TEG: A New Post-Layout Optimization Method

Shuo Zhang

Wayne Dai

School of Engineering

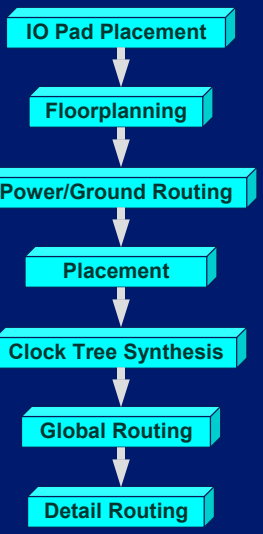
University of California at Santa Cruz

April 8, ISPD'02

Outline

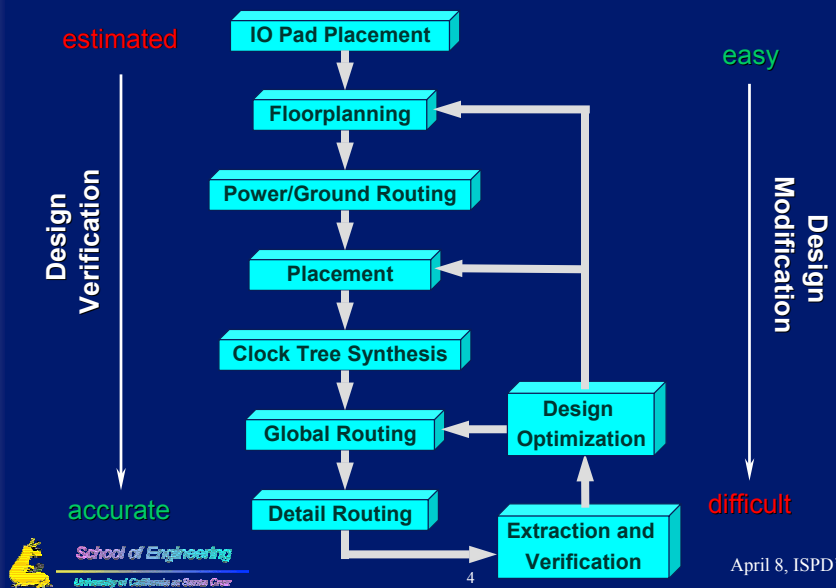
- Design optimization in post-layout stage
- TEG: A new post-layout optimization method
- Topological layout and operations in TEG
- Experimental results
- Conclusion

Physical Design Flow



Post-layout:
after detail routing, precise wire geometry information is available.

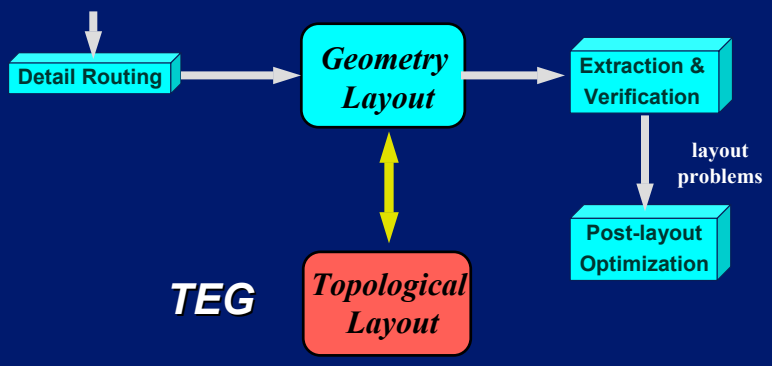
PD:Verification vs. Modification



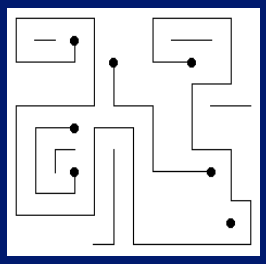
Post-layout Optimization

- Whether the design can be verified and analyzed precisely? ☑
- What and where is the design problem? ☑
- What layout optimization or modification should be performed? ☑
- How modifications can be achieved with geometry constraints all over the layout? ?

