually complete

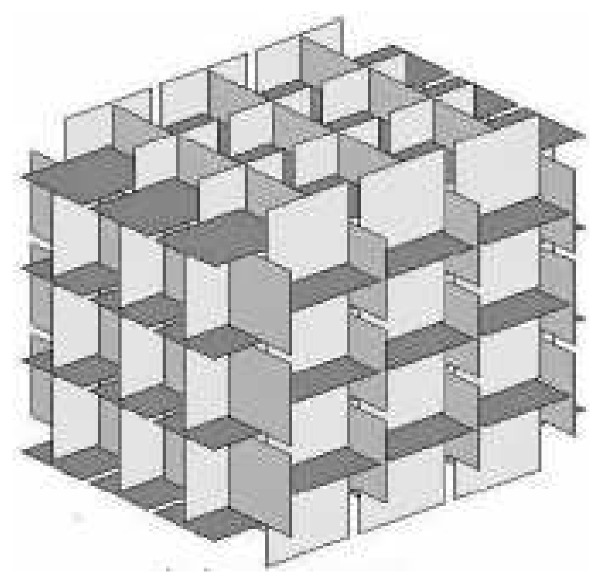


- Encoding packing processes
  - O-tree [Guo 1999]
  - B*-tree [Wu 2000]
  - Complete, topology depends on modules



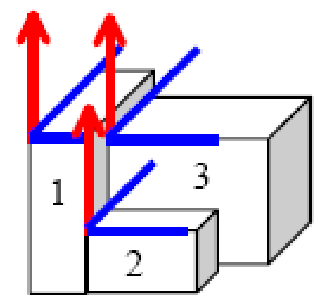
Extending to 3-D

- 2-D slicing → 3-D slicing [Cheng 2005]
  - Straight forward, incomplete
- Sequence pair in 2-D → [Yamazaki 2000]
  - Sequence triple (3), incomplete
  - Sequence quintuple (5) complete, highly redundant
- 2-D BSG → 3-D BSG [Zhang 2007]
  - Incomplete

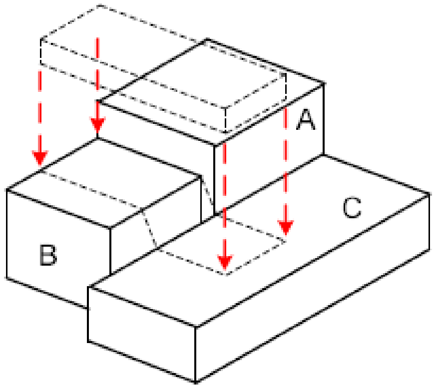


Extending to 3-D (cont.)

- 2-D TCG → 3-D subTCG [Yu 2004]
  - Straight forward and complete, highly redundant
- 2-D CBL → 3-D CBL [Ma 2005]
  - Complex, incomplete

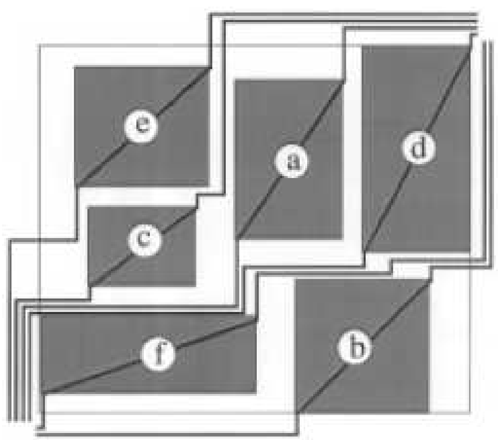


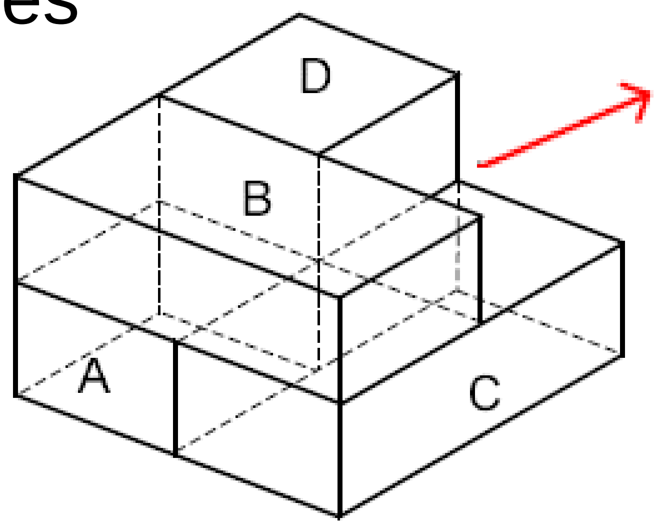
- 2-D O-tree → 3-D 1 Tree + 2 sequences [Wang 2008]
  - Complex, incomplete



Why difficult to extend?

