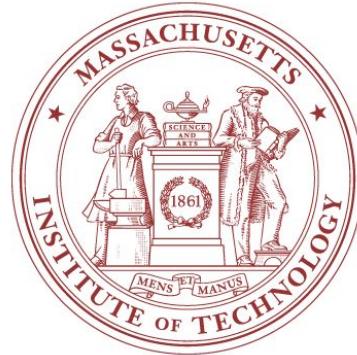


# L14 – Physical Design

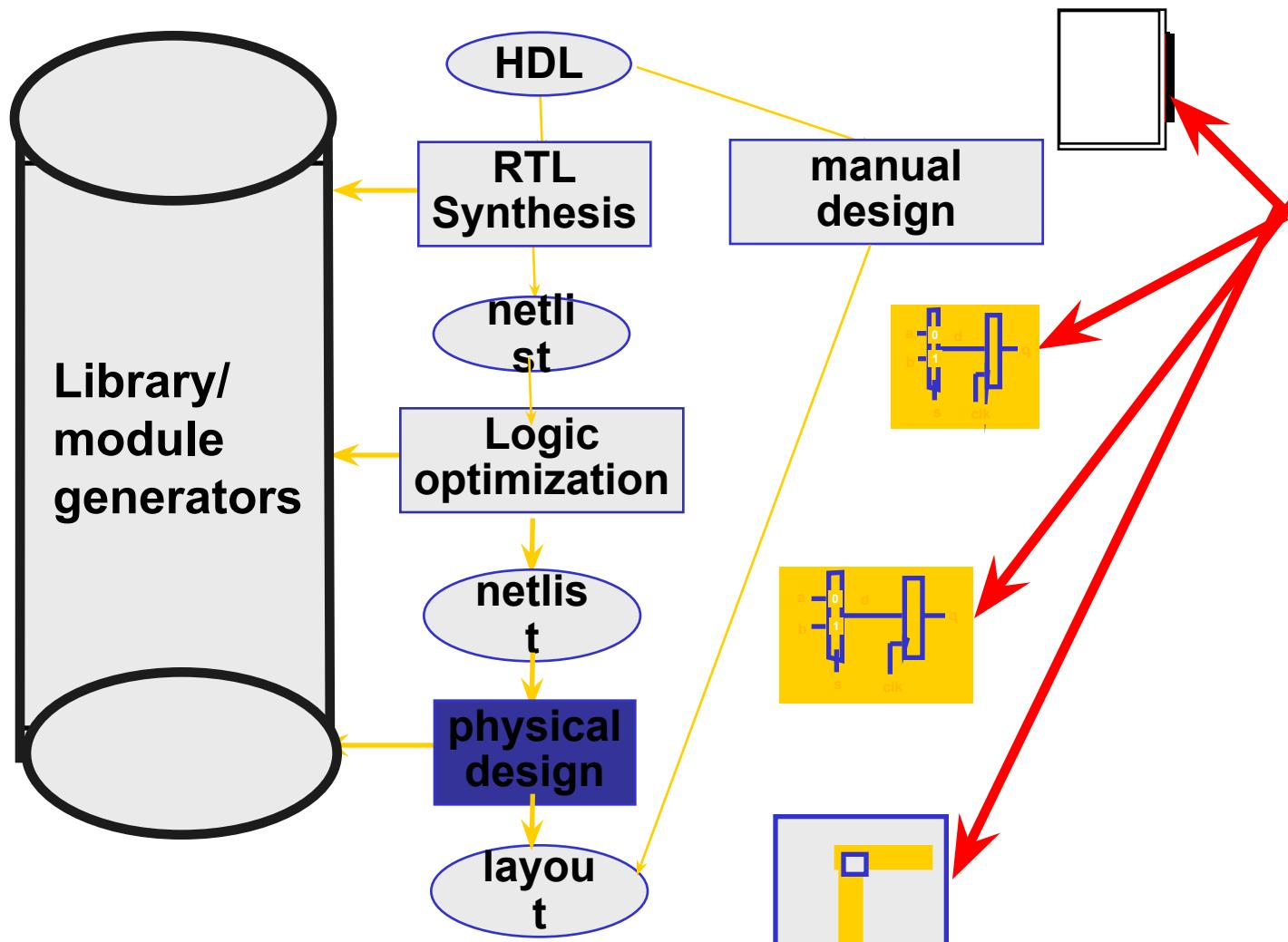
---

6.375 Spring 2007  
Ajay Joshi

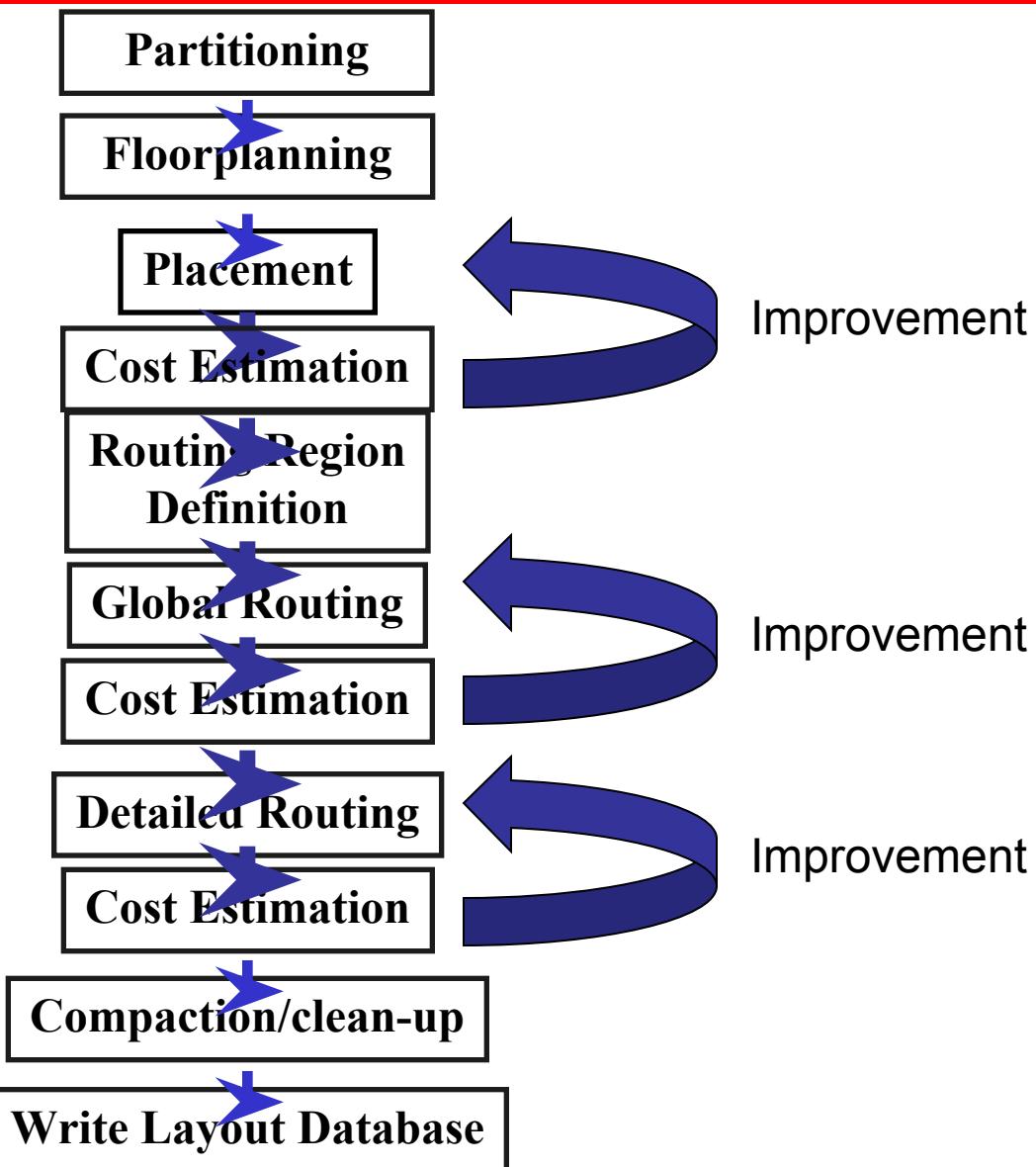


Massachusetts Institute of Technology

# RTL design flow



# Physical design – overall flow

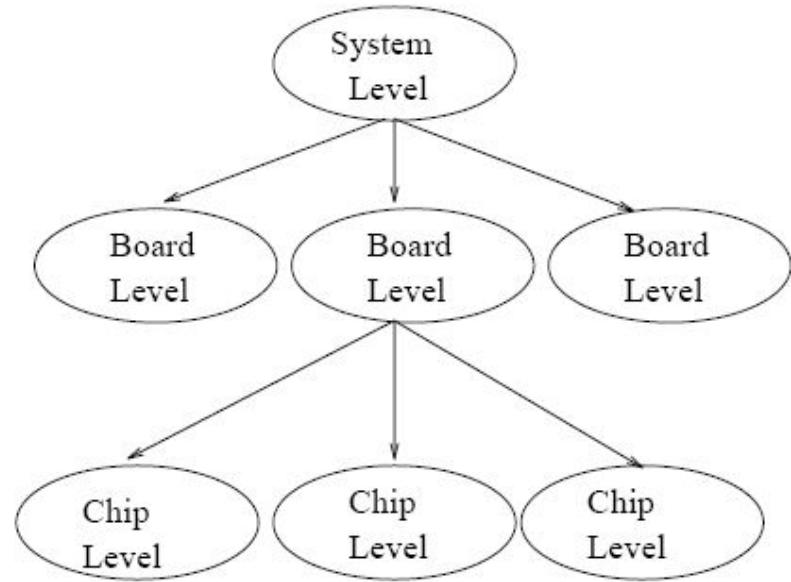
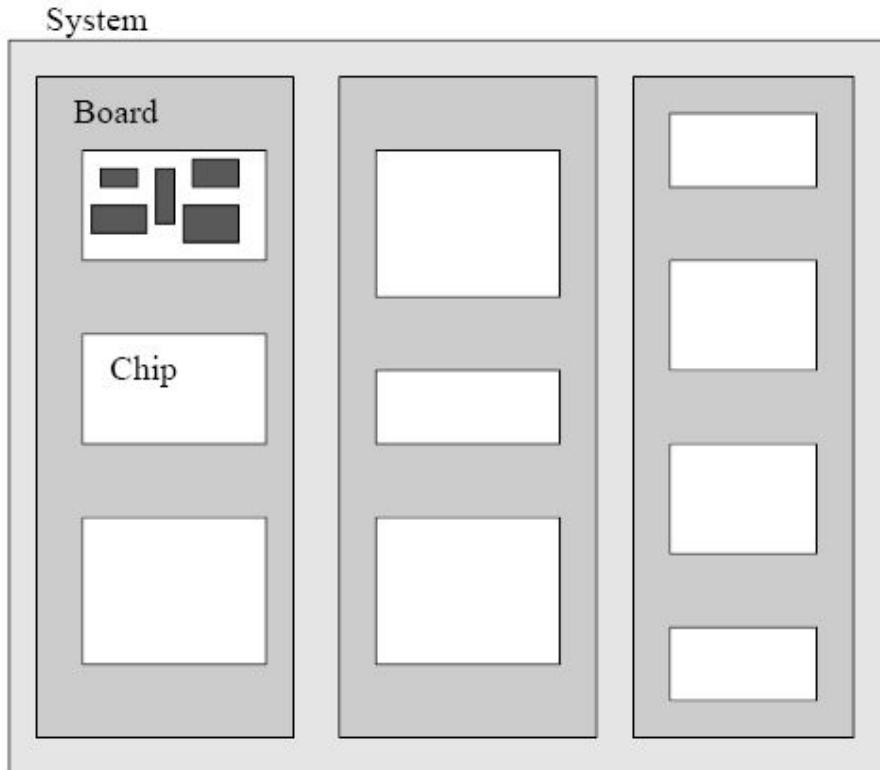
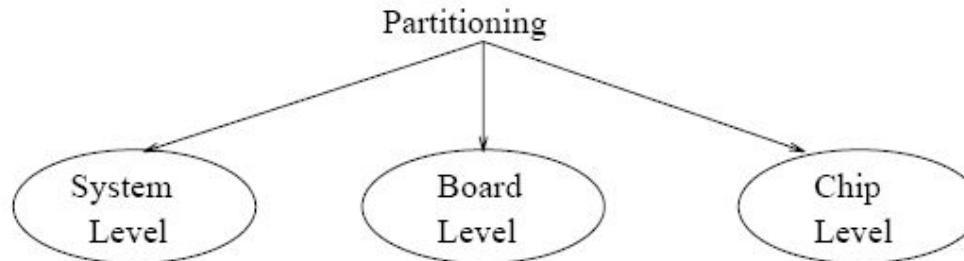


# Partitioning

---

- ❑ Decompose a large complex system into smaller subsystems
- ❑ Decompose hierarchically until each subsystem is of manageable size
- ❑ Design each subsystem separately to speed up the process
- ❑ Minimize connection between two subsystems to reduce interdependency

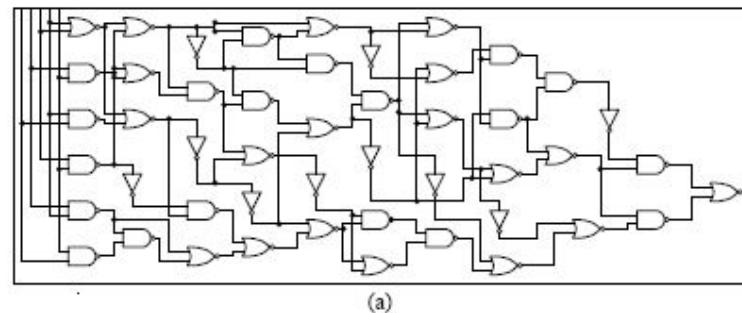
# Partitioning at different levels\*



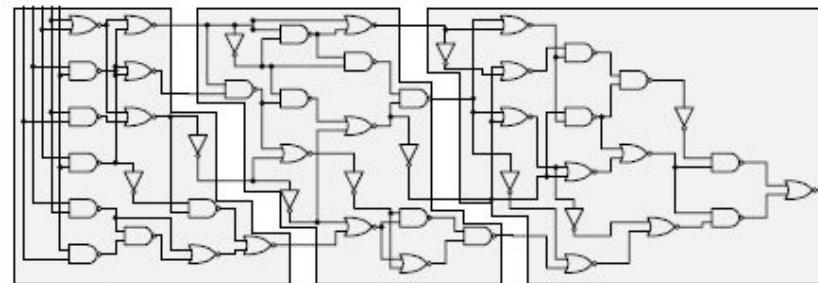
\* © Sherwani 92

# Partitioning example\*

Input size = 48



(a)



(b)

Cut 1 = 4

Size 1 = 15

Cut 2 = 4

Size 2 = 16

Size 3 = 17

\* © Sherwani 92

# Partitioning problem

---

