

# Global routing

Bill Swartz

InternetCAD.com



# Disclaimer

- Paper written in 5 days
- Please forgive any errors, typos, omissions, and possible legal transgressions

# Background

- TimberWolf Systems / InternetCAD.com formed in 1994.
- Concentrated on producing high quality results for small chips and block designs.
- Claim to fame: 486, Pentium series, Alpha, and Centrino

# But...

- One company no longer uses the product and one no longer exists...
- Business is now predominantly DRAM.
- Wrote grouter (2D) and igrouter (3D).
- Helped write TimberWolfGR and SGGR.

# Outline

- Introduction
- Applications
- Previous work / state-of-the art
- Advanced objectives and requirements
- Alternative methodologies
- Itools global router
- Summary

# Physical Design Stages

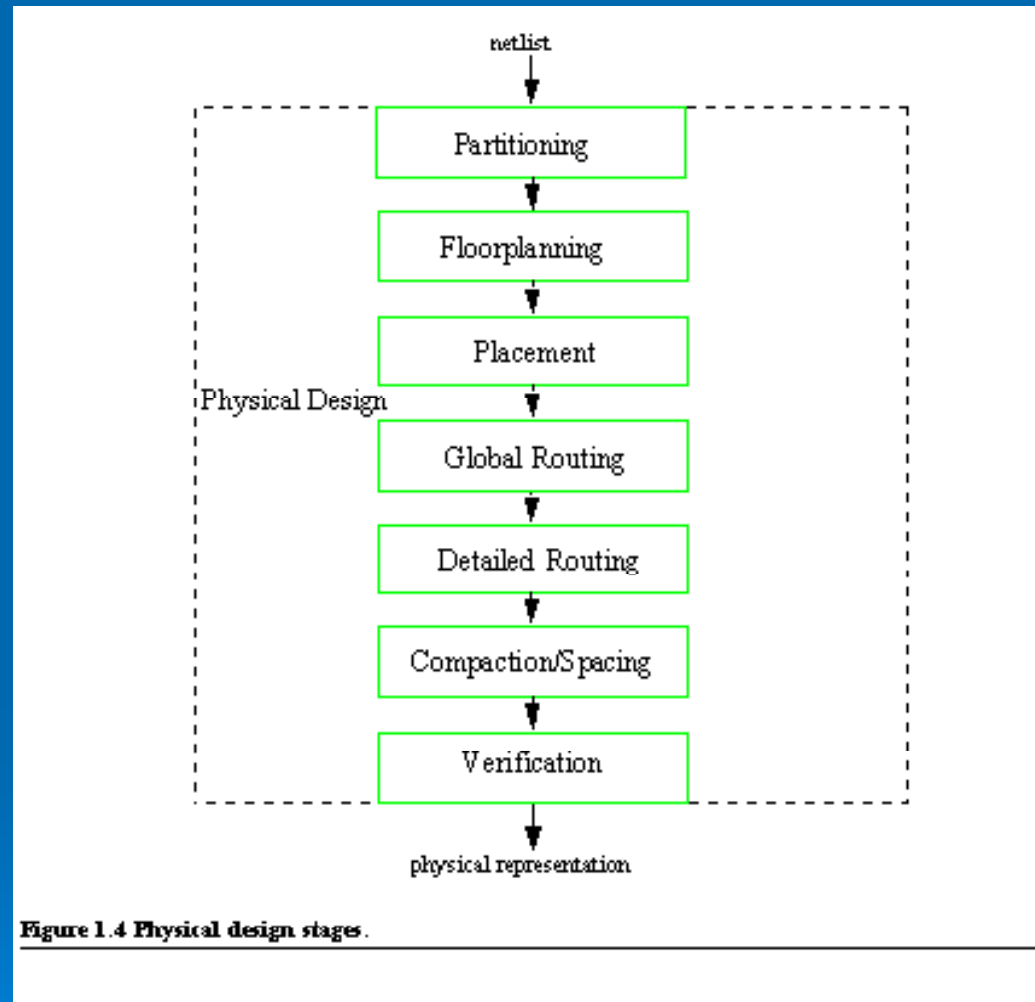
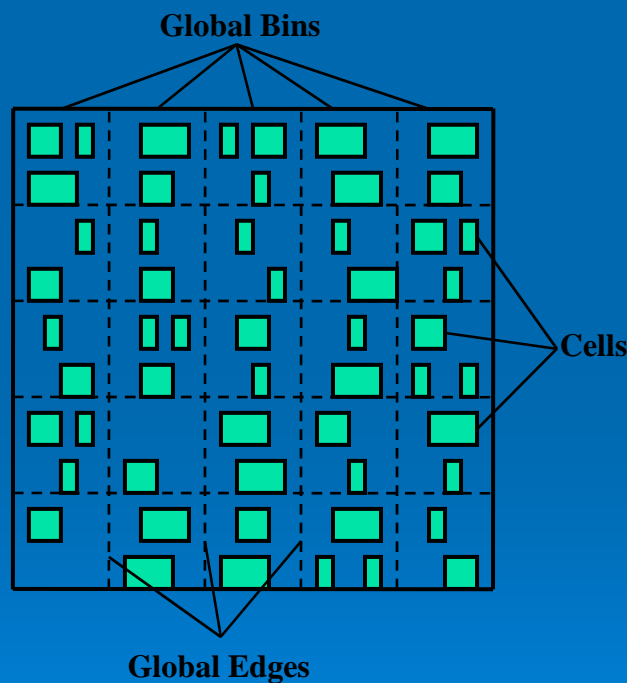


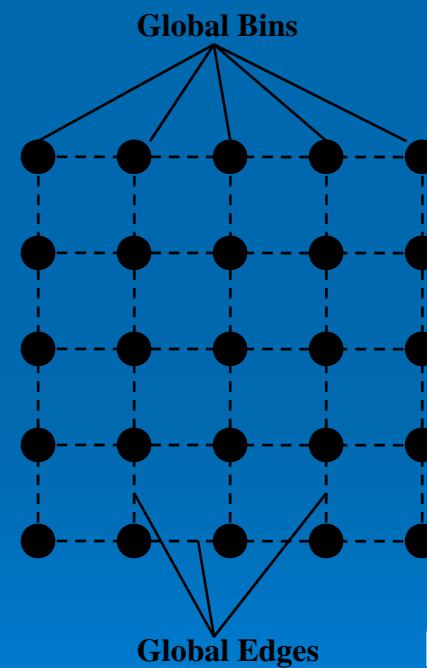
Figure 1.4 Physical design stages.

# Global routing definition

- Assign net segments to physical regions.
- Model regions as a graph.
- Nets are embedded into graph
- Minimize the total overflow on all global edges



(a)



(b)

From Pan M, Chu C.

# Cost function

$$\sum_{n \in N} W(n)$$

*subject*  $P_c = 0, 1 \leq b \leq B$

# Global routing difficulty

- Easiest of three primary subtasks
- Fast
- Detail router can ignore global routing to complete design
- Global routing is not necessary for sufficiently small designs
- Negotiated detail router (Pathfinder) + A\* search performs poor man's global routing

# Global and Detail Router Programs

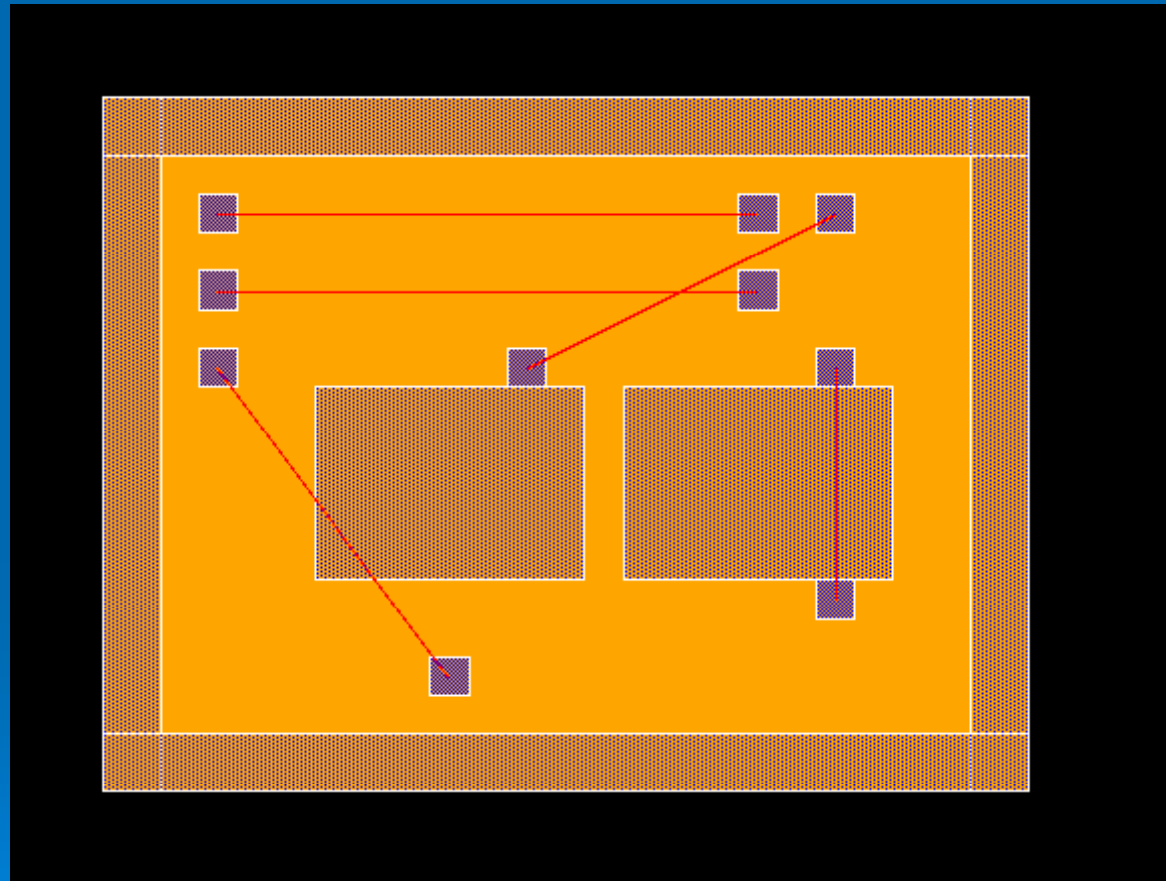
- Combination of base, meta, and hierarchical algorithms.
- Base : how to route a single net or net segment
  - Maze, pattern, line search, gridded, etc.
- Meta : how to route a set of nets
  - Ripup and Reroute, Pathfinder, Graph-Based, Lagrange Relaxation, Linear Programming, Network Flows, etc.
- Hierarchical : how to route a set of regions
  - Multilevel, channel-based methods, wire ordering (Groeneveld)

# Detail Routing Meta Algorithms

- Now appear in global routers
- Ripup and Reroute
- Graph-based Conflict Removal
- Congestion Negotiation (Pathfinder)