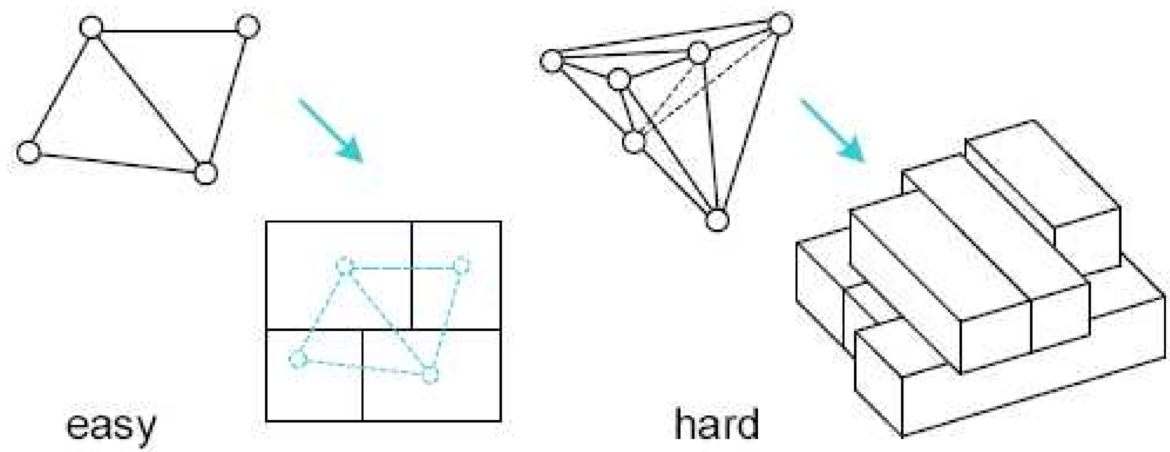
- A structure topologically "unsortable"
  - Unlike 2-D cases…
  - B, C, D cannot be sorted in the arrow's direction
  - More complex structures





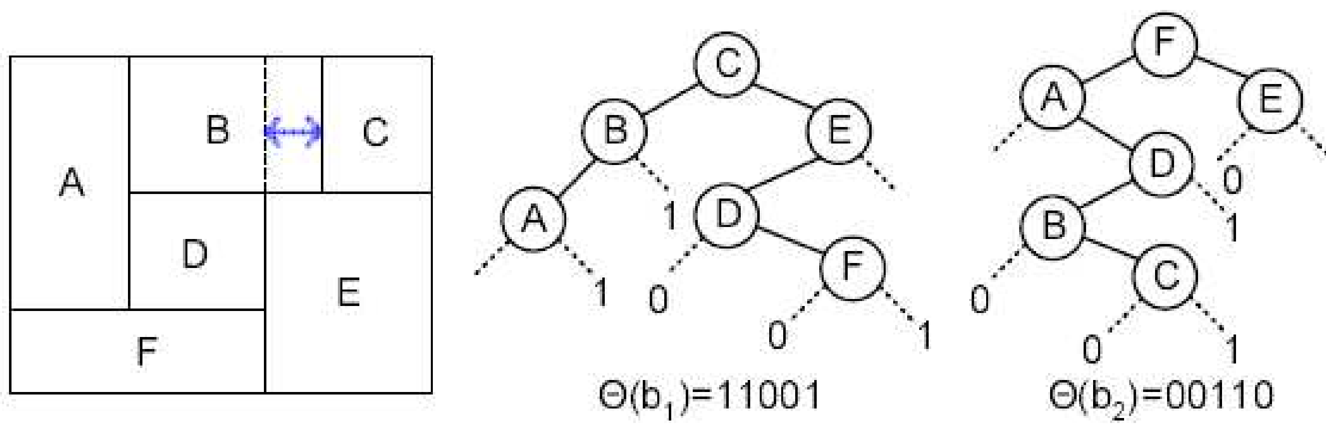
Why difficult to extend? (cont.)