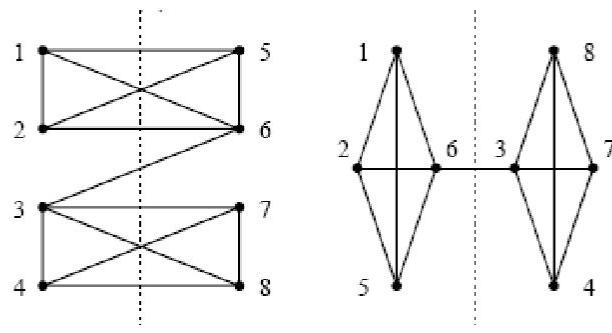
- Objective:
  - Minimize interconnections between partitions
  - Minimize delay due to partitioning
- Constraints
  - Number of terminals in each subsystem (Count ( $V_i$ ) <  $T_i$ )
  - Area of each partition ( $A_i^{\min} < \text{Area}(V_i) < A_i^{\max}$ )
  - Number of partitions ( $K_{\min} < k < K_{\max}$ )
  - Critical path should not cut boundaries

# Kernighan-Lin algorithm

---

- Input: Graph representation of the circuit
- Output: Two subsets of equal sizes
- Bisection algorithm :
  - Initial bisection
  - Vertex pairs which gives the largest decrease in cutsize are exchanged
  - Exchanged vertices are locked
  - If no improvement is possible and some vertices are still unlocked then vertices which give smallest increase are exchanged

# K-L algorithm example\*



(a) Initial Bisections

(b) Final Bisections

$i$	Vertex Pair	$g(i)$	$\sum_{j=1}^i g(i)$	Cutsize
0	-	-	-	9
1	(3,5)	3	3	6
2	(4,6)	5	8	1
3	(1,7)	-6	2	7
4	(2,8)	-2	0	9