# The problem

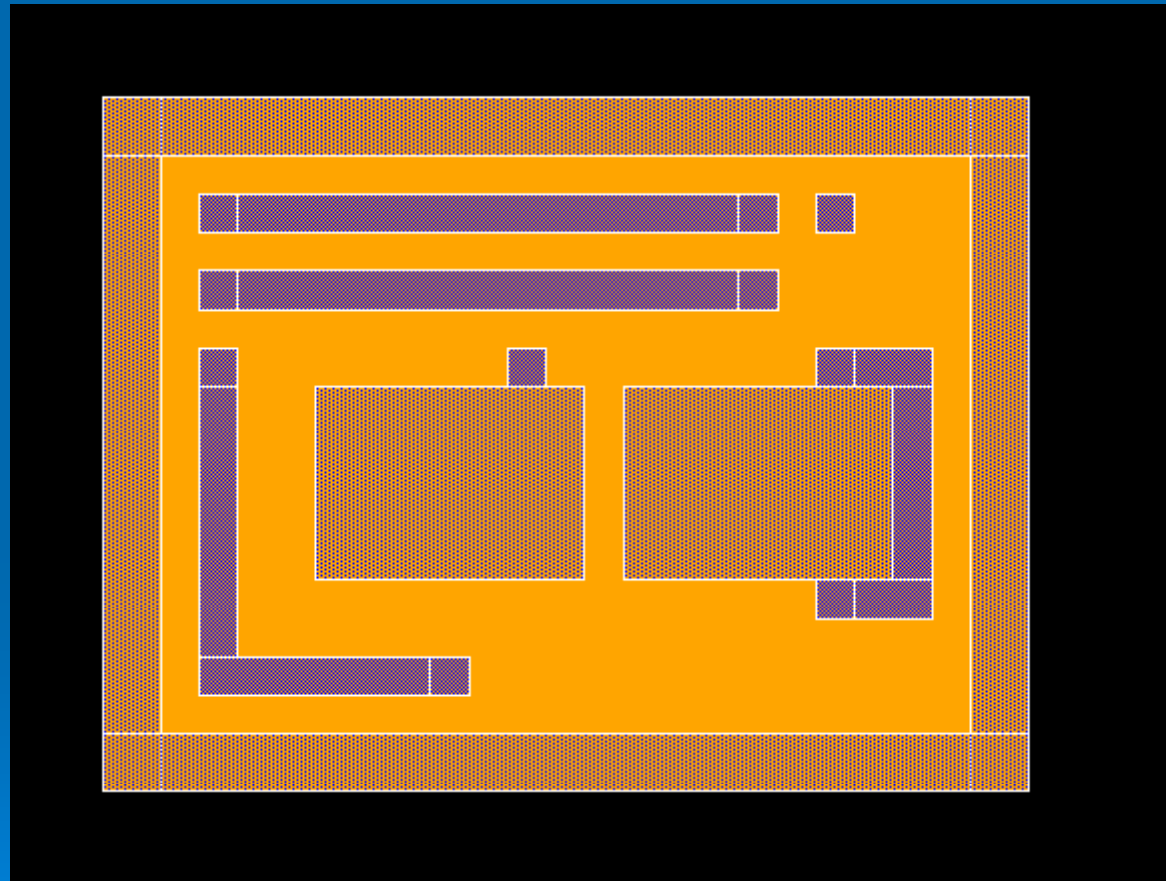
Shaded area keepouts; pins blue; fly-lines denote net connections  
Single routing layer



From Tatsuo Ohtsuki, "Layout Design and Verification" 1986

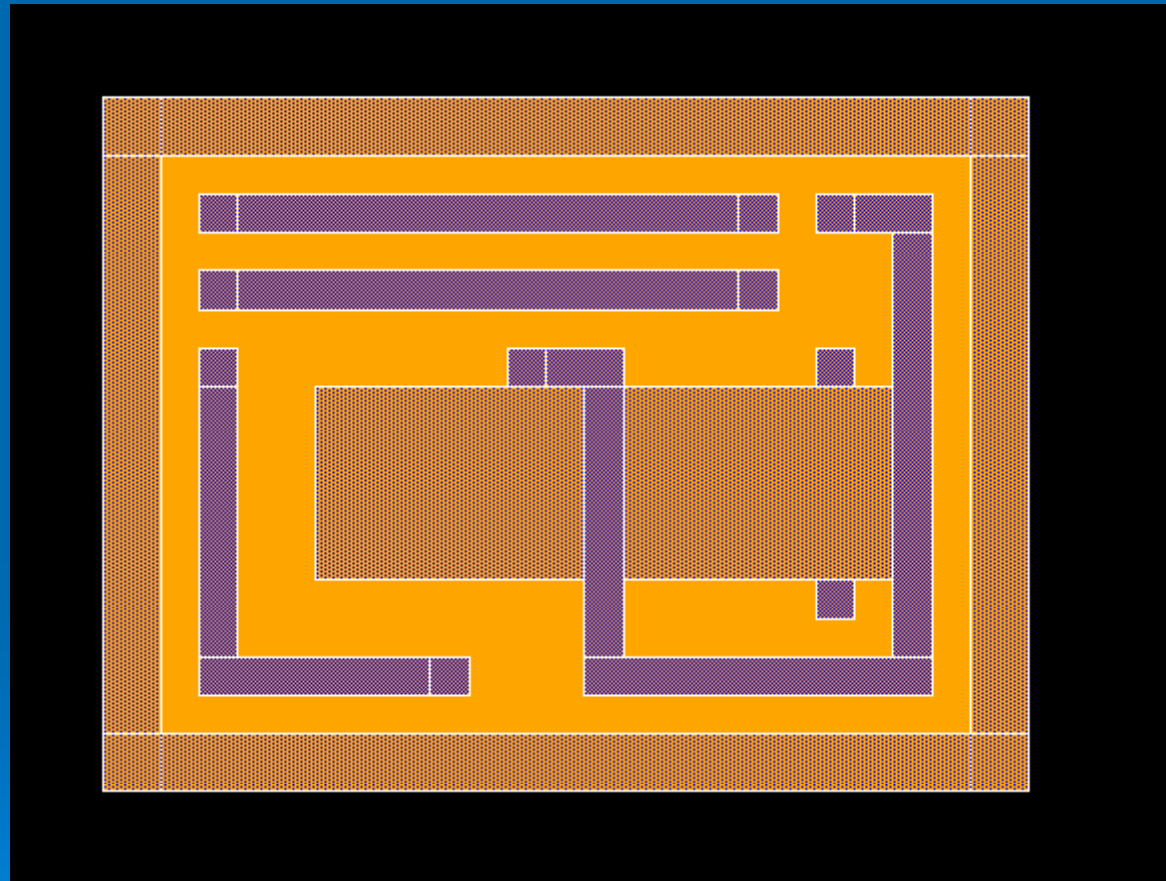
# Net 1 is blocked

Route net 2 first



# Net 2 is blocked

Ordering nets won't solve problem

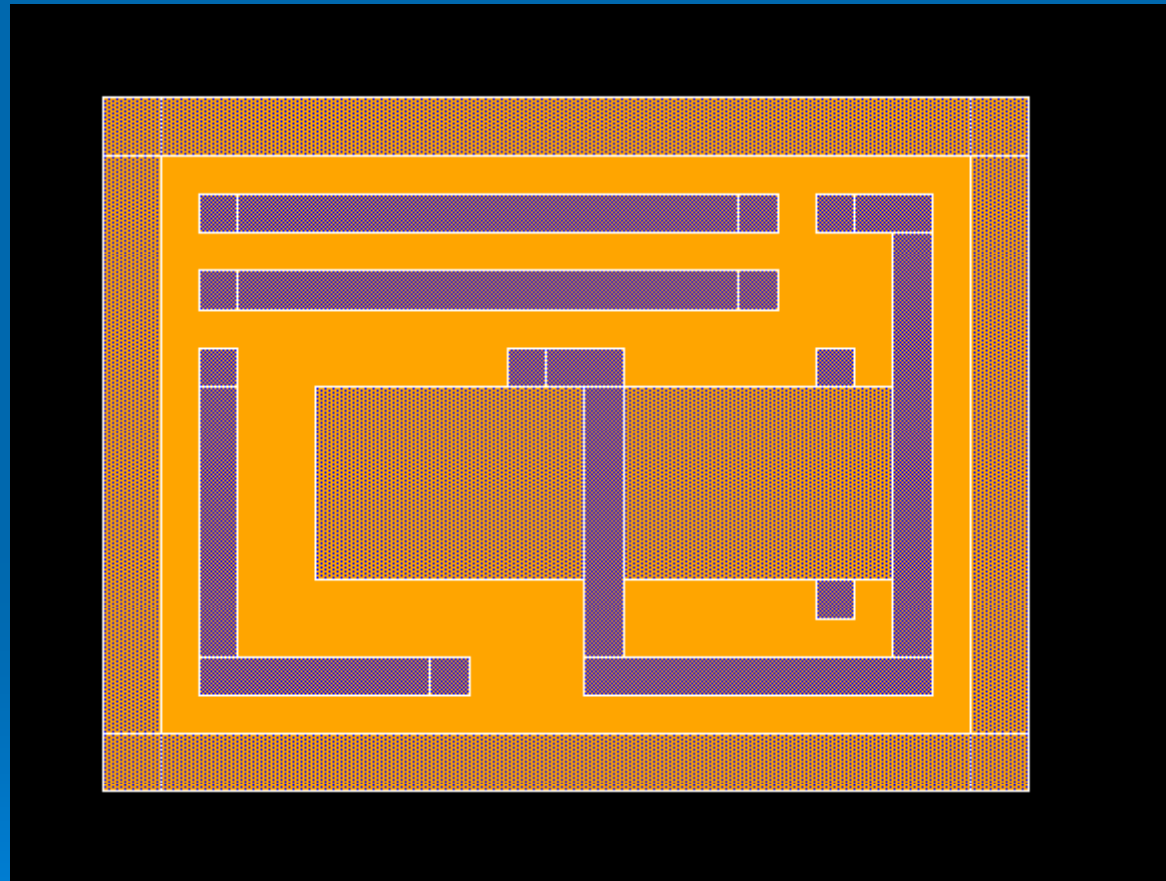


# Ripup and Reroute

- Ripup and reroute net segments - not the entire net!
- Ripup of entire net does not work.