- A “rectangular dual” formulation for 2-D [Kozminski & Kinnen 1984] revised for 3-D
  - Given a graph G=(V,E), find a set of cuboids as V, with contact relations as E
  - Tractable in 2-D, intractable in 3-D [Wang 2009]



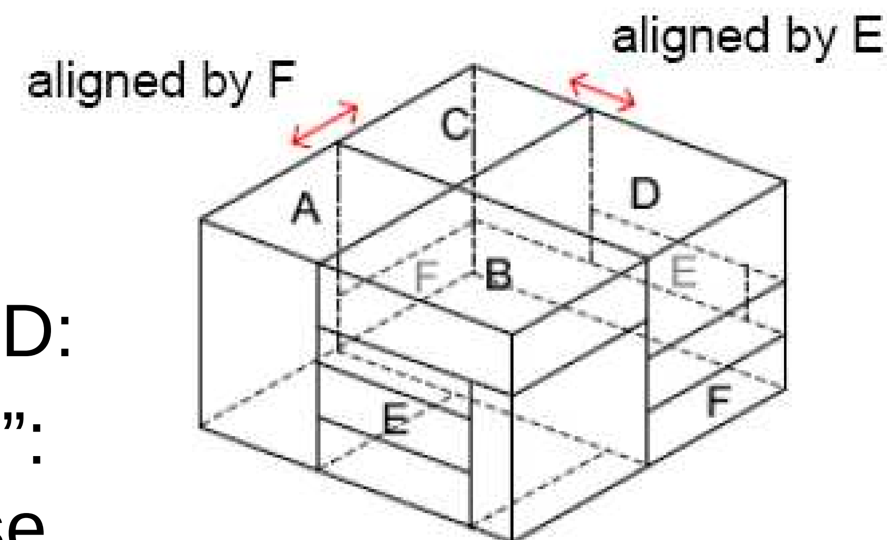
Our Approach

- Try to resolve the "unsortability" problem by tree based representations
  - Sequences are always "sorted" in some way
- 2-D mosaic floorplan and twin binary tree
  - One-to-one matching [Yao 2001]



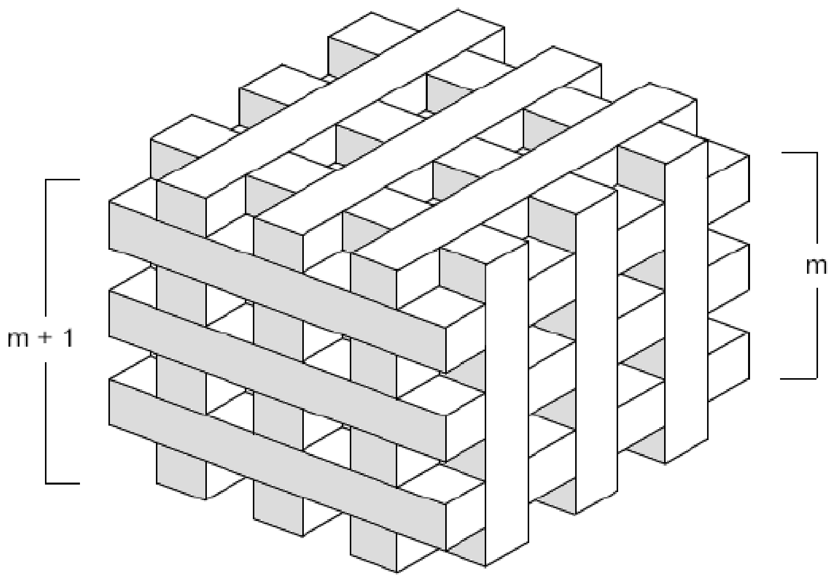
3-D Mosaic Floorplans

- Extending the concept to 3-D
  - Each module is a cuboid
  - Cuboids separated by internal rectilinear polygons
  - No empty space
  - Can be complete
  - Non-degenerate property is lost in 3-D:
  - A, B, C, D form a “+”: 2-D degenerate case



3-D Mosaic Floorplans (cont.)