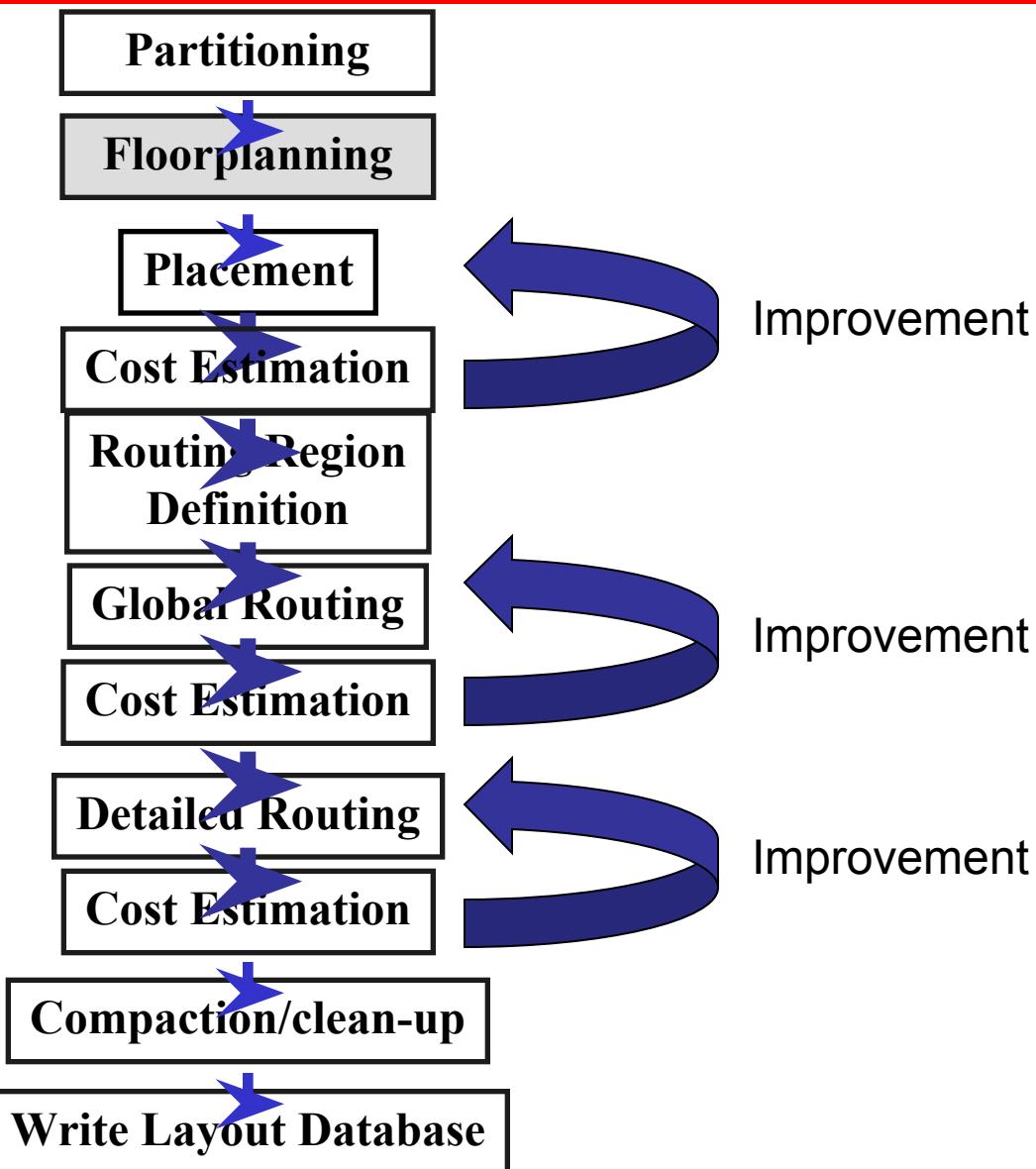
\* © Sherwani 92

# Partitioning methods

---

- Top-down partitioning
  - Iterative improvement
  - Spectral based
  - Clustering methods
  - Network flow based
  - Analytical based
  - Multi-level
- Bottom-up clustering
  - Unit delay model
  - General delay model
  - Sequential circuits with retiming

# Physical design – overall flow



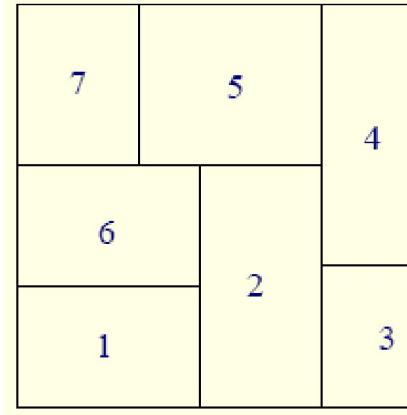
# Floorplanning

---

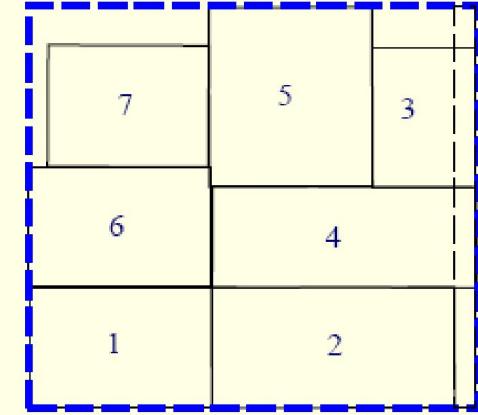
- ❑ Output from partitioning used for floorplanning
- ❑ Inputs:
  - Blocks with well-defined shapes and area
  - Blocks with approximated area and no particular shape
  - Netlist specifying block connections
- ❑ Outputs:
  - Locations for all blocks

# Floorplanning problem\*

- Objectives
  - Minimize area
  - Reduce wirelength
  - Maximize routability
  - Determine shapes of flexible blocks
- Constraints
  - Shape of each block
  - Area of each block
  - Pin locations for each block
  - Aspect ratio



An optimal floorplan,  
in terms of area



A non-optimal floorplan

\* Sung Kyu Lim

# Slicing floorplan sizing\*

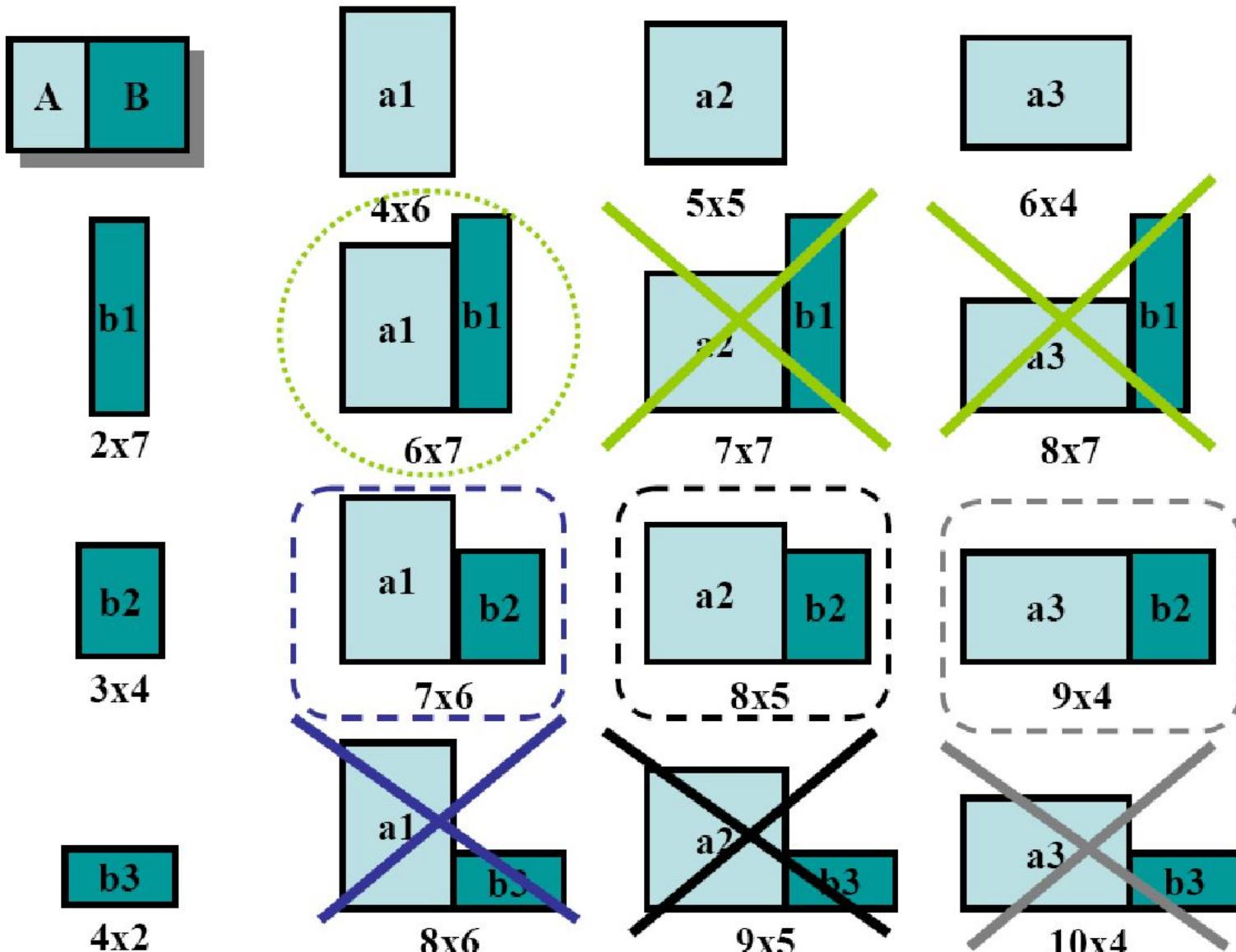
---

- General case: all modules are soft macros
- Phase 1: bottom-up
  - Input – floorplan tree, modules shapes
  - Start with a sorted shapes list of modules
  - Perform vertical\_node\_sizing and horizontal\_node\_sizing
  - On reaching the root node, we have a list of shapes, select the one that is best in terms of area
- Phase 2: top-down
  - Traverse the floorplan tree and set module locations