# Ripup/Reroute: Net 2 is blocked

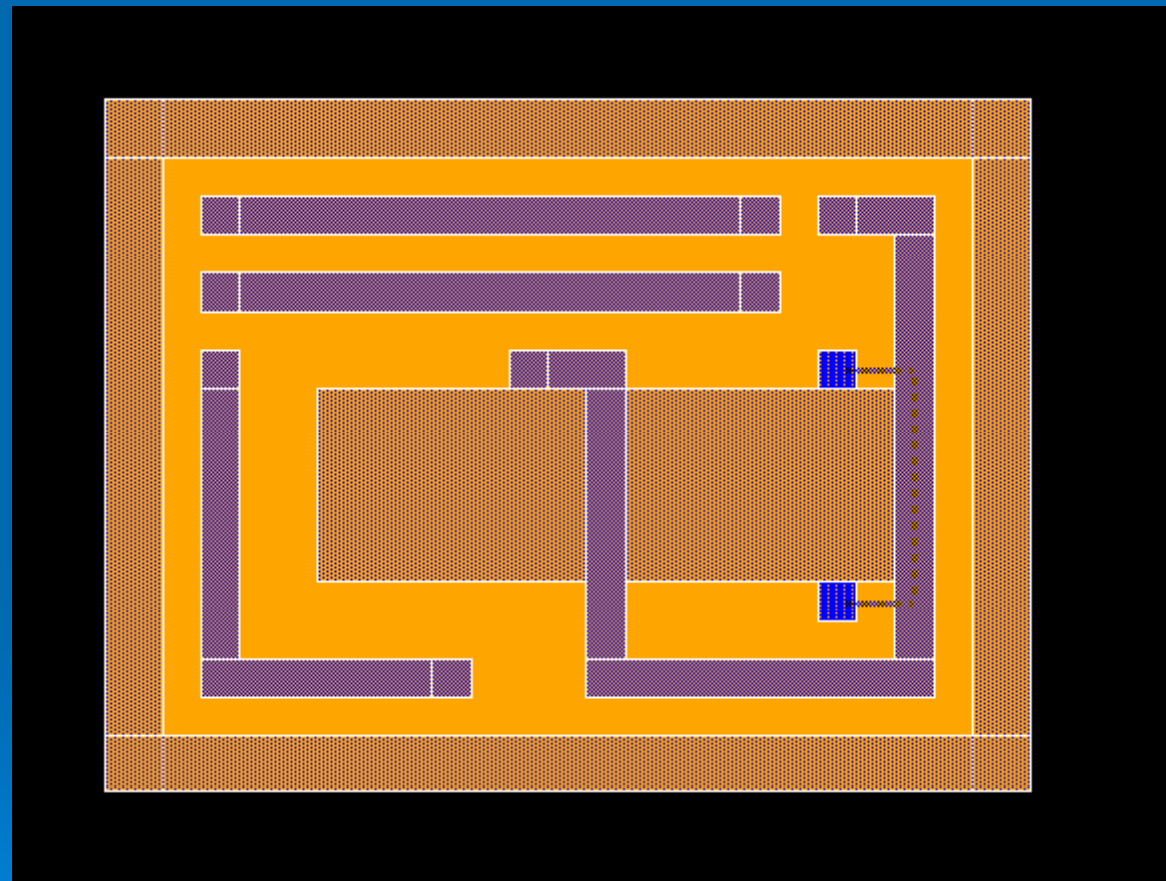
Initial state



Recursion depth 0

# Ripup/Reroute: Route net 2

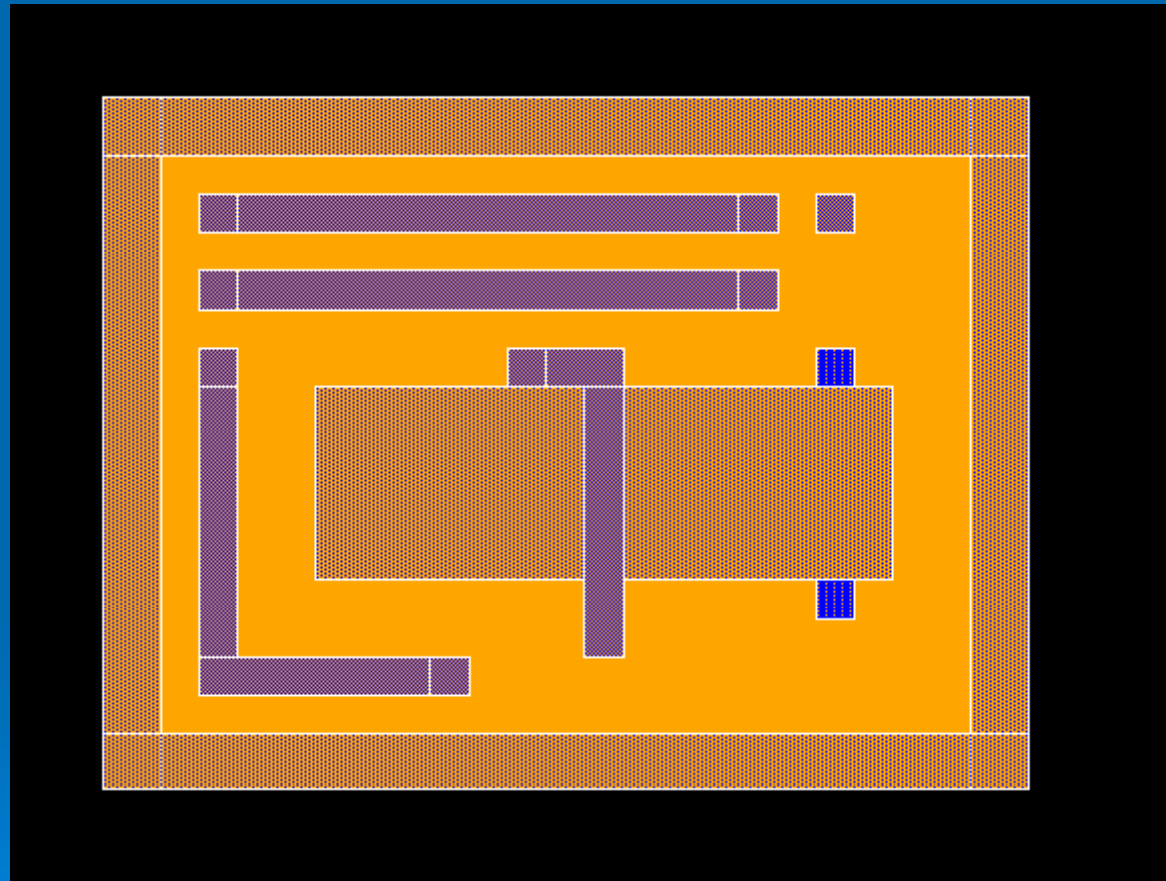
Route borrowing net 1's segment



Recursion depth 1

# Ripup/Reroute: Ripup net 1 segment

Remove net 1 segments

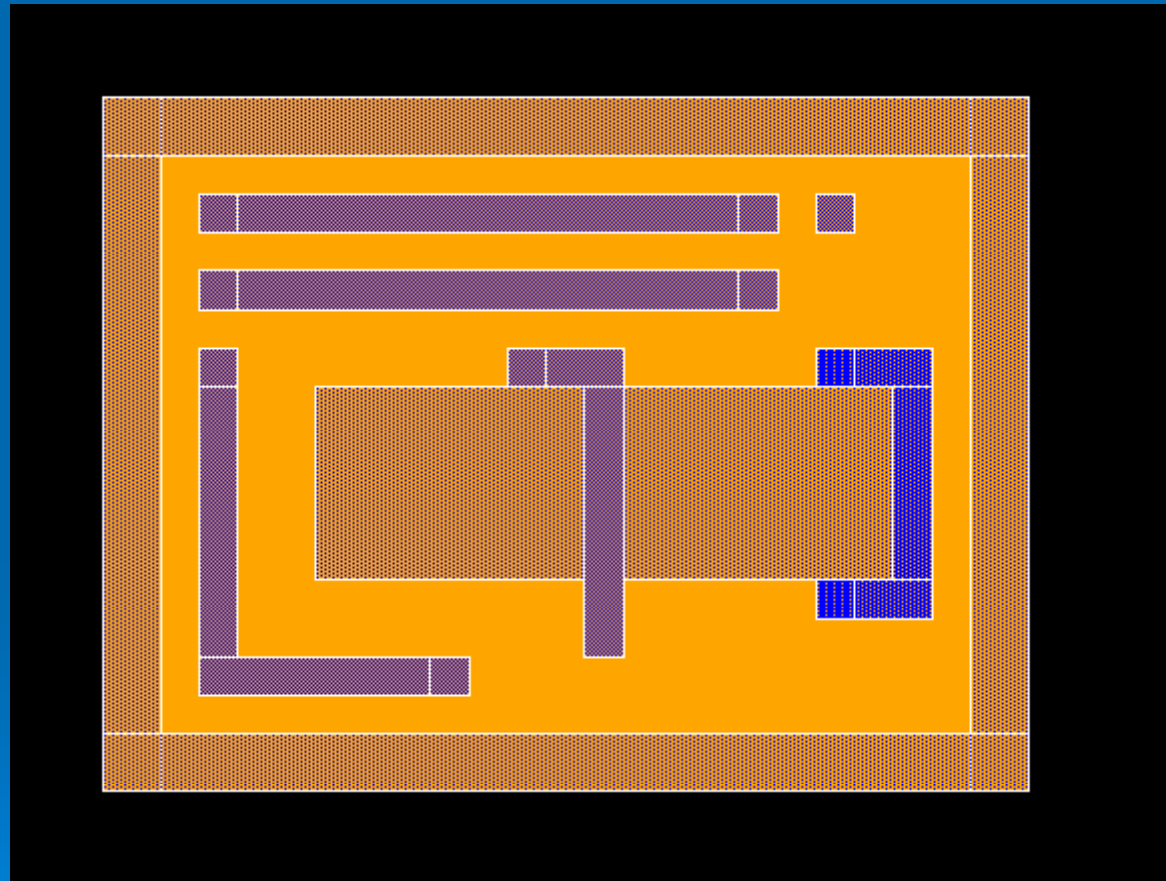


Recursion depth 0

# Ripup/Reroute: Route net 2

But we fail to route 1

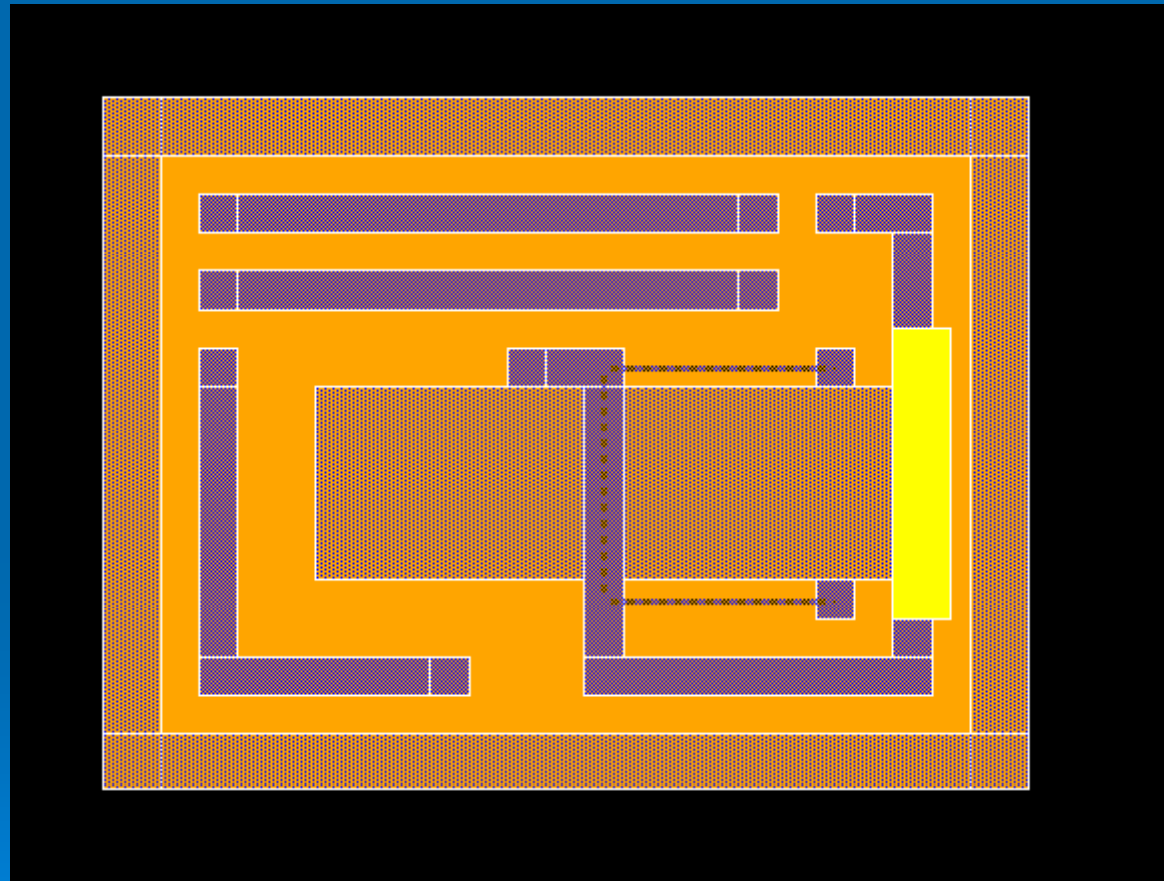
Recursion depth 0





# Ripup/Reroute: Route net 2

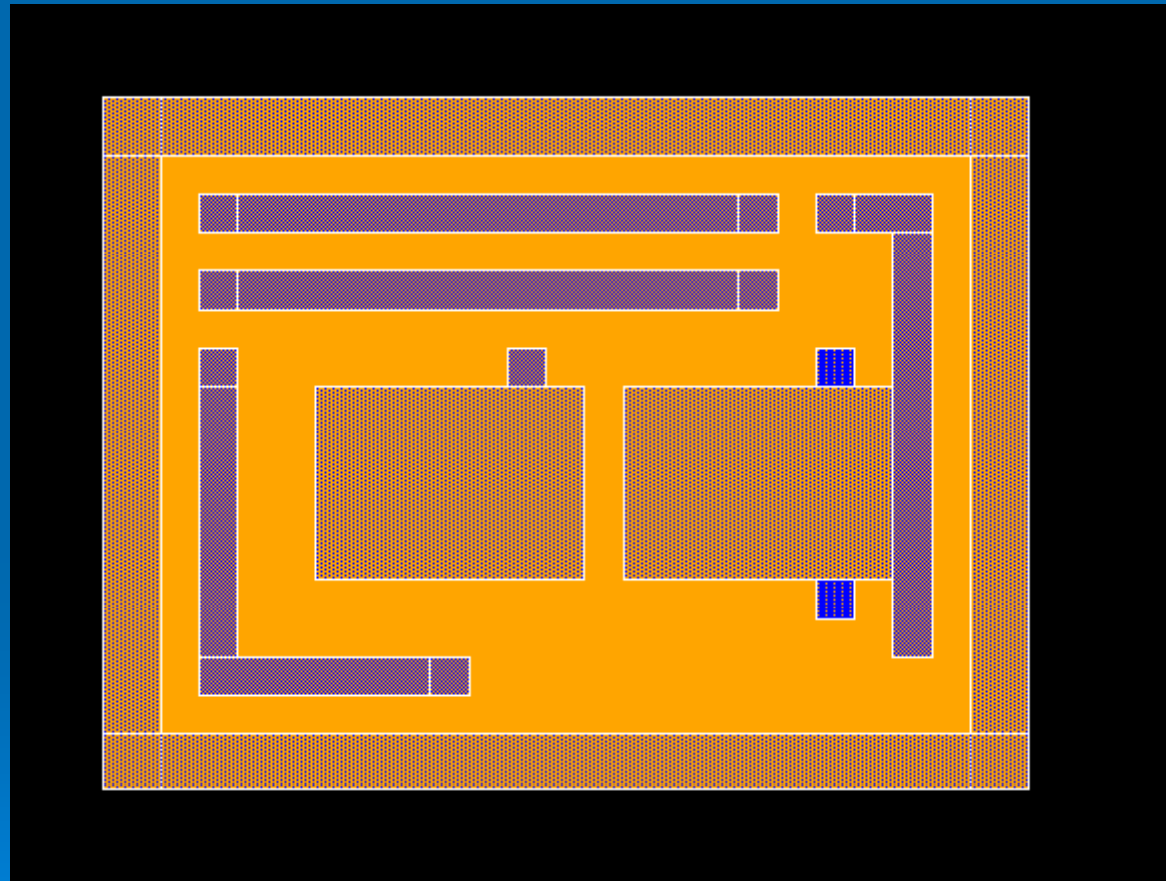
Block previous failure and route borrowing from net 1