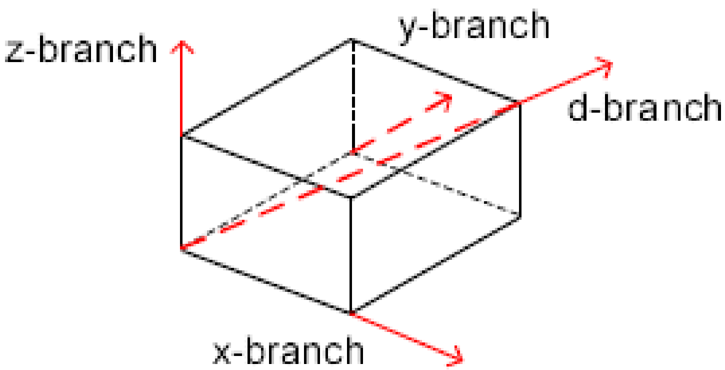
- Completeness
  - 2-D mosaic floorplans need $n - \lfloor \sqrt{4n-1} \rfloor$ dummy modules, O(n)
  - 3-D needs $O(n^{1.5})$ dummy modules

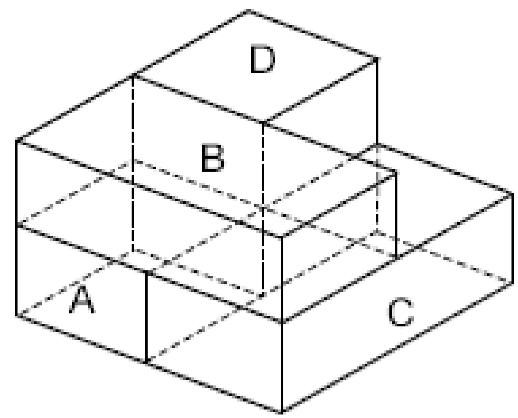


(Also shows why 3D-BSG is incomplete)

Quaternary Trees

- Binary tree for 2-D, quaternary for 3-D
  - x, y, z for branches along the 3 axex
  - d for diagonal branch



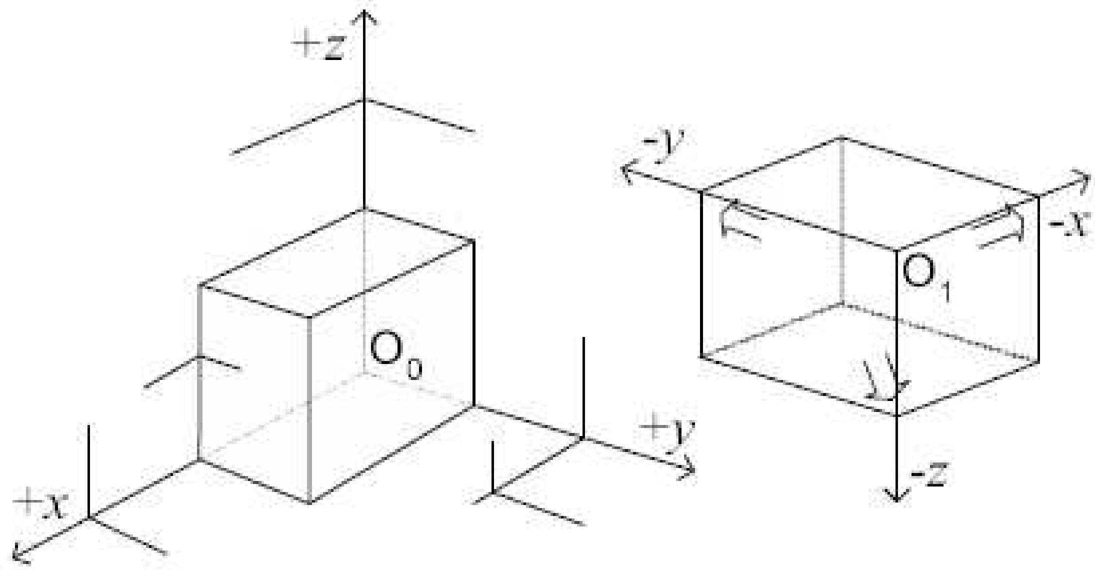


  - x: A → C, y: A → D, z: A → B
  - Another d-branch for the module placed on C

Quaternary Trees (cont.)