\* Sung Kyu Lim



# Sizing example\*

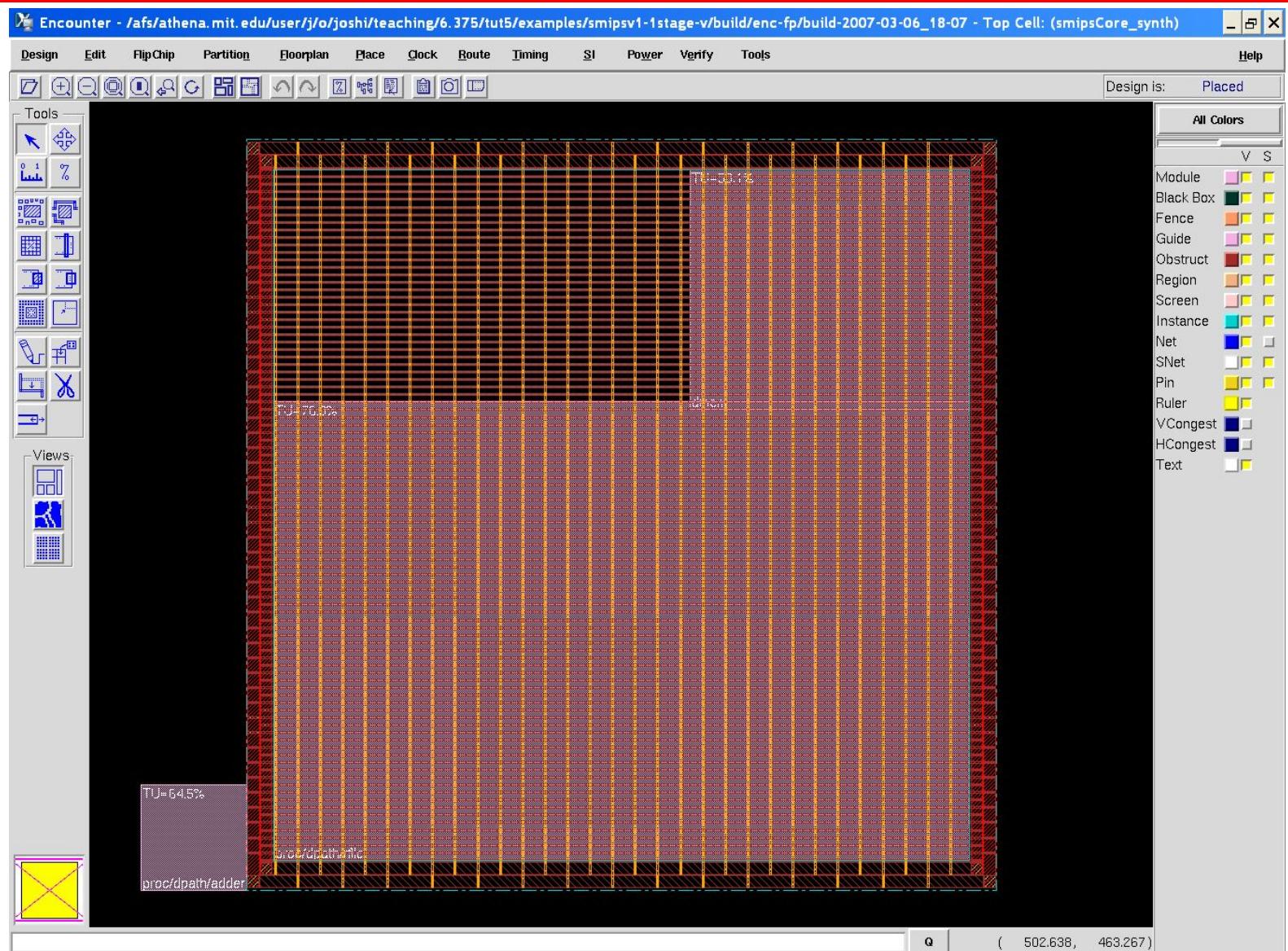


# Floorplanning Algorithms

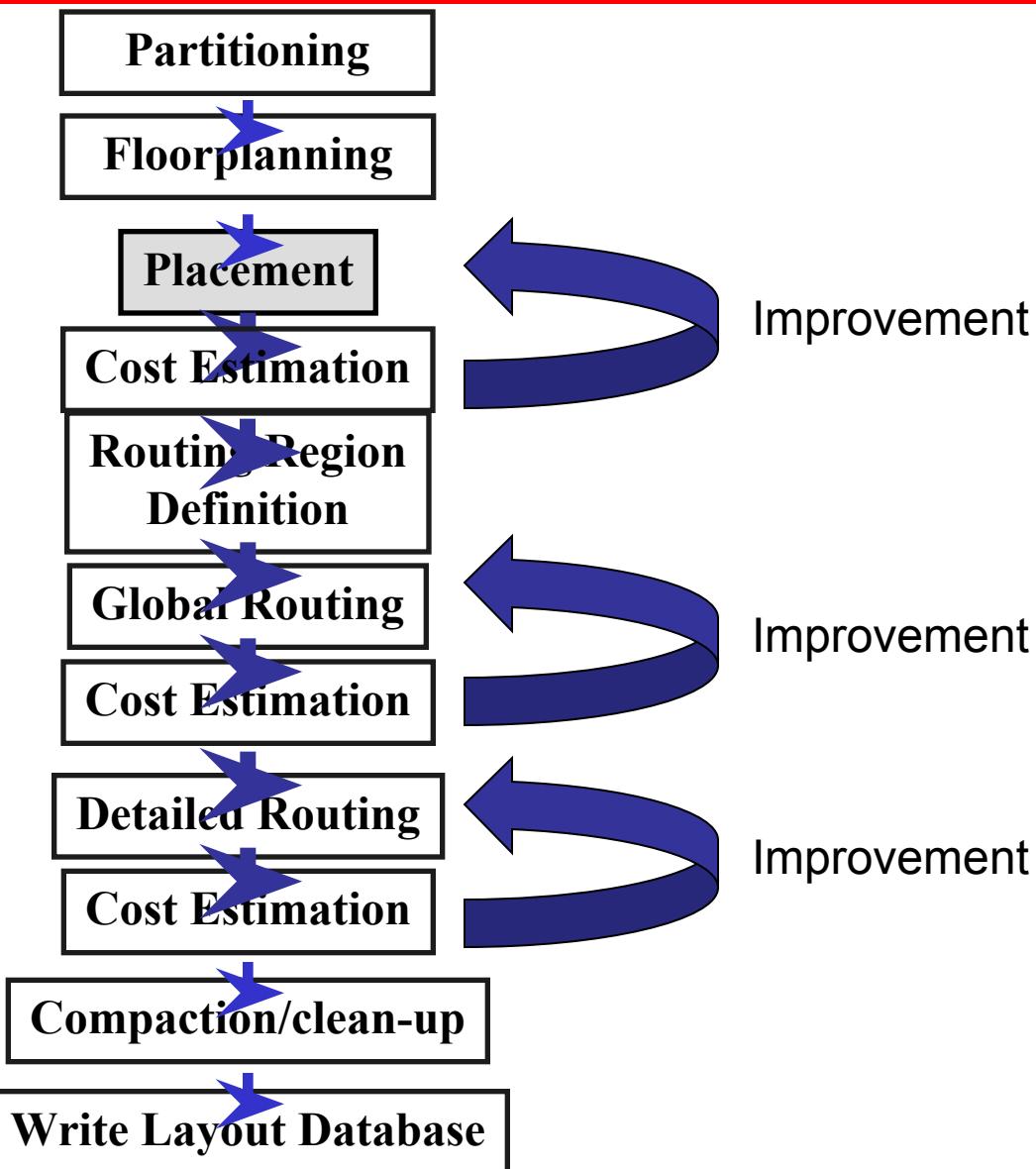
---

- Stockmeyer algorithm
- Simulated annealing
- Linear programming
- Sequence-pair based floorplanning

# Floorplanning - Encounter



# Physical design – overall flow

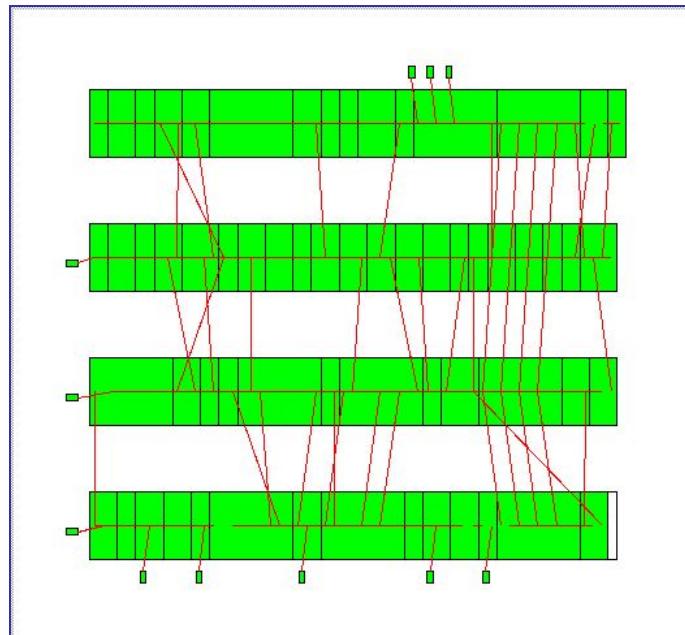


# Placement

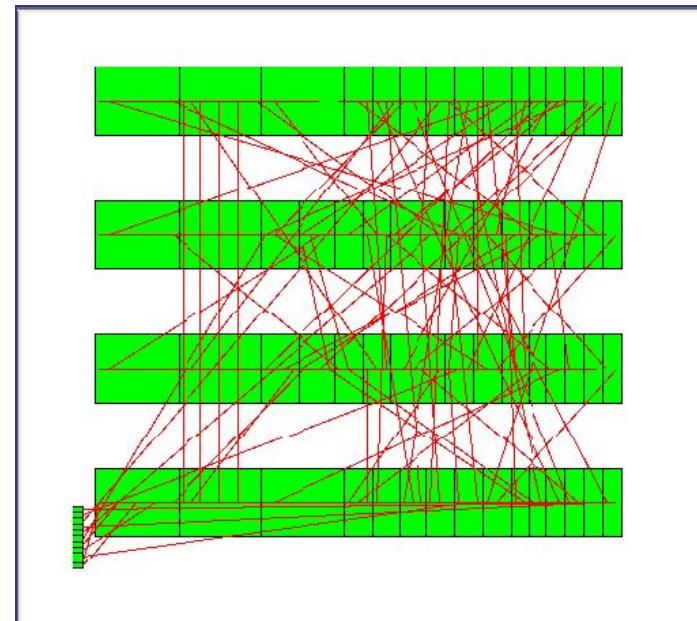
---

- The process of arranging circuit components on a layout surface
- Inputs : Set of fixed modules, netlist
- Output : Best position for each module based on various cost functions
- Cost functions include wirelength, wire routability, hotspots, performance, I/O pads