Recursion depth 1

# Ripup/Reroute: Ripup net1 segment

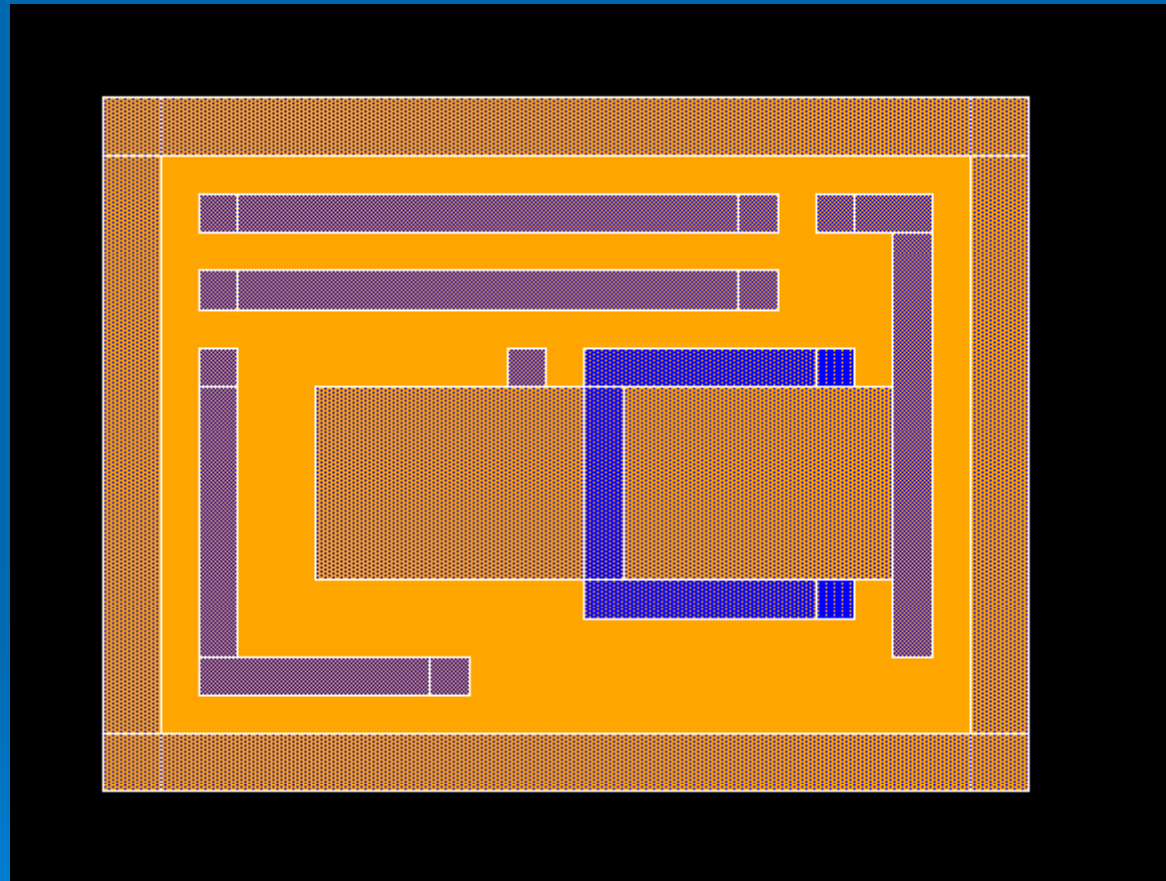
Prepare to route net 2



Recursion depth 0

# Ripup/Reroute: Ripup of segment of net 2 to route net 1

Routing succeeds for net 2

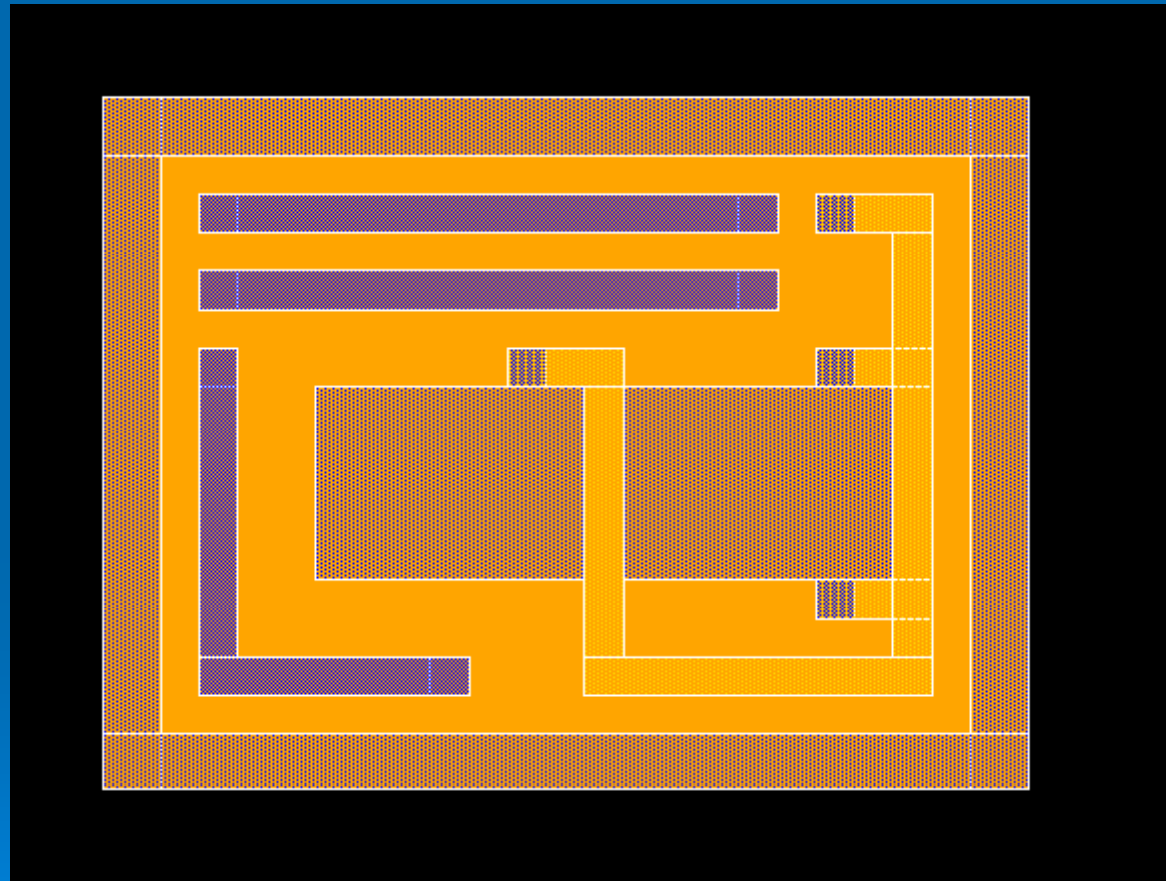


Recursion depth 0



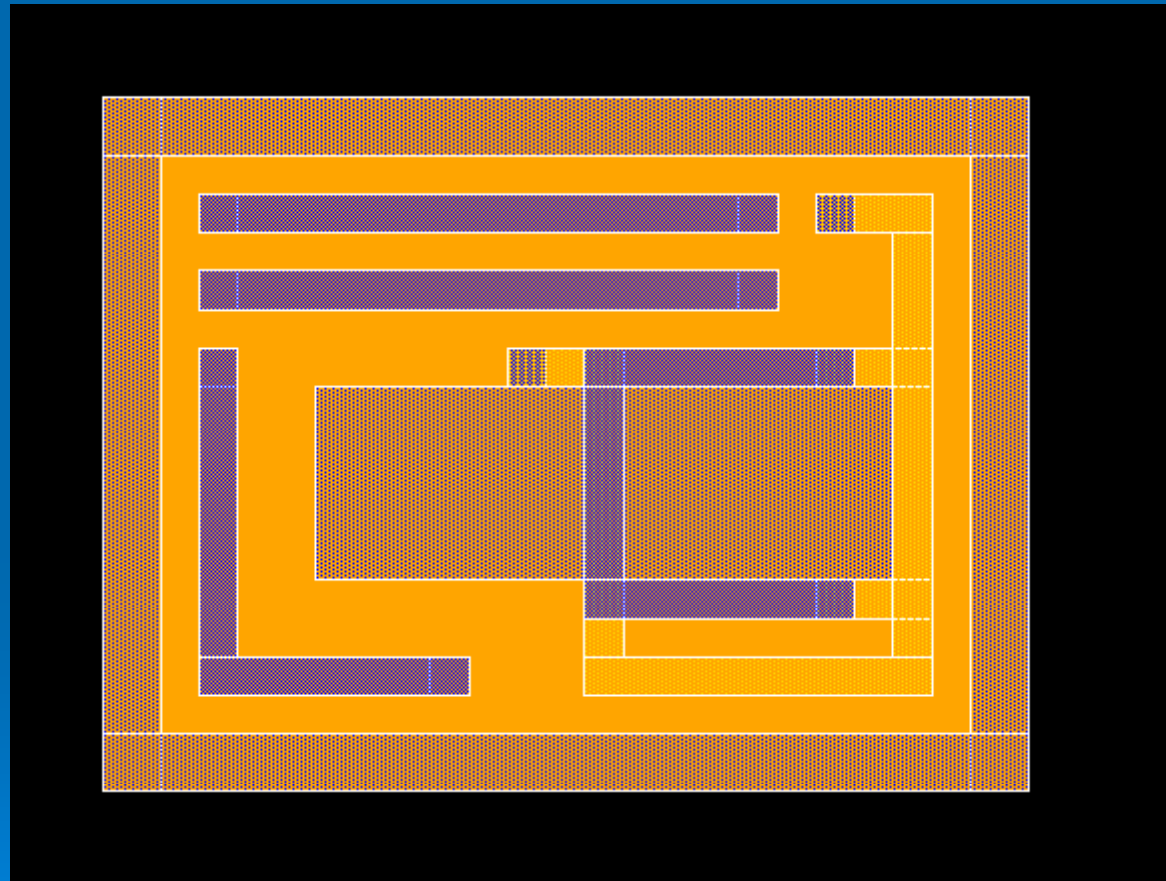
# Graph-based : Initial conflicts

Conflicts shown in orange

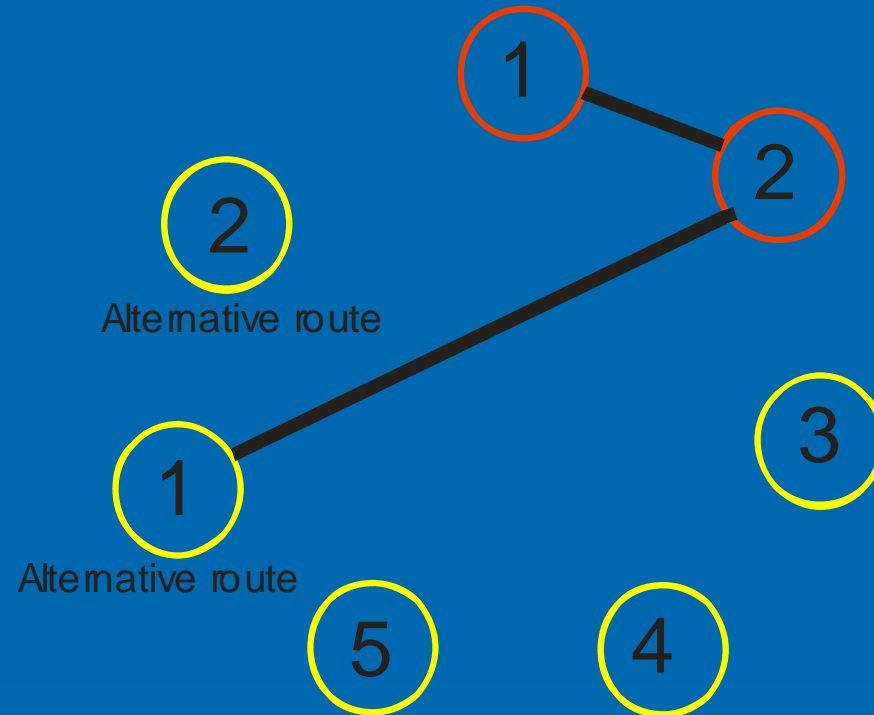




# Graph-based: Alternative candidate for net 2



# Graph-Based: Conflict Graph

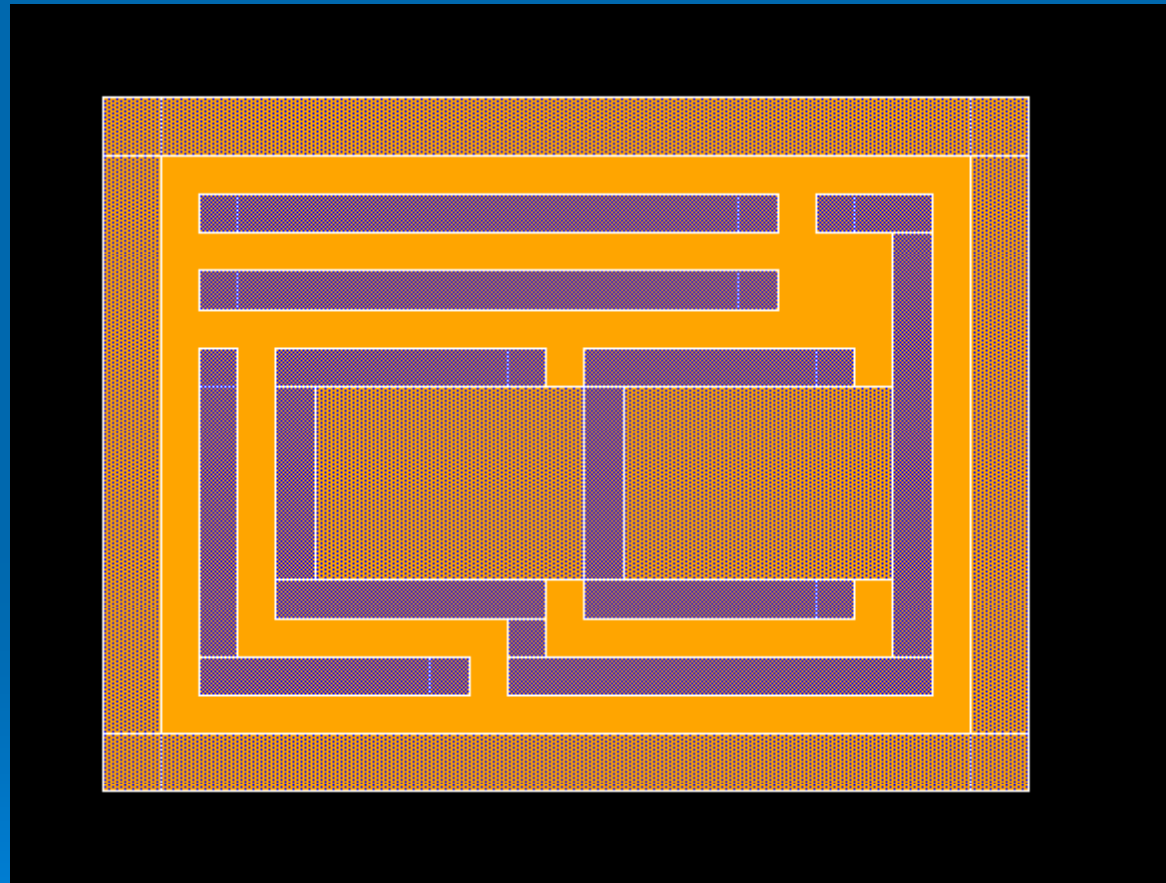


Each candidate route becomes a node in conflict graph

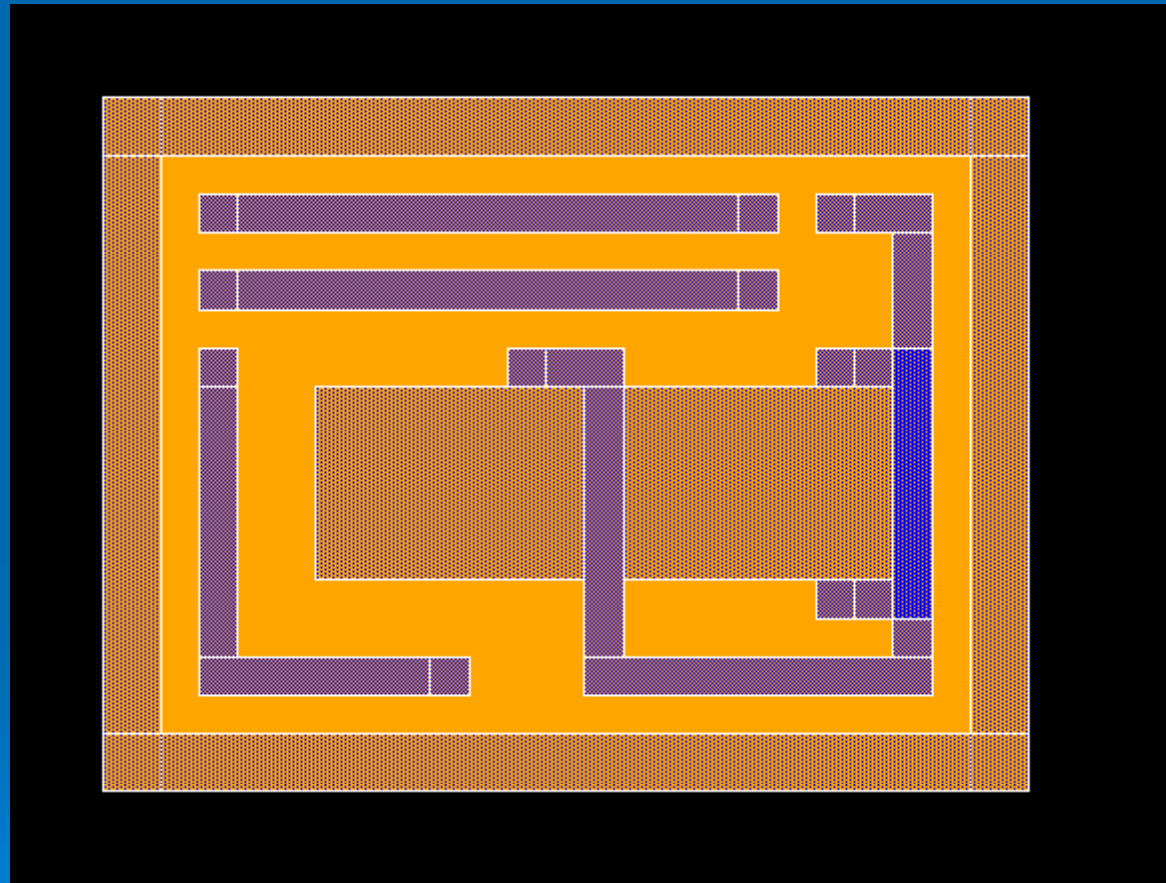
Conflicting route candidates form edges

Find set of nodes with no conflicting edges

# Graph-based: Conflicts resolved



# Pathfinder: Initial Conflicts







# Detail Routing Meta Algorithms

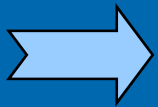
	Relative Speed of Convergence		Weakness
	Easy Instance	Difficult Instance	
Ripup and Reroute	Fast	Slow	Recursion tree traversal order
Graph-based	Medium	Medium	Missing candidates
Congestion Negotiation (Pathfinder)	Slow	Faster	Penalty increment

# Negotiated Congestion Router without GR

- Advantages
  - No granularity problem
- Disadvantages
  - Doesn't scale well
  - Impractical for large problems

# Age of global routing

- Today problems involve many millions of gates
  - Time to market and productivity important



The age of global routing

# Global routing applications

- Full chip
- Subchip or block level
- Specialty
  - Analog
    - Symmetry
    - Length constraints

# Design Methodologies

- Fixed die