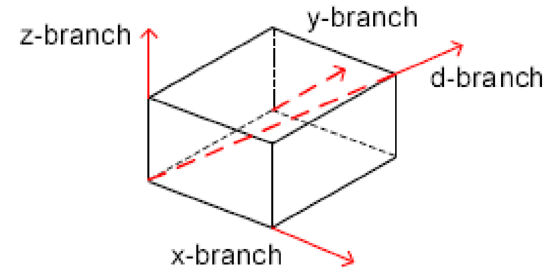
- Definition of quaternary tree T_0 rooted at the (-x, -y, -z) corner of the floorplan
- Node A has a child node B on branch:
  - x: when $A_{+x} = B_{-x}$, $A_{-y} = B_{-y}$, and $A_{-z} = B_{-z}$
  - y: when $A_{+y} = B_{-z}$, $A_{-x} = B_{-x}$, and $A_{-z} = B_{-z}$
  - z: when $A_{+z} = B_{-z}$, $A_{-x} = B_{-x}$, and $A_{-z} = B_{-z}$
  - d: when $A_{+x} = B_{-x}$, $A_{+y} = B_{-y}$, and $A_{+z} = B_{-z}$
- Tree T_1 rooted at the (+x, +y, +z) corner
  - Flipping all the “+/-”s above

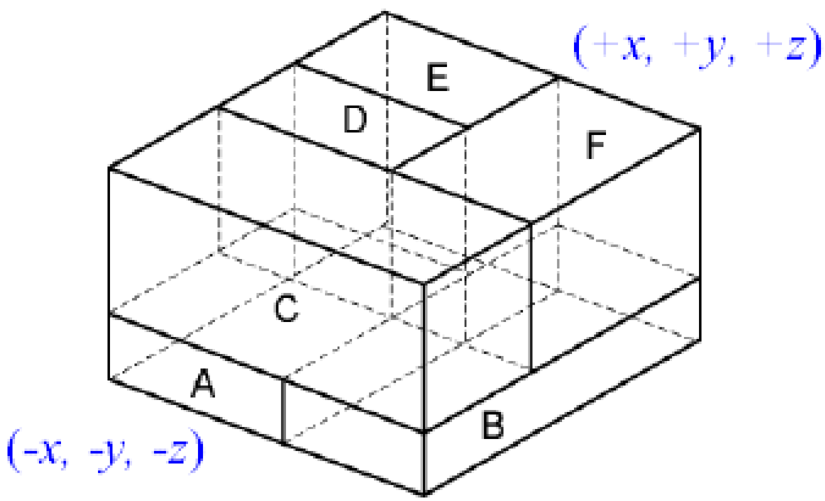
Twin Quaternary Tree

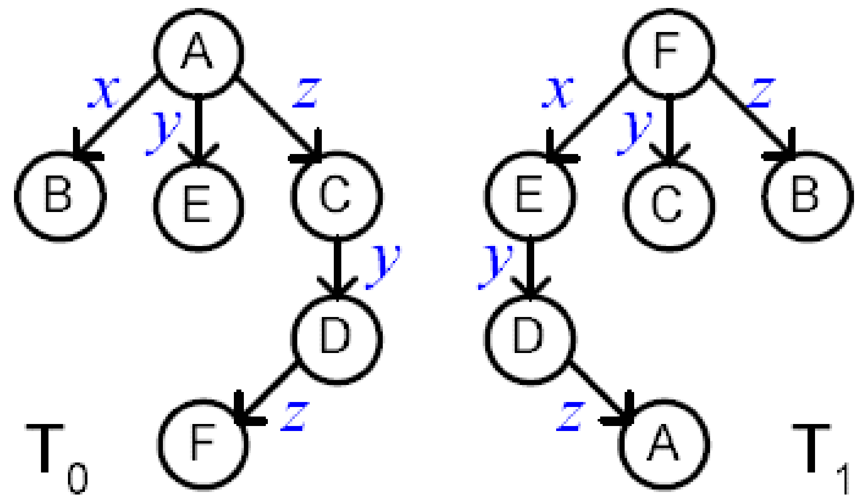


- Given a 3-D mosaic floorplan of n cuboids, T_0 and T_1 can be constructed as quaternary trees of n nodes.
- (T_0, T_1) is called a twin quaternary tree (TQT)