# Good placement vs Bad placement\*



- Good placement
  - No congestion
  - Shorter wires
  - Less metal levels
  - Smaller delay
  - Lower power dissipation



- Bad placement
  - Congestion
  - Longer wire lengths
  - More metal levels
  - Longer delay
  - Higher power dissipation

\* S. Devadas



# Simulated annealing algorithm

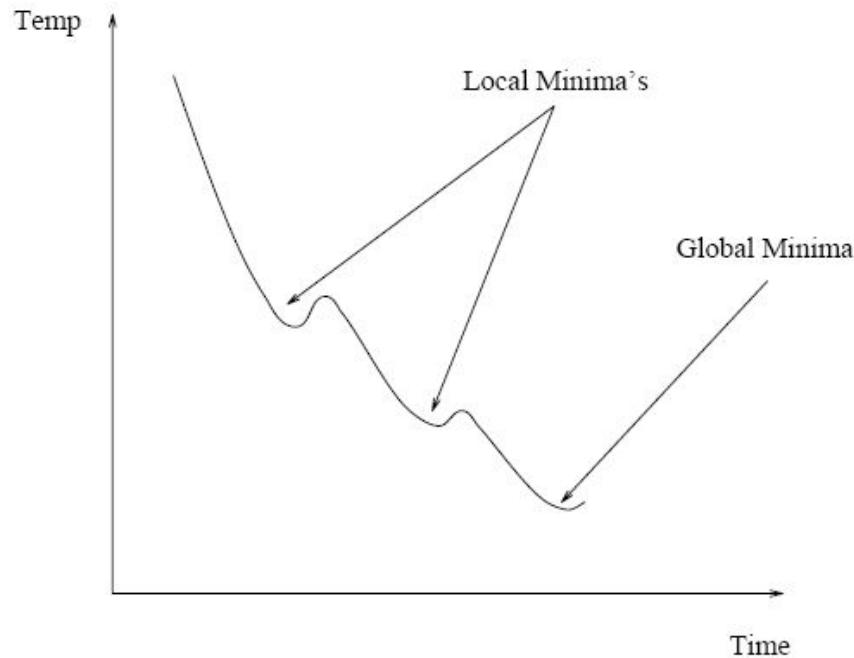
---

- Global optimization technique
- Cooling schedule is adopted
- An action performed at each new temperature
- Estimate the cost associated with an action
- If new cost < old cost accept the action
- If new cost > old cost then accept the action with probability  $p$
- Probability  $p$  depends on a temperature schedule – Higher  $p$  at higher temperature

# Annealing curve

---

## The Annealing curve



\* © Sherwani 92

# Placement using simulated annealing

---

- Use initial placement results – e.g. random placement
- Two stage process\*
  - Stage 1
    - Modules moved between different rows and same row
    - Module overlaps allowed
    - Stage two begins when temperature falls below a certain value
  - Stage 2
    - Module overlaps removed
    - Annealing continued, but interchange adjacent modules in the same row