- Variable die
- Constrained die

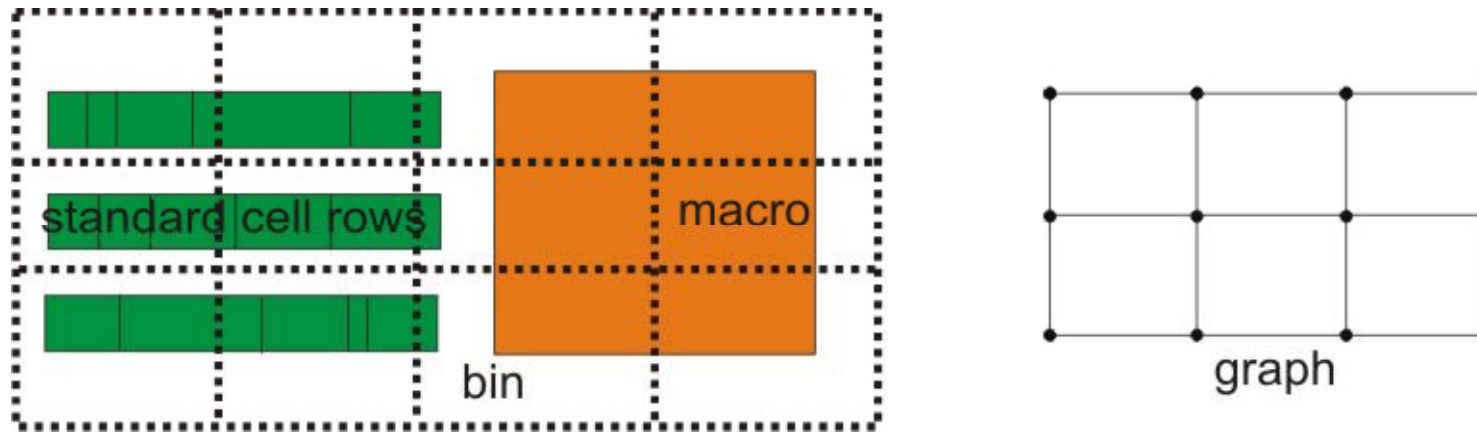
# Fixed Die Methodology

- Dominates industry and academia
- Placement consists of two stages:
  - Coarse
  - Detail
- Global routing
  - Timing / Congestion Driven
- Detail routing
  - Multilevel to increase capacity and fix problems due to abstraction

# Previous Work

- Multicommodity Flow Albrecht C (2001)
- BoxRouter Cho M. and Pan D., (2006)
- Lagrange Relaxation Roy J, and Markov I (2007)
- FastRoute Pan M. and Chu C. (2007)
- Rich literature. Too many to mention

# Bin Based



Each region becomes a node in the graph

Adjacent regions form edges in the graph

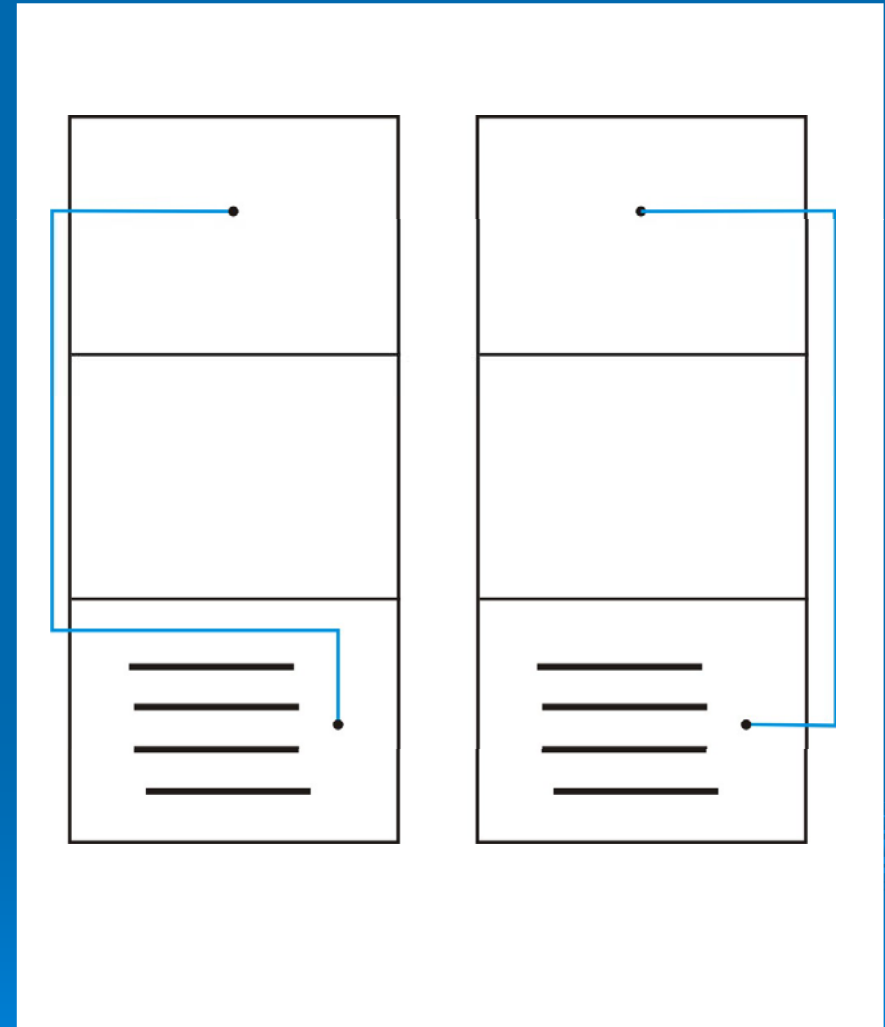
Graph may be 2D or 3D

# Global Routing Issues

- Bin Model Granularity Errors
- Extending the Domain of Global Routing
  - Placement
  - Detail Routing
- Methodology issues
  - Variable Die
  - Constrained Die

# Granularity Errors

- Bin model counts edge crossings
- Bin model fails to capture in-bin density
- Bin size 15-50 tracks typically
- ISPD benchmark contest 30-50 tracks
- Effect prominent in better placements
- Obvious during detail routing
- Wire widths not taken into account



# Why bin boundary approximation?

- Pin locations are known
- Exact is easy enough with interval trees
- Non-uniform wire pitches (3D case)
- Use layer direction orthogonality
  - Layers either horizontal or vertical
- $O(\log N)$  time complexity

# Global router should do placement

- Half-perimeter wire length is replaced with Steiner trees
  - Far more accurate wire length
- Accurate density calculations