Example



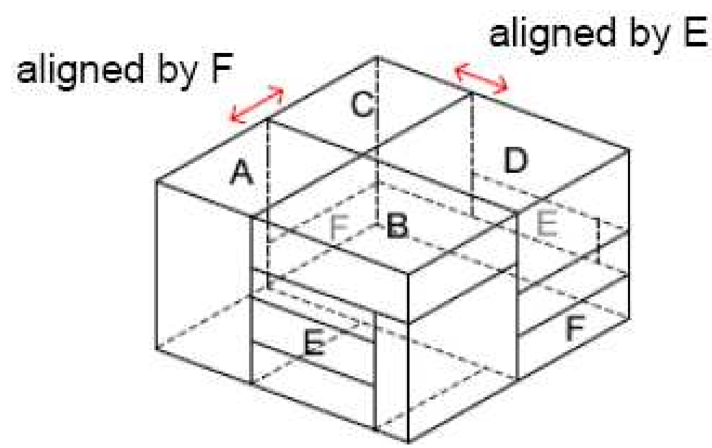




- When the 3-D floorplan has only one layer, the TQT is also TBT (twin binary tree)

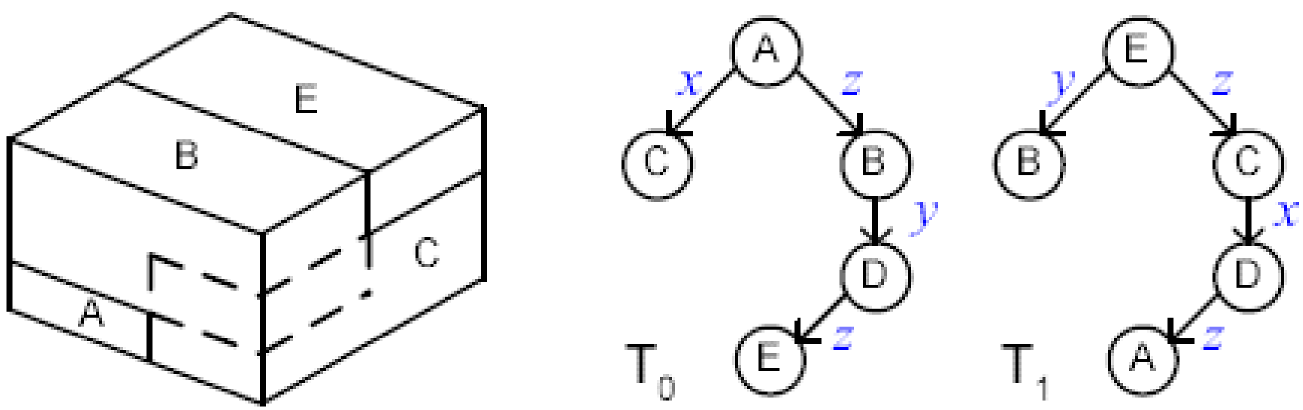
Effective encoding

- A twin quaternary tree pair (T_0,T_1) derived from a 3-D mosaic floorplan uniquely decides the floorplan
- The reverse is not true
  - A mosaic floorplan may have multiple TQT representations
  - As in the case with a 2-D degenerate surface