\* Sechen DAC86

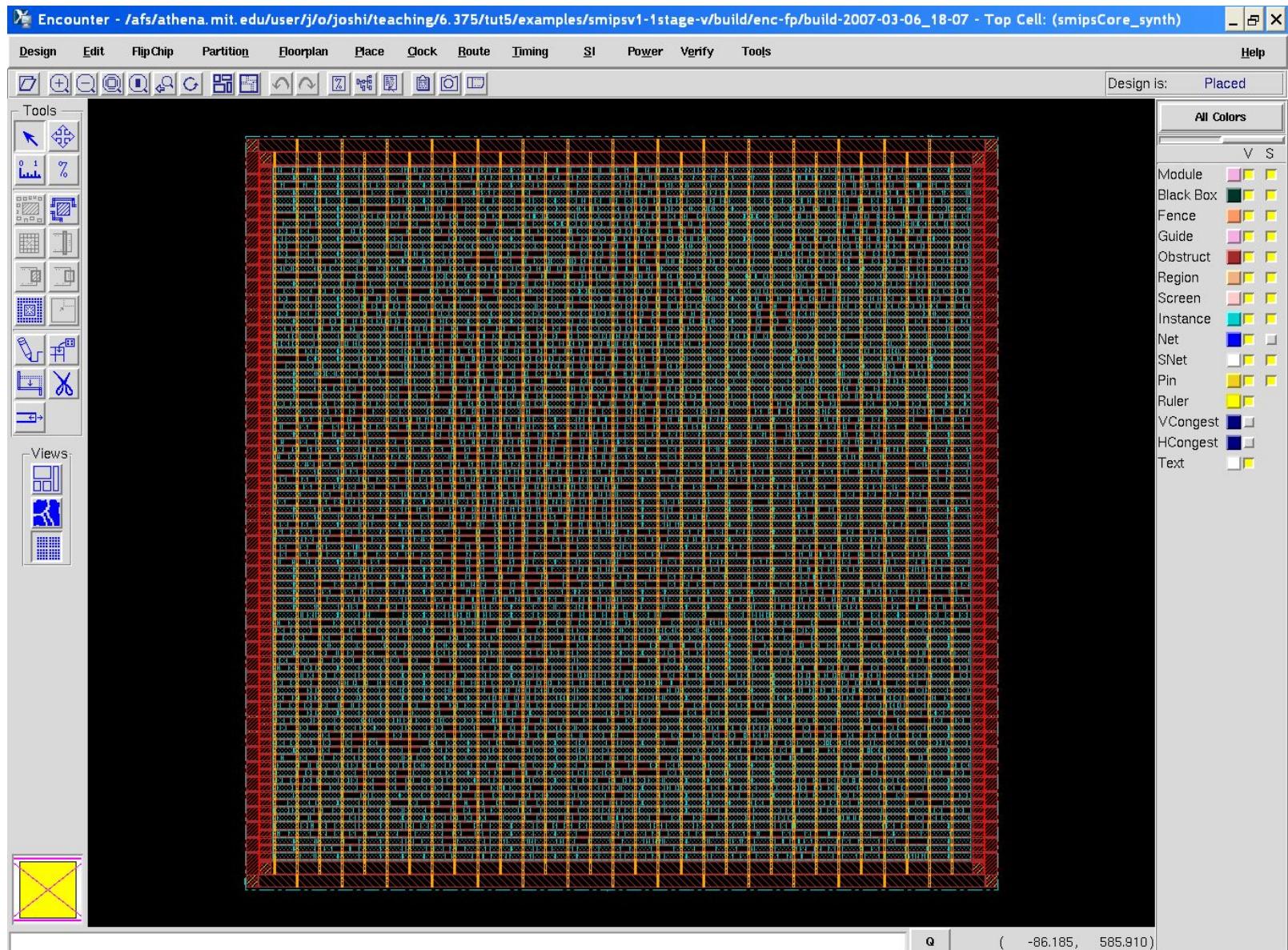


# Placement methods

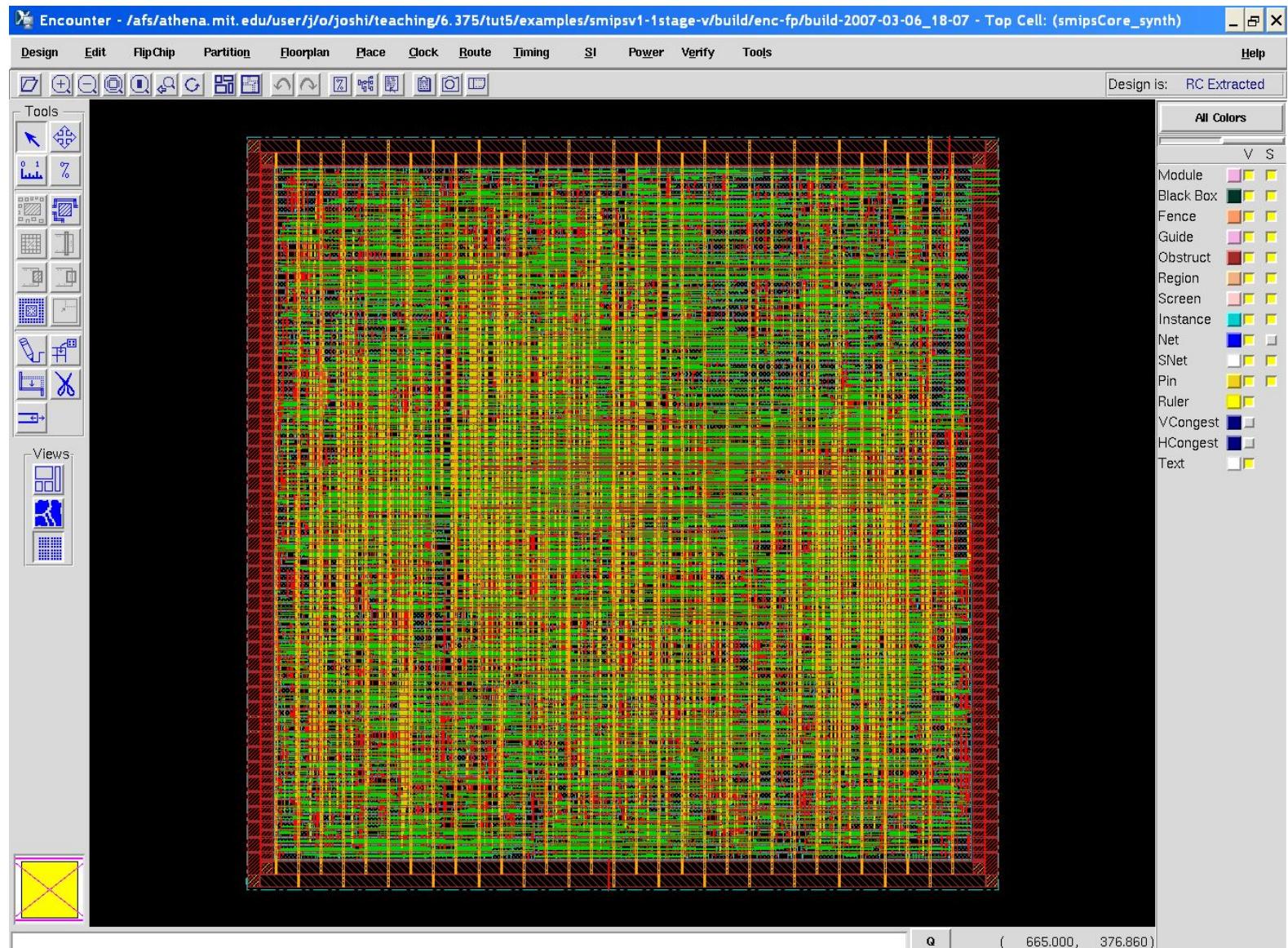
---

- Constructive methods
  - Cluster growth algorithm
  - Force-directed method
  - Algorithm by Goto
  - Min-cut based algorithm
- Iterative improvement methods
  - Pairwise exchange
  - Simulated annealing – Timberwolf
  - Genetic algorithm
- Analytical methods
  - Gordian, Gordian-L

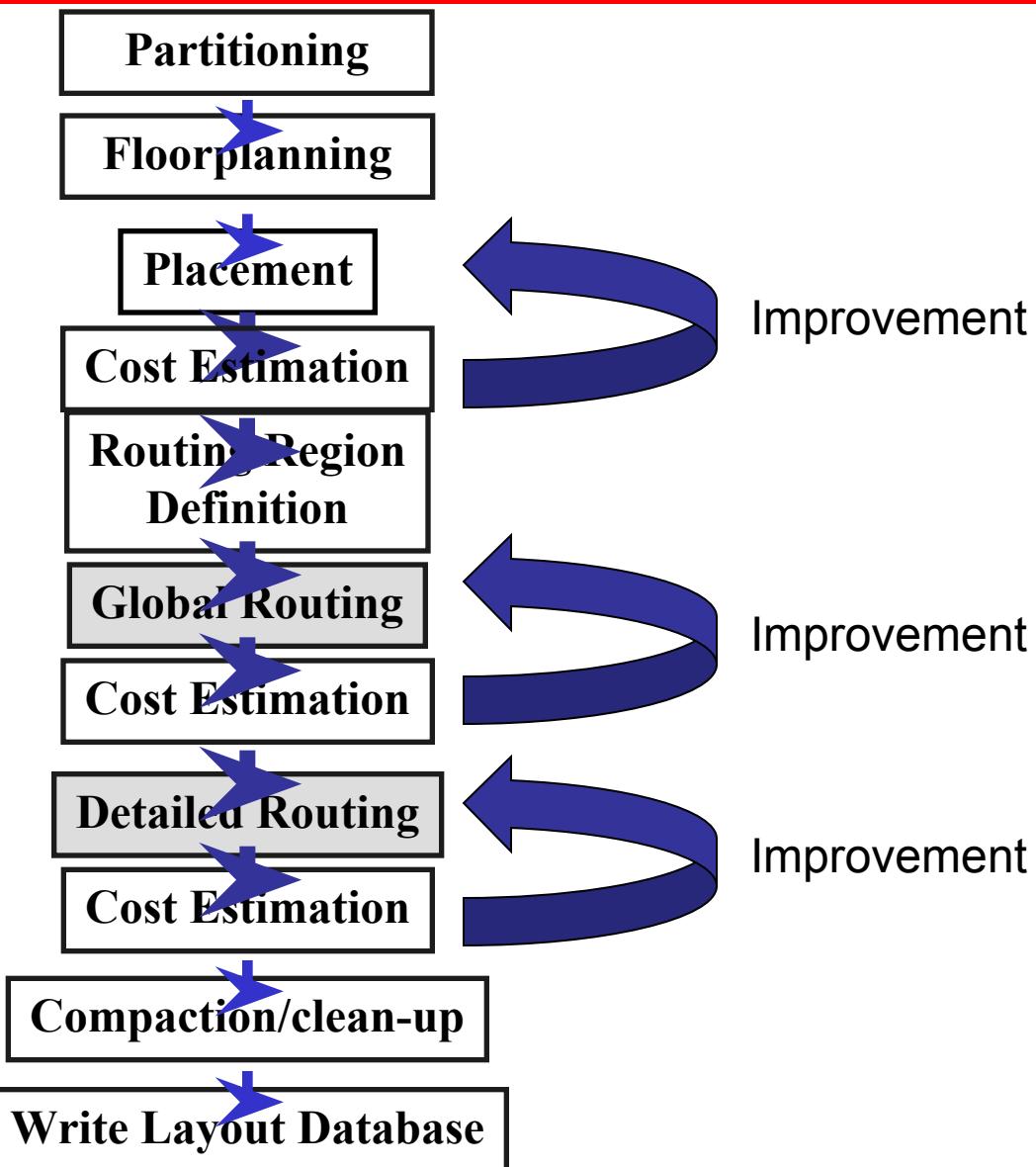
# Placement - Encounter



# Optimized placement - Encounter



# Physical design – overall flow



# Routing

---

- Connect the various standard cells using wires
- Input:
  - Cell locations, netlist
- Output:
  - Geometric layout of each net connecting various standard cells
- Two-step process
  - Global routing
  - Detailed routing

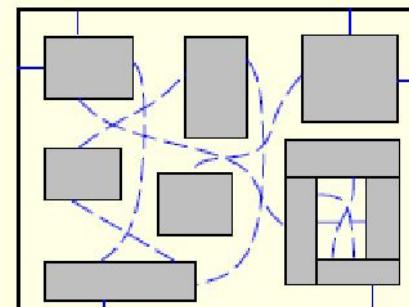
# Global routing vs detailed routing\*

## Routing

placement

- Generates a "loose" route for each net.
- Assigns a list of routing regions to each net without specifying the actual layout of wires.

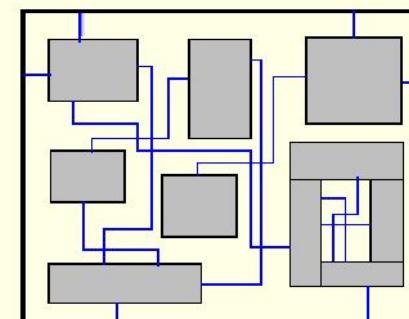
global routing



detailed routing

- Finds the actual geometric layout of each net within the assigned routing regions.

compaction



\* Sung Kyu Lim

# Routing problem formulation

---

## □ Objective

- 100% connectivity of a system
- Minimize area
- Minimize wirelength

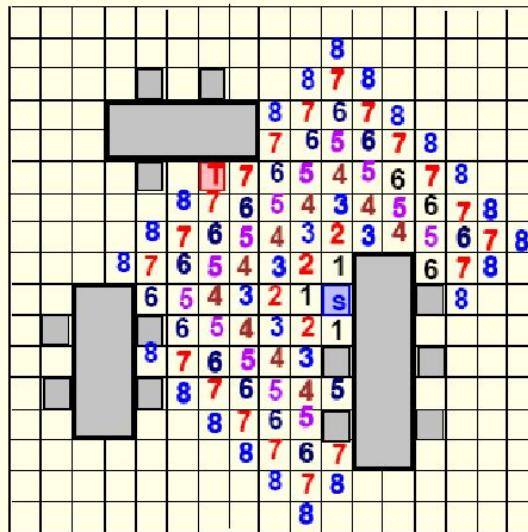
## □ Constraints

- Number of routing layers
- Design rules
- Timing (delay)
- Crosstalk
- Process variations

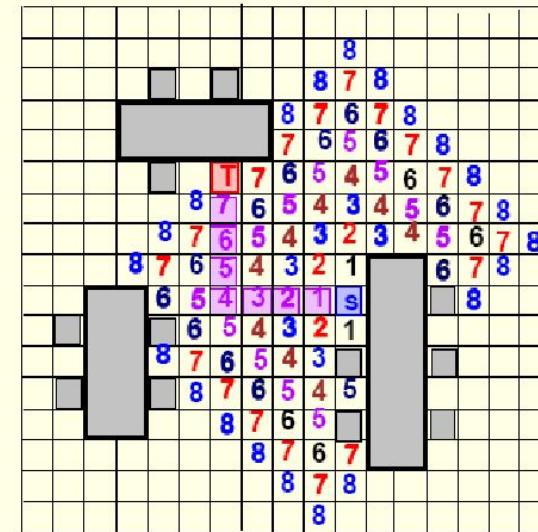
# Maze routing - example

## Lee Algorithm

- Find a path from  $S$  to  $T$  by “wave propagation”.