  - Detail routing possible if  $d_b \leq s_b + k, \forall b \in B$
- Why not detail placement?

# Congestion: a poor metric

- Cost = wirelength + congestion + timing

$$C = \beta_w W + \beta_t P_t + \beta_c P_c$$

$$W = \sum_{n=1}^N W(n)$$

$$P_t = \sum_{p=1}^{N_p} D_p$$

$$D_p = f(R, C, l, t_g)$$

$$P_c = \sum_{b=1}^B P_c(b)$$

Fails dimension analysis

Reliance on benchmarks for tuning

# Timing Penalty Revisited

- Rewrite timing penalty in terms of net length

$$P_i = \begin{cases} \text{length}(p) - \text{upperBound}(p) & \text{length}(p) > \text{upperBound}(p) \\ \text{lowerBound}(p) - \text{length}(p) & \text{length}(p) < \text{lowerBound}(p) \\ 0 & \text{otherwise} \end{cases}$$

where

$$\text{length}(p) = \sum_{\forall n \in p} W_n$$

Successfully used on many microprocessor designs

# What about Congestion?

- Detour length replaces congestion term
- How far to travel to get to uncongested area
- Timing and congestion rewritten in terms of length
- All terms in cost function are length-based
  - Scales properly
  - Tradeoffs between terms straightforward

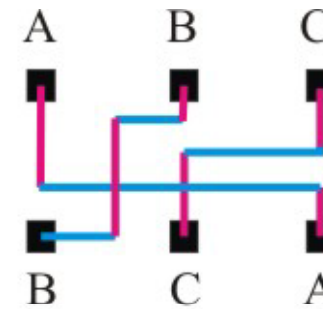
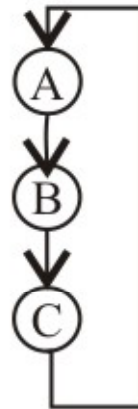
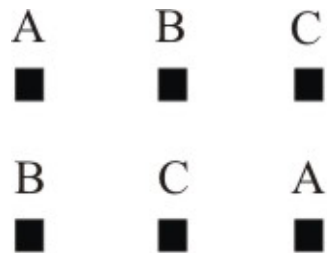
# Calculating Detour Length

- During placement, use congestion map to detour length algorithm. Kangh and Xu (2003)
- But during global routing, we have all the information to directly compute it.
- Possible metrics
  - How far are we outside the bounding box of a net's pins
  - Excess wire length above minimum Steiner tree bound
  - Net segment detour length
- Use traditional congestion metric as a tie breaker.