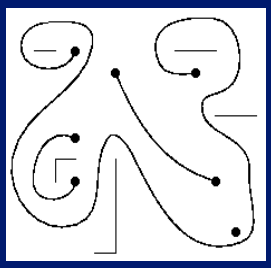
TEG: A Post-layout Optimization Method



Layout: Geometry vs. Topological



Geometry

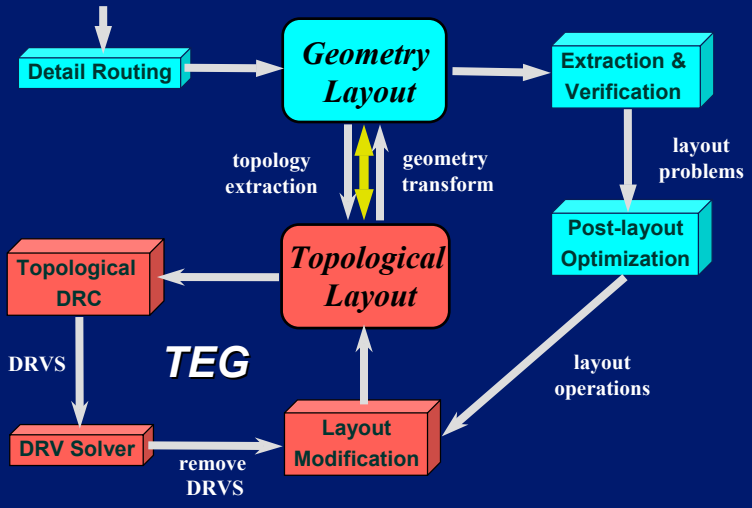


Topological

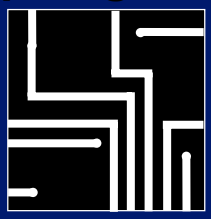
- Each net has determined geometry path.
- Net path: a sequence of hard wire segments.
- Any layout change is restrained by surrounding wire segments.

- Each net has wiring topology.
- Wiring topology: relationships between wires and layout features.
- No geometry wire, no wire constraints.

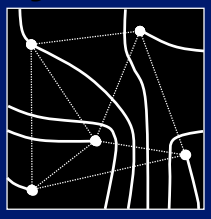
Topological Layout Optimization Flow



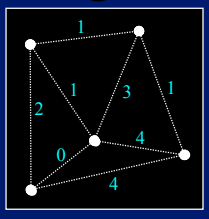
Topological Layout Encoding



Geometry



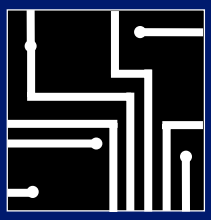
RBS



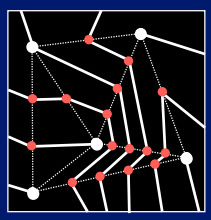
Yu's

- Rubber Band Sketch: represents wires as rubber-bands.
  - \+ Clear and straightforward layout change operation.
  - – Data size grows dramatically as design size increases.
- Yu's encoding (1997): represents wire paths as the numbers of crossing wires on each triangulation edge.
  - \+ Data size is significantly decreased.
  - – Many computations are required.
  - – Only support uniform design rule.

TEG: Triangulation Encoding Graph



Geometry



TEG

TEG: Improved from Yu's, representing wires as crossing points between wire path and edges in the layout triangulation.

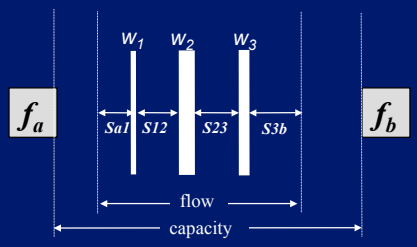
\+ Capability: Designed for the large scale applications
  A small design (3k nets/8.6k pins) RBS:TEG:Yu = 2.7:1.1:1.0
  Large scale designs TEG/Yu = 105% ~ 114%

\+ Practicability: Support multiple spacing/width, practical design rules

\+ Efficiency: Good support for topological layout operations
  Design rule check, DRV solver, layout change, topology extraction

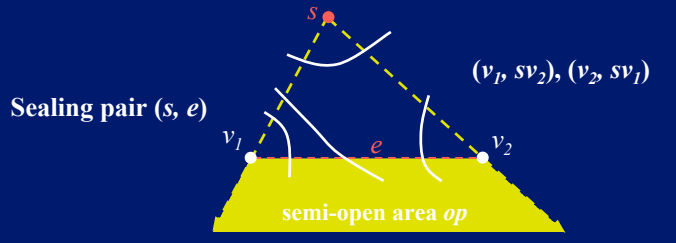
Topological Design Rule Check

- DRC: Whether a topological layout represents a valid geometry layout without any design rule violations (ROUTABILITY).
- Classic Routability Theorem (Maly 1990): A layout is routable if and only if all cuts are safe. $O(n^2)$ cuts, n – number of features



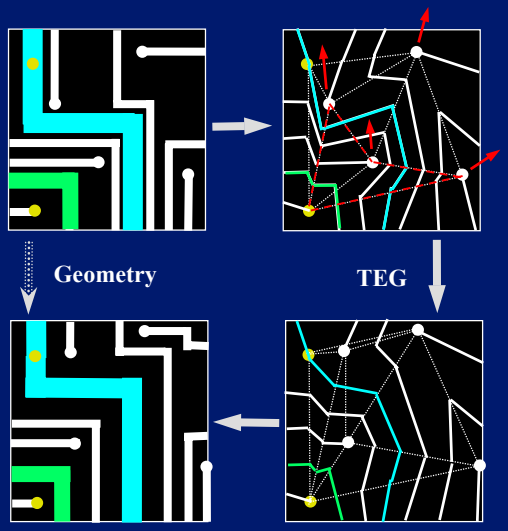
A cut is safe: flow <= capacity, otherwise it is a DRV