Filing



Retrace

- Time & space complexity for an  $M \times N$  grid:  $O(MN)$  (**huge!**)

\* Sung Kyu Lim

# Maze routing

---

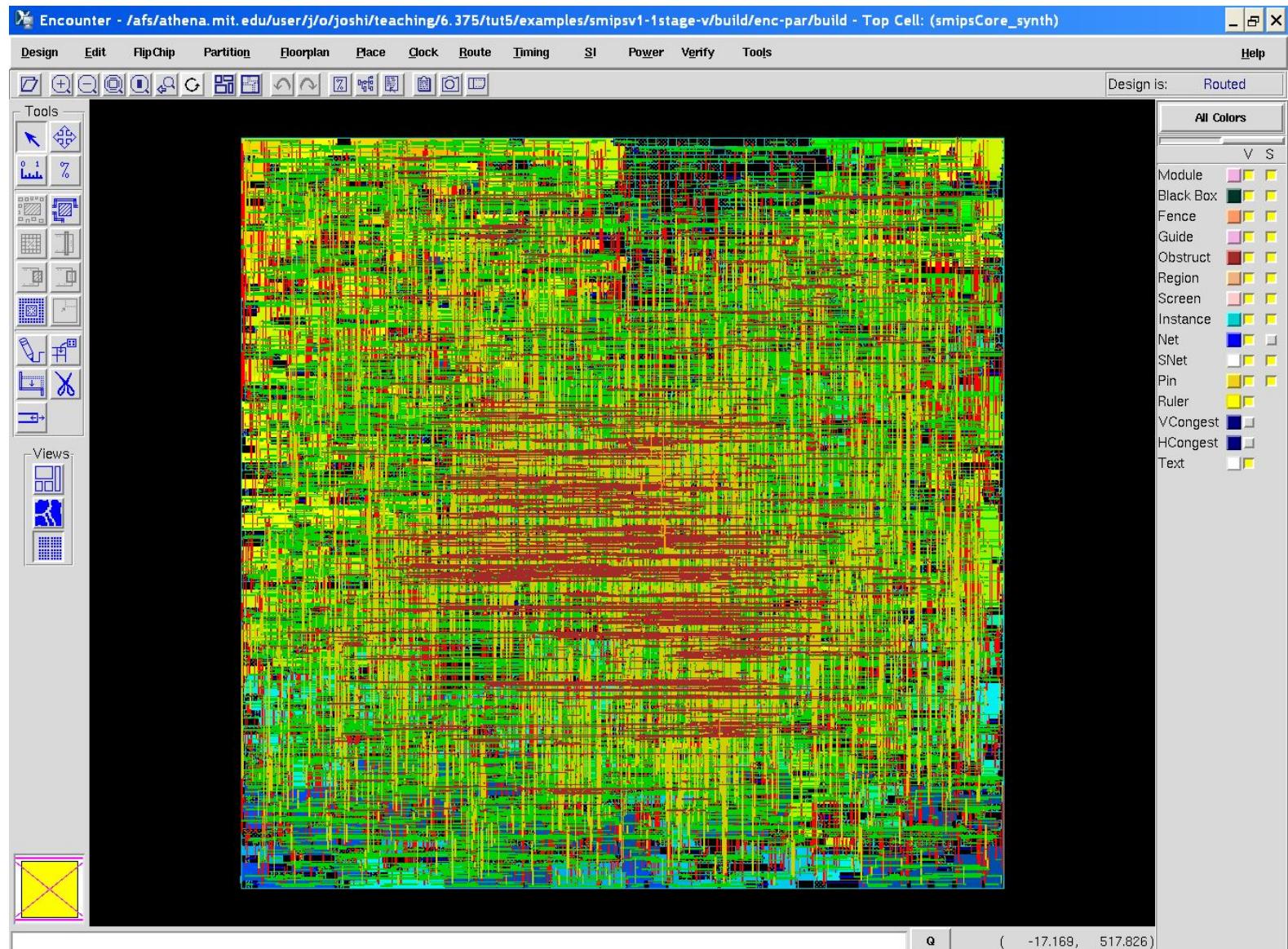
- Mainly for single-layer routing
- Strengths
  - Finds a connection between two terminals if it exists
- Weakness
  - Large memory required as dense layout
  - Slow
- Application – global routing, detailed routing

# Routing algorithms

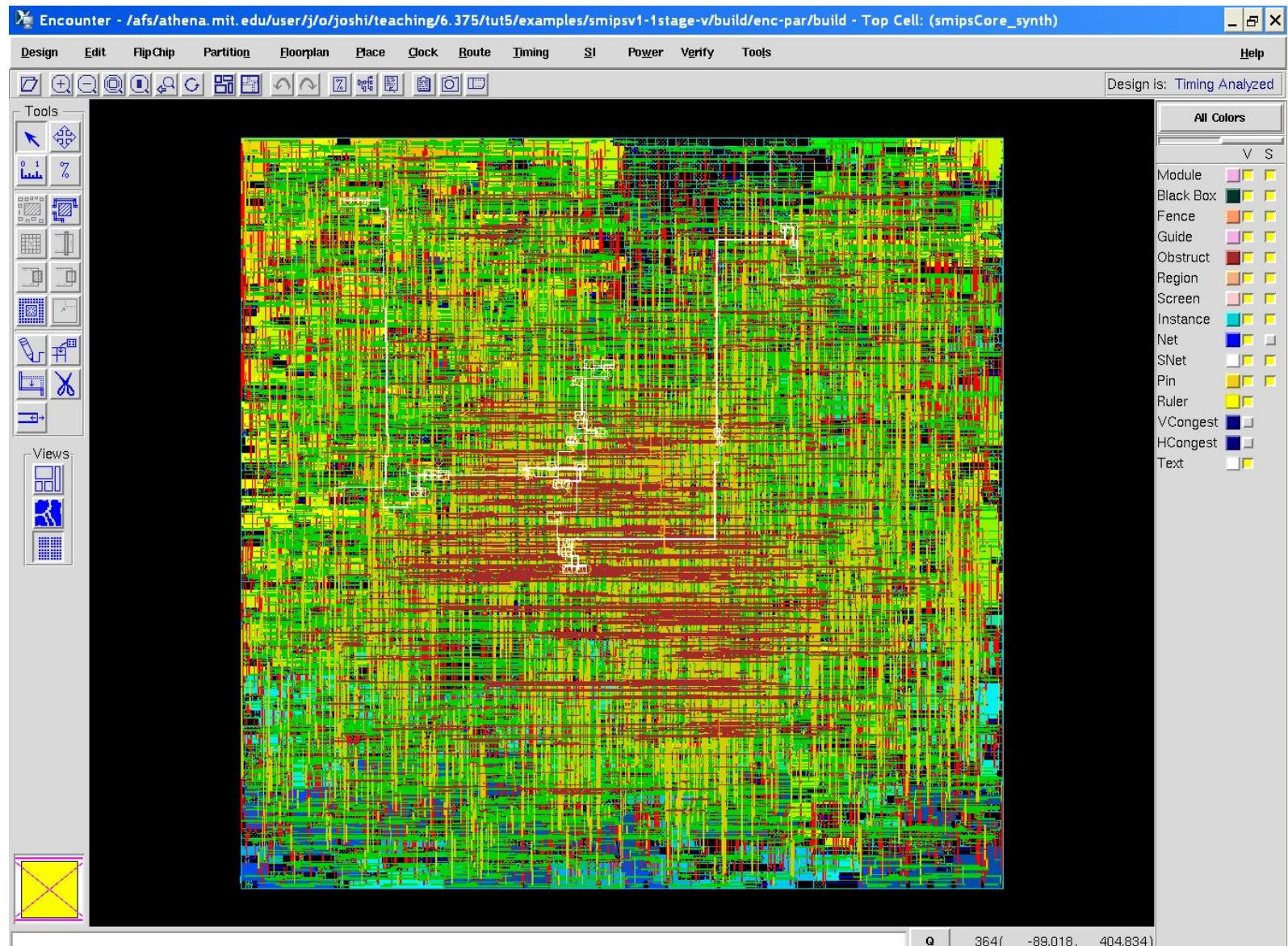
---

- Global routing
  - Maze routing
  - Cong/Preas algorithm
  - Spanning tree algorithm
  - Steiner tree algorithm
- Detailed routing
  - 2-L Channel routing: Basic left-edge algorithm
  - Y-K algorithm