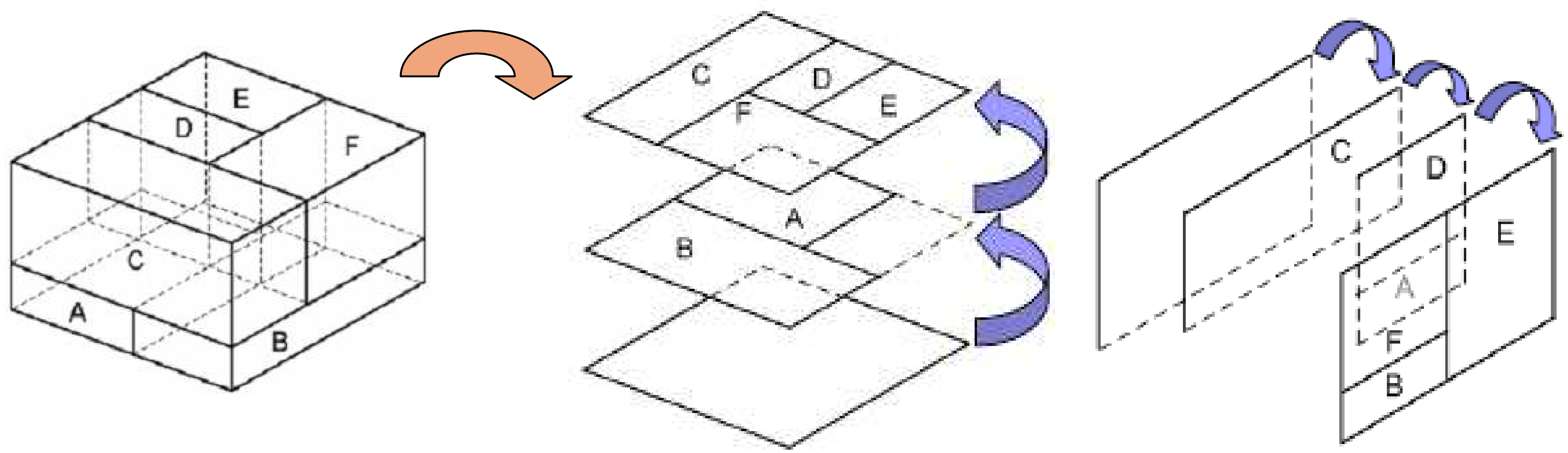
False Mosaic Floorplans

- (T_0,T_1) has the "twin" property when
  - 1) All module surfaces are consistent
  - 2) No two modules overlap on space
  - 3) No empty space in the floorplan
- For TBT 1) → 2),3), but not for TQT



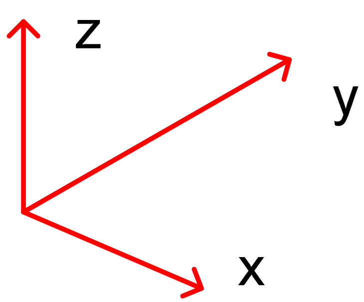
Recovering Floorplan from TQT

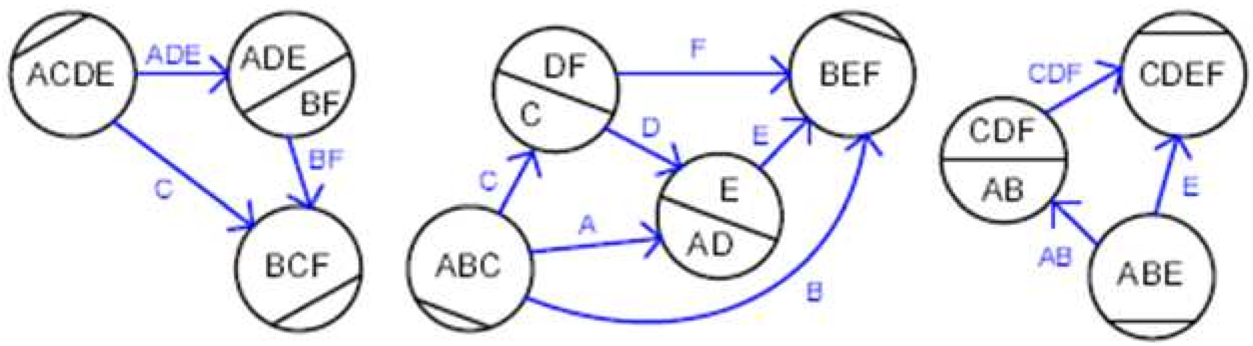
- Given a TQT=(T_0,T_1) of n nodes, we can recover the 3-D mosaic floorplan and check its correctness in O(n) time
- First, extract all the internal rectilinear polygons



Recovering Floorplan from TQT

- Module surfaces grouped into polygons by branch equations
- Each module has 3 inequalities
  - $A_{-x} < A_{+x}$, $A_{-y} < A_{+y}$ and $A_{-z} < A_{+z}$
- Topological order of polygons