TEG: Sealing Pair Routability Theorem



- SP: a vertex-edge pair of a triangle in the layout triangulation.
- SP Status: unsealed -- a DRV is found in the determining process
  sealed -- each cut between s and any vertex inside *op* is safe if there is no DRV between inside *op*.
- A layout is routable if all SPs are sealed in TEG. $O(n)$ SPs
  - To determine the status of one SP, it is a recursive process with constant time in each recursive-call.
  - Experimental results show there are totally $O(n)$ recursive-calls.

TEG: Design Rule Violation Solver

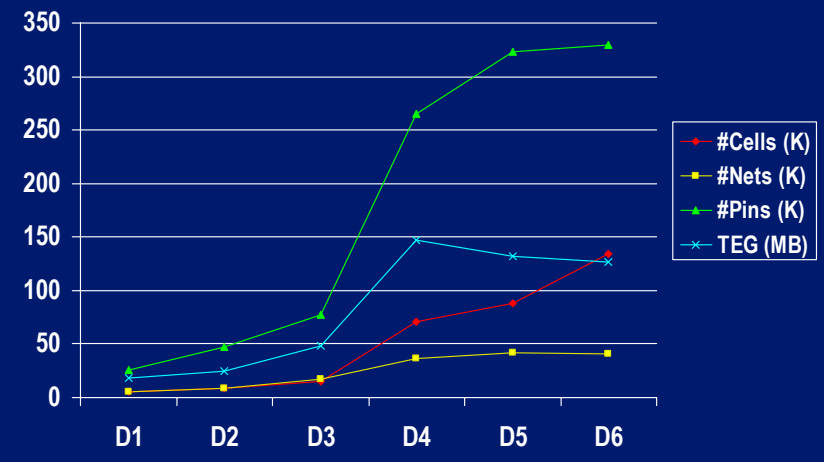


- A DRV: not enough layout resource in the local area, defined by surrounding features (terminals, vias…).
- DRV Solver: Move resource to the violation area; move DRV toward the resource area.
- DRV solvers gives each layout change as much resource as possible from the global view of the layout.

TEG: Layout Modification

- Core change operation in TEG: Vertex-moving
  - Key element in DRV solver
  - Via relocation, obstacle/cell moving, buffer insertion…
- Vertex-moving algorithm
  - Triangulation reconstruction and encoding update
  - Running time: $O(m \cdot n)$, n – number of wires, m – number of vertices influenced by moving.

Experimental Results



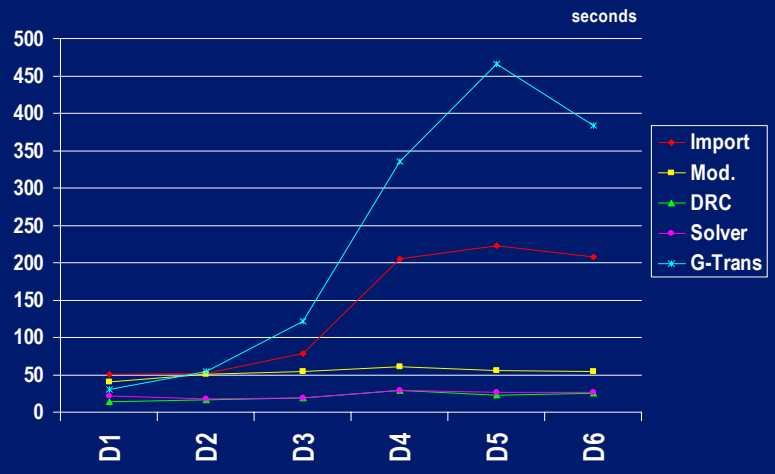
Examples: Cells (4.9K~134K), Nets (5.0K~41.8K), Pins (26K~330K)

Running Time - Setup

1. Import the original routed design (LEF/DEF) into TEG.
2. Modification: 500 randomly selected via-moving
3. Topological DRC then DRV solver
4. Geometry Transform.