# Global router should do detail routing

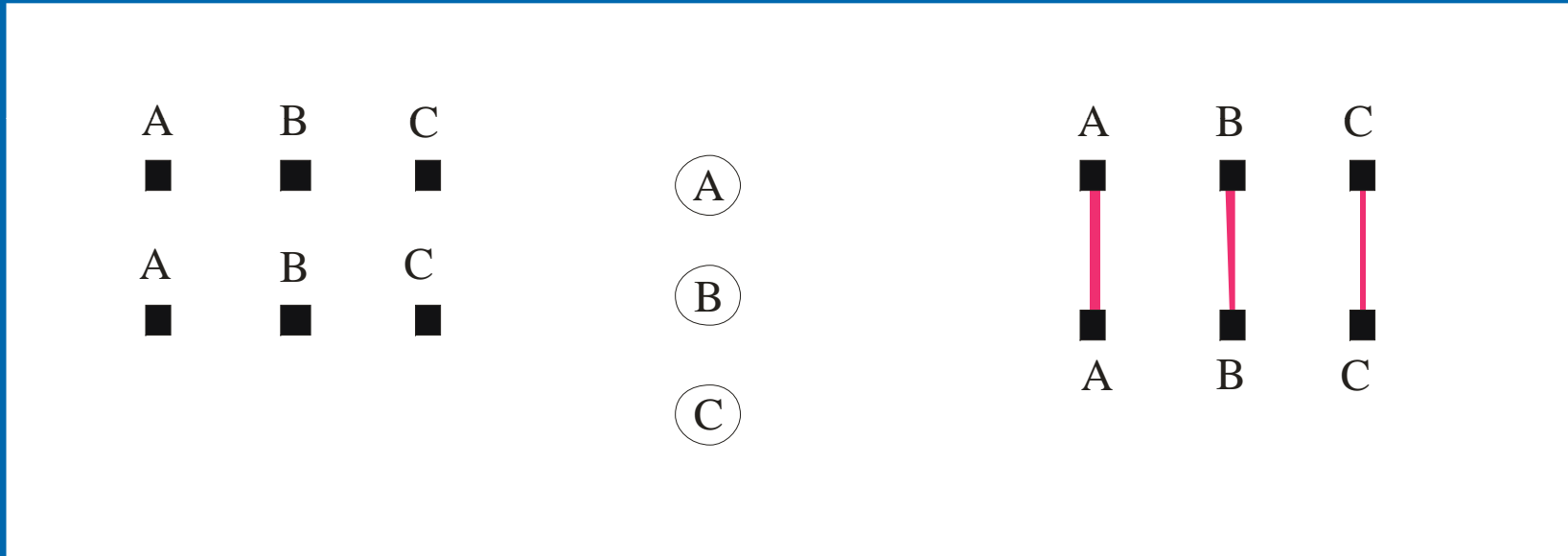
- Alignment of orthogonal resources within the bin or region
- Known as virtual pin resources or feedthru assignment
- Long range planning
- Minimize vertical constraint cycles

# Minimizing Vertical Constraints



Forces new column

# Minimizing Vertical Constraints



Most detail routing fails at the pins

Detail router appreciates help from the global router

# Minimizing Vertical Constraint Cycles

- Ways to break cycle
  - Move virtual pin assignment
  - Swap equivalent pins of a gate (rewrite netlist)
  - Rotate a cell
  - Move a cell

# Track assignment

- 1D routing
- Crosstalk
- Noise
- Early channel-based detail routers (Greedy, Mighty)
- Solve using linear assignment, partitioning, linear programming...

# Simultaneous Routing

- Power, clock, signal routed simultaneously
  - Wide wire requires planning
  - Verified Saxena and Gupta (2003)

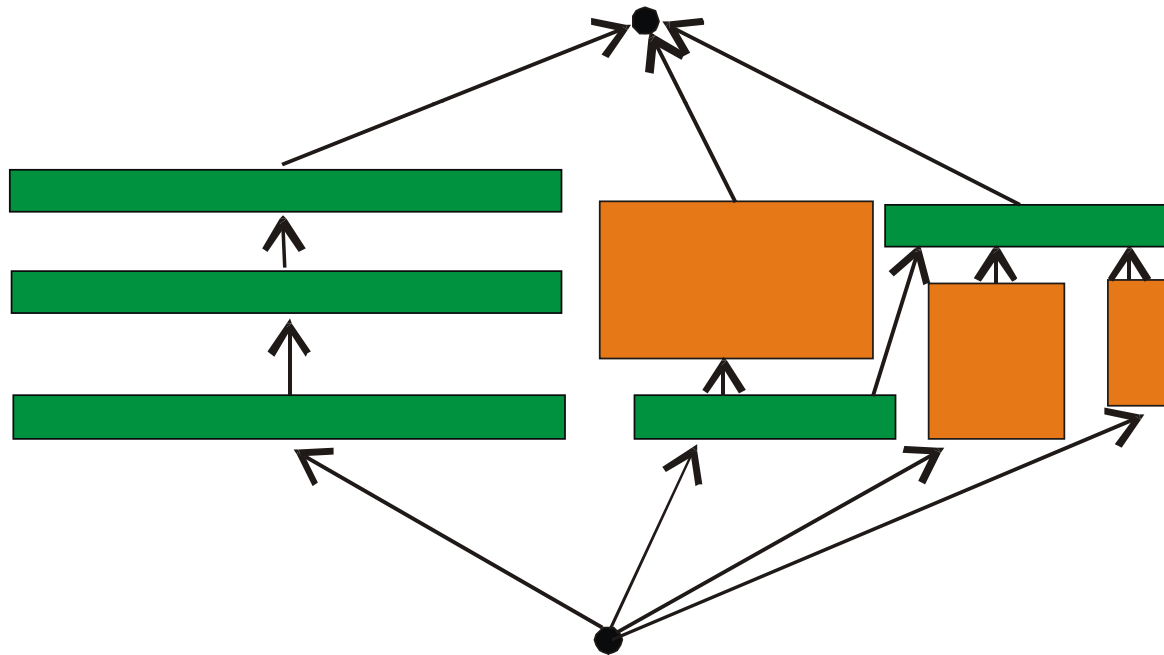
# Synthesis during global routing

- Buffer synthesis
- Gate resizing
- Scan chain synthesis
- Clock tree synthesis

# Other design methodologies

- Variable die
- Constrained die

# Variable Die



White space is manipulated to insure no overflow  
Placement is not fixed.  
Relative placement may change as well

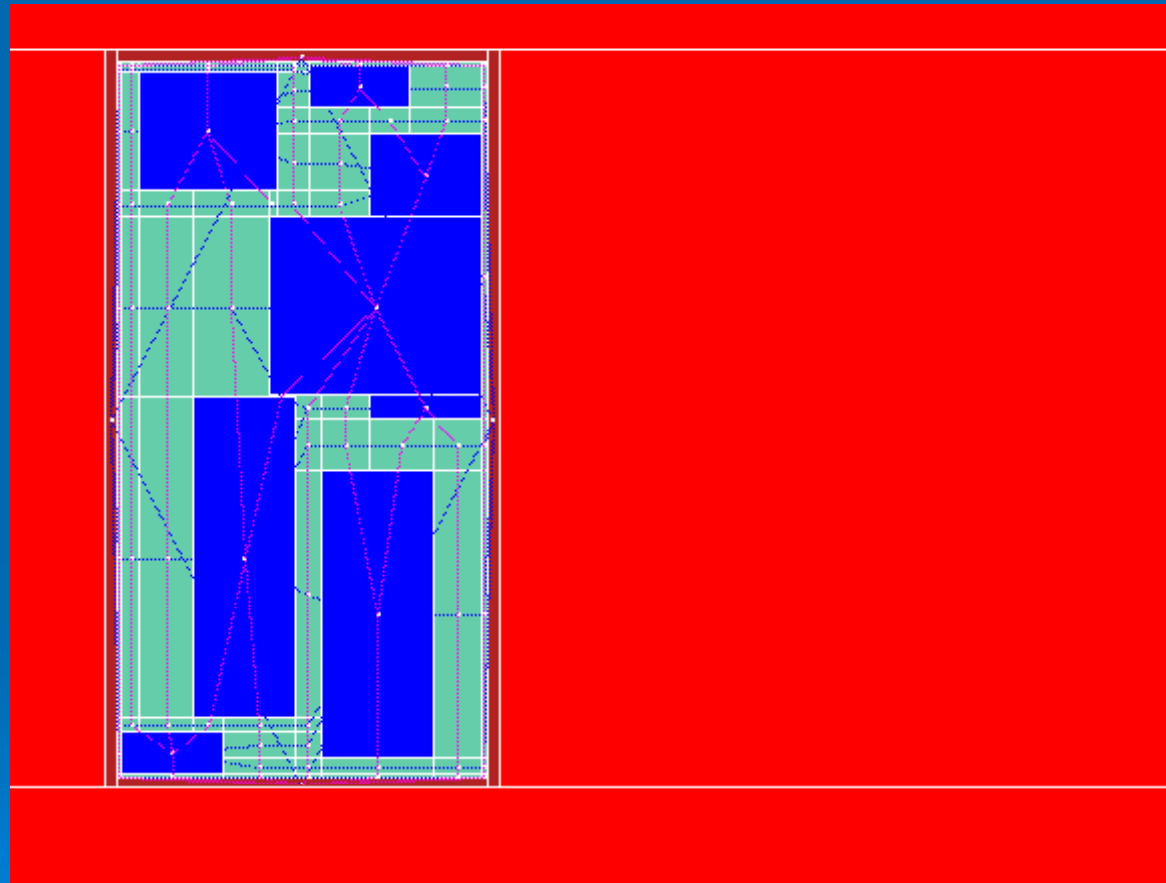
# Variable Die

- Applicable to block level
- Early stages of chip level
- Minimize area
- Trade off # of routing layers versus cost
- Guarantees completion
- Minimizes congestion (controversial)
- Original P&R algorithm using channels