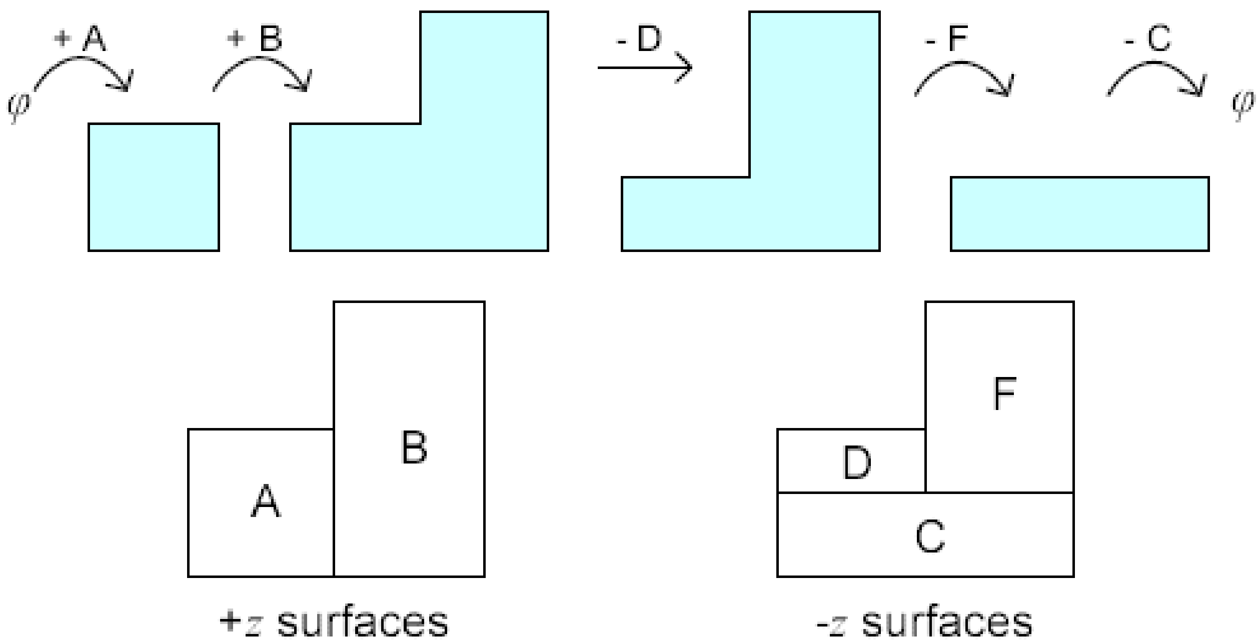




$\pm x$ polygons $\pm y$ polygons $\pm z$ polygons

Recovering Floorplan from TQT

- Two sides of each internal polygon should be identical
  - Checked using a map based O(n) algorithm



Solution Space

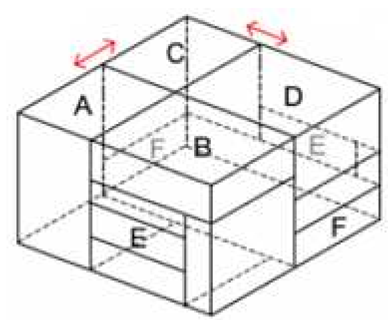
- Count of n-node p-ary tree $\frac{1}{(p-1)n+1}\binom{pn}{n}$
- Two n-node quaternary tree $\frac{1}{(3n+1)^2}\binom{4n}{n}^2$
- Total number of TQT has an upper bound

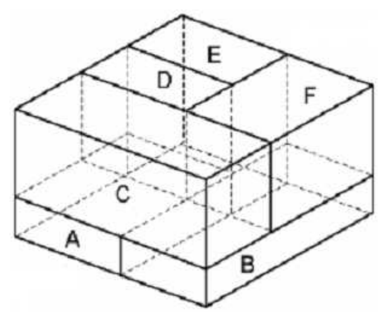
$$\frac{(n!)^2}{(3n+1)^2}\binom{4n}{n}^2$$

- Actual number much smaller, but still prohibitingly large
- Efficient operations on solutions needed for metaheuristic optimizations

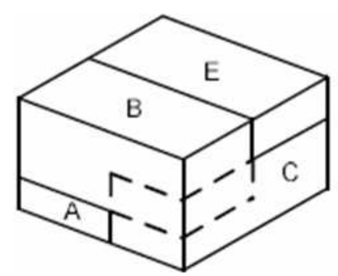
Property like Twin Binary Tree?

- TBT property allows
  - Direct manipulation on trees → a new solution
  - Tree rotations [Young 2003]
- TQT property: open question





?



Conclusions and Future Works

- Floorplan representations, 2-D → 3-D
  - Much more complex
  - Properties are lost, by using straight forward extensions
  - Sequences are not efficient for complete solution space (not necessarily important)
- Needed by 3-D IC design
  - New data structures, encoding schemes, algorithms, …

Q & A

- Thank you for your attention