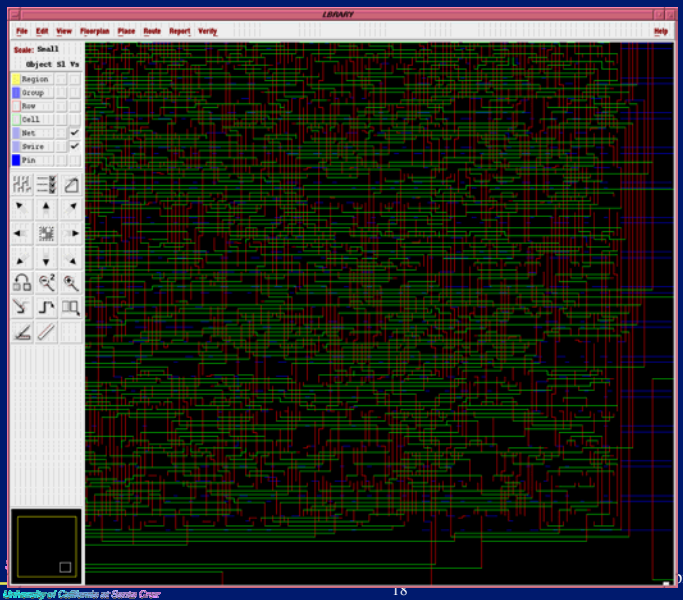
PIII 550, 768MB memory, Linux platform

Running Time

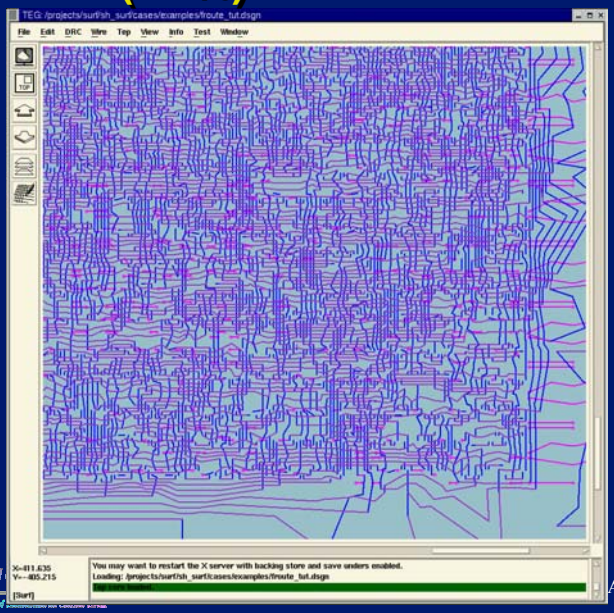


Mod: 5000 via-moving, DRC: ~30000 SPs, Solver: 10 DRVs

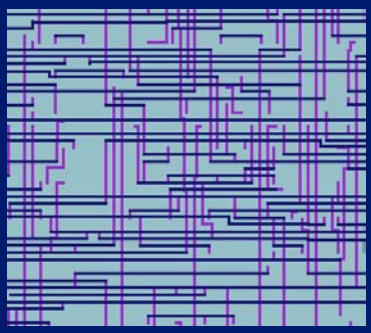
Layout D1 (~5%) in Cadence SE



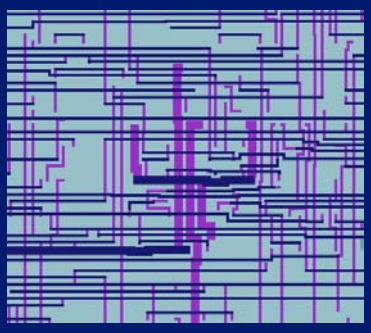
Layout D1 (~5%) in TEG



Wire Sizing by TEG



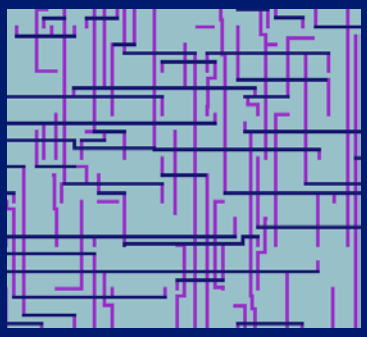
Original Layout



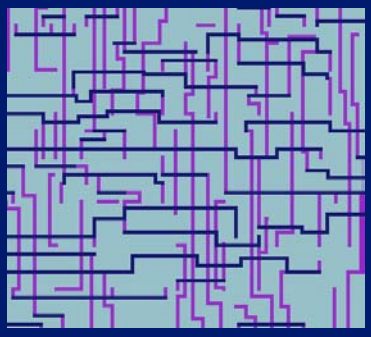
After wire sizing

Fix crosstalk-delay, reduce IR-drop …

Wire Distribution by TEG



Original Layout



After wire distribution

Reduce wire density difference: decrease CMP process variation, improve yield and manufacturability…

Conclusion

- **In post-layout stage, the challenge is how to achieve preferred layout modifications: even the small change is not acceptable due to the limited local resource.**
- **TEG: A new post-layout optimization method which processes the layout topologically, with an improved topological layout encoding mode and a set of layout operation procedures.**
- **TEG provides an incremental layout modification environment for post-layout optimizations.**
- **TEG is efficient and effective for industry IC designs.**