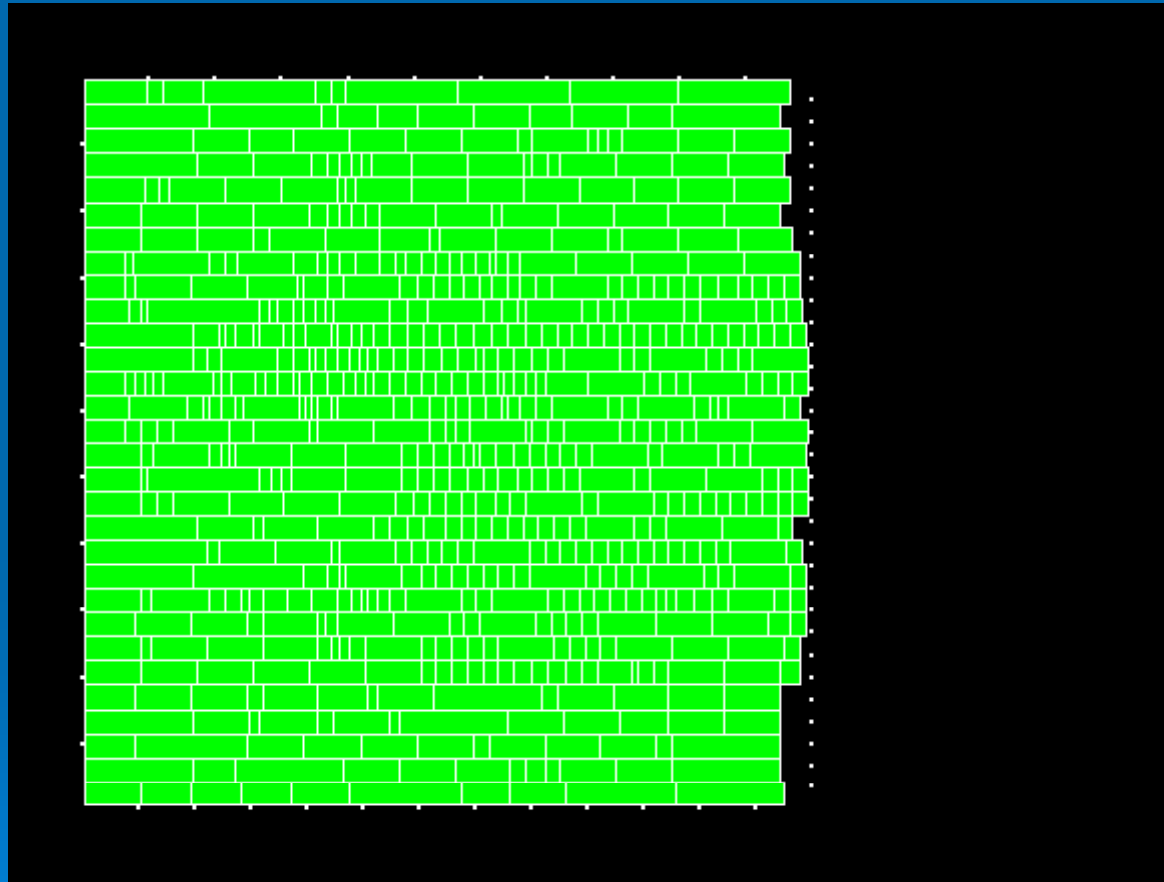
# Variable Die Algorithm

- Virtual pin assignment determines orthogonal (vertical) resources
- Partition into virtual regions to determine horizontal resources
- If all pins at center of standard cell
  - Perfect decomposition
  - Global router decouples virtual regions
  - Perfect parallelism – all core regions at once.
  - Blocks/macros complicate things

# Macro Regions



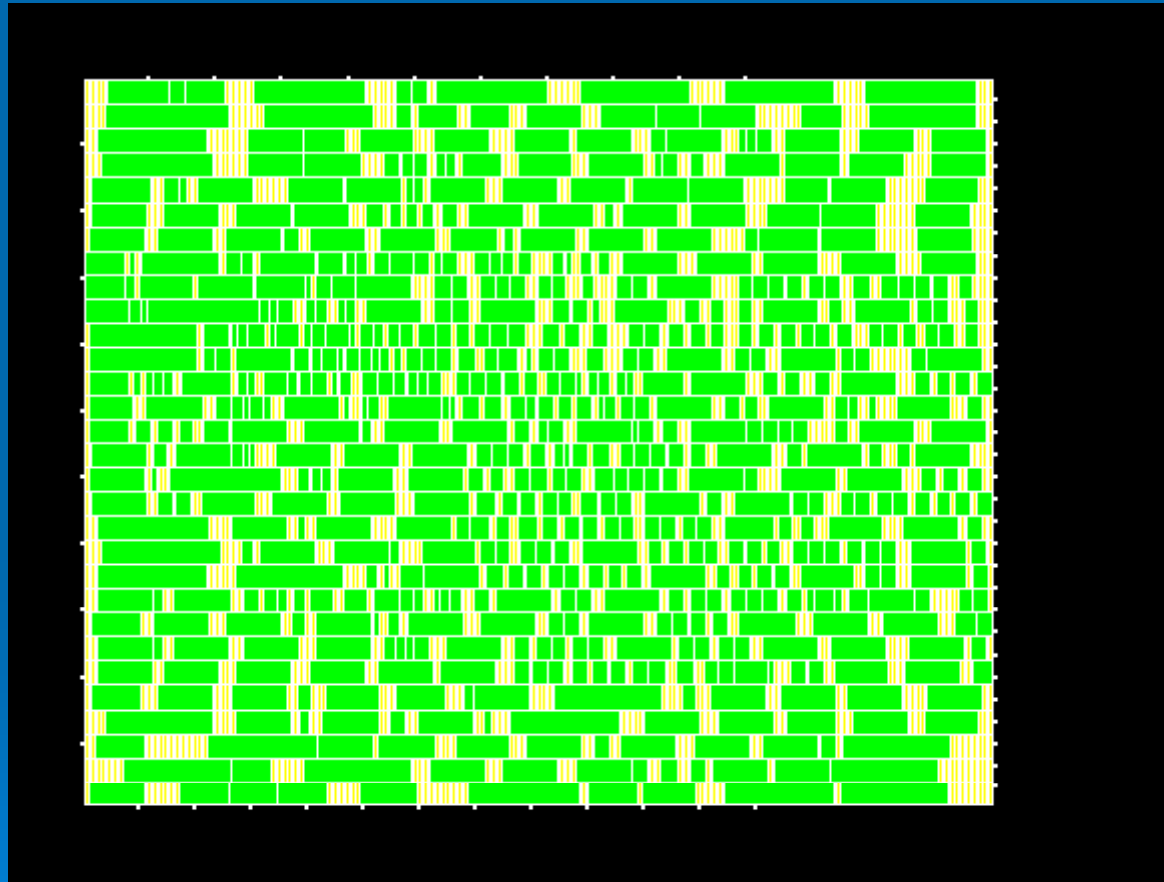
# Input to Global Router



# Region Definition



# Output of Global Router



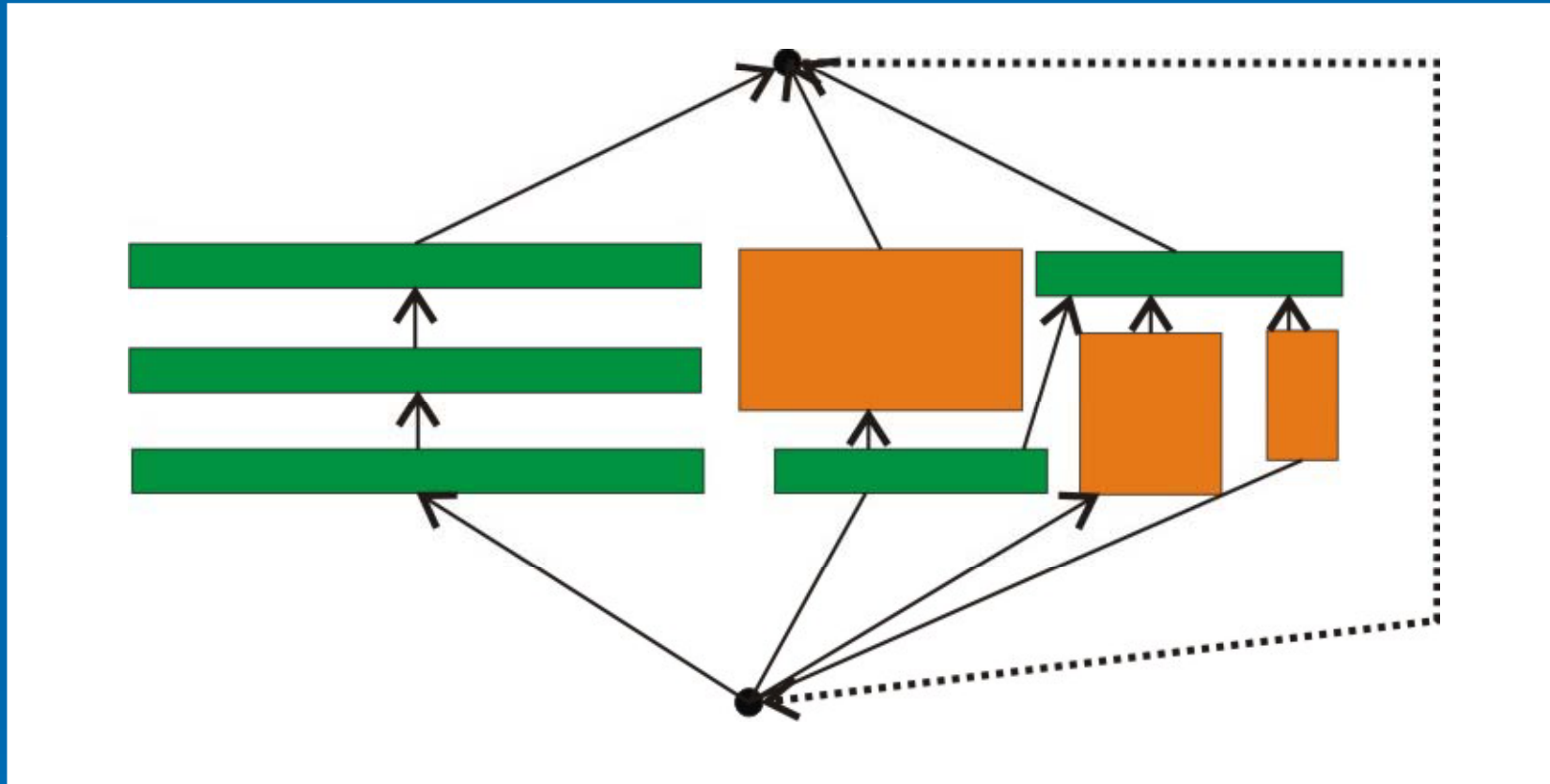
# Chip Design

- Chip design process uses iterative refinement not one-shot
- Design space exploration at beginning
  - What is possible?
- As knowledge is acquired, more and more aspects become fixed
  - Die size become fixed
  - I/Os become fixed

# Constrained Die

- Designer cares only about interface of block or chip remain constant during iterations
  - Footprint important
  - Physical location of internals irrelevant
- Propose Constrained Die Methodology

# Constrained Die



Constrained die compaction graph in y direction.  
Dotted line shows constrained die fixed edge.

# Compacting global router

- Global router compacts virtual regions such that block/chip constraints maintained.
- Maximize degrees of freedom
- Delay whitespace allocation as much as possible
- Guarantee solution
- Trade offs much easier
- ECO compatible

# Variable Die Algorithm Problems

- Detail routing slow if not in parallel
  - Compaction is done by maze router because topology changes
- Global router errors
  - Estimate off by 1 track
  - Assignments to wrong region
- Solved by Multilevel but this negates parallelism
- Complicated