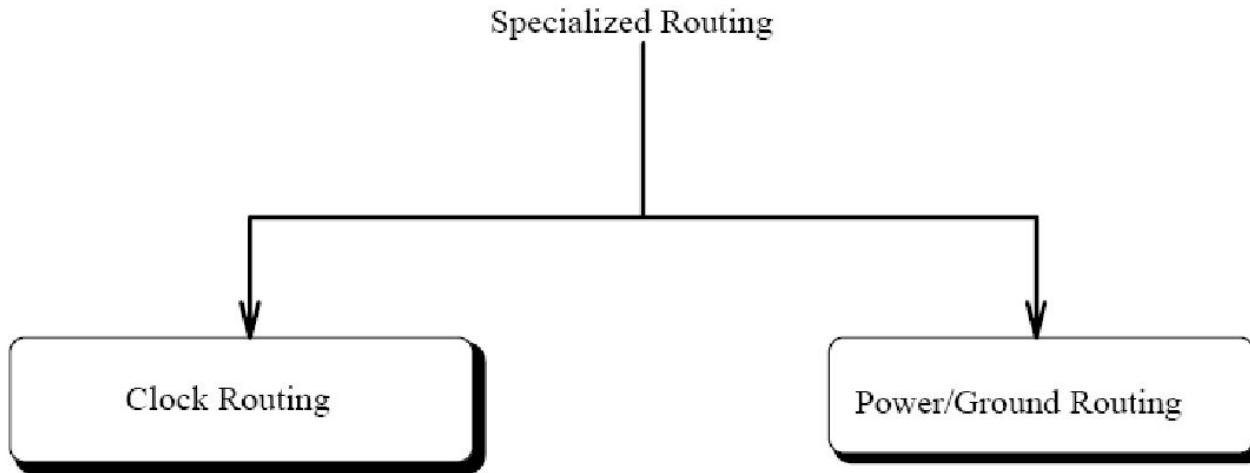
# Detailed routing - Encounter



# Critical path - Encounter

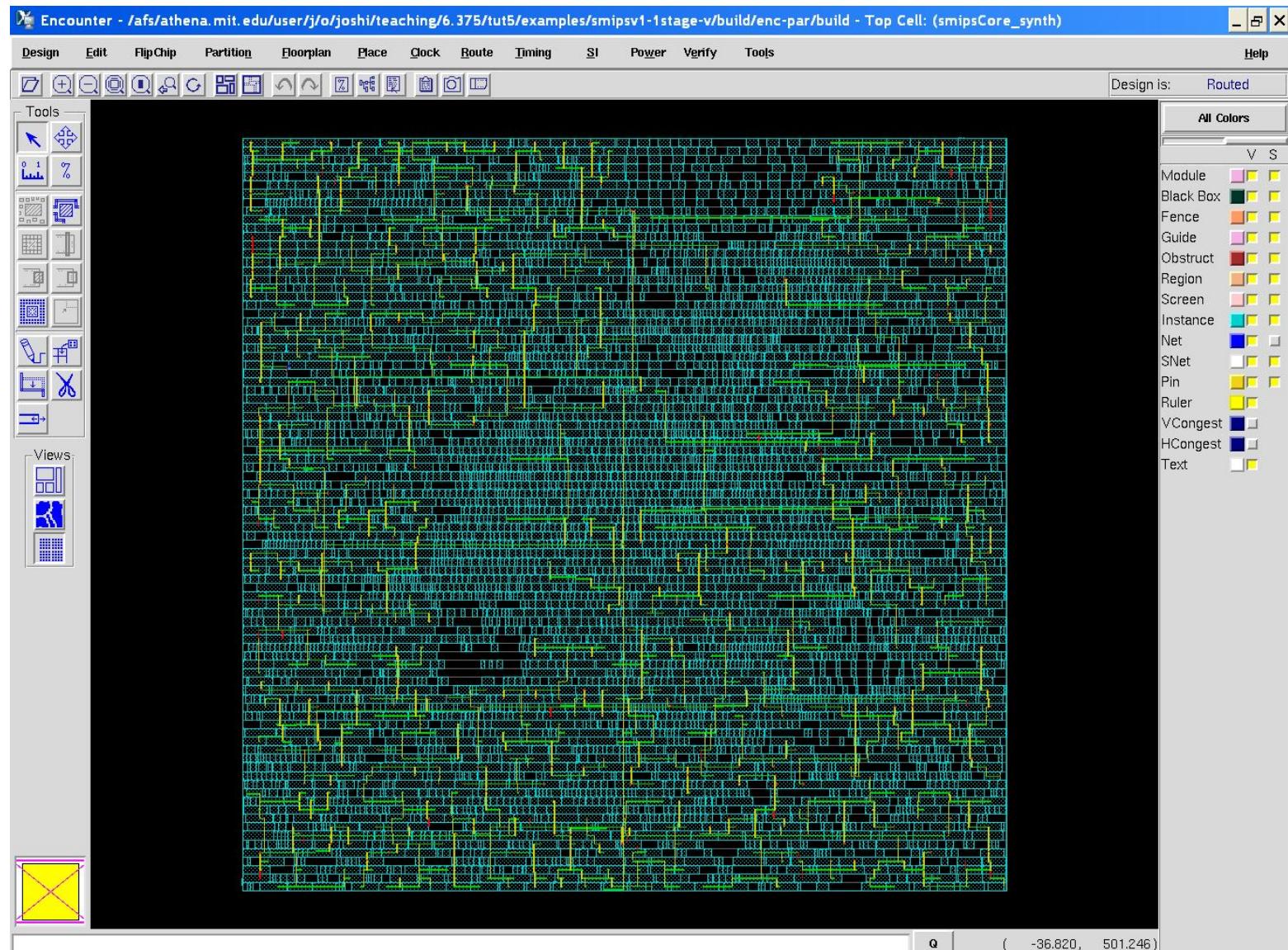


# Specialized routing

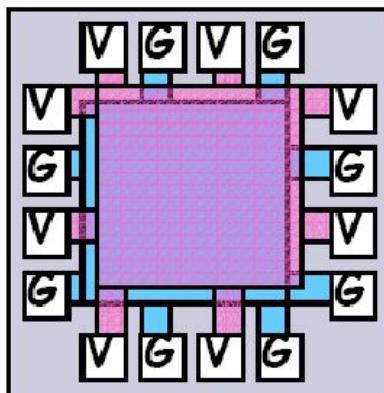
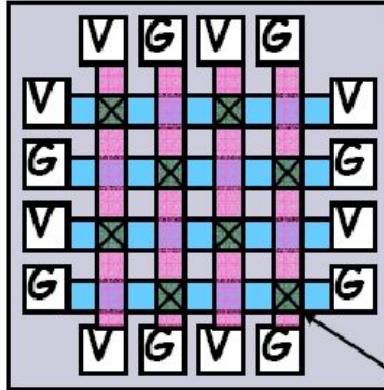
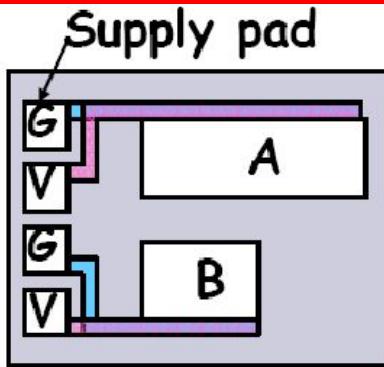


- Routing clock nets such that
  - clock arrives simultaneously
  - clock delay is minimum
- Routing of power/ground net on
  - Low resistance metal lines

# Clock distribution - Encounter



# Power routing\*



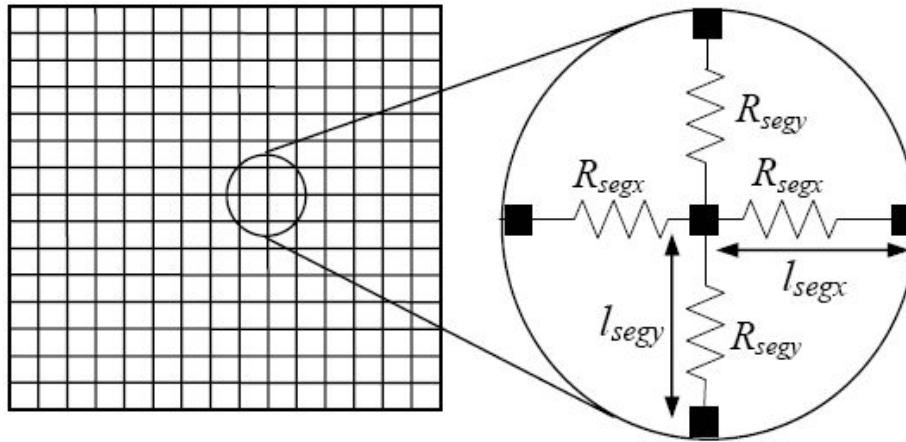
Routed power distribution on two stacked layers of metal (one for VDD, one for GND). OK for low-cost, low-power designs with few layers of metal.

Power Grid. Interconnected vertical and horizontal power bars. Common on most high-performance designs. Often well over half of total metal on upper thicker layers used for VDD/GND.

Via

Dedicated VDD/GND planes. Very expensive. Only used on Alpha 21264. Simplified circuit analysis. Dropped on subsequent Alphas.

# Power distribution issues



Kaveh Shakeri

- Noise in power supply
  - IR drop (static)
  - L di/dt (transient)
- Electromigration
- Solution: decoupling capacitance, wire material

# Summary

---

- ❑ Looked at the physical design flow
- ❑ Involved several steps
  - Partitioning
  - Floorplanning
  - Placement
  - Routing
- ❑ Each step can be formulated as an optimization problem
- ❑ Need to go through 2 or more iterations in each step to generate an optimized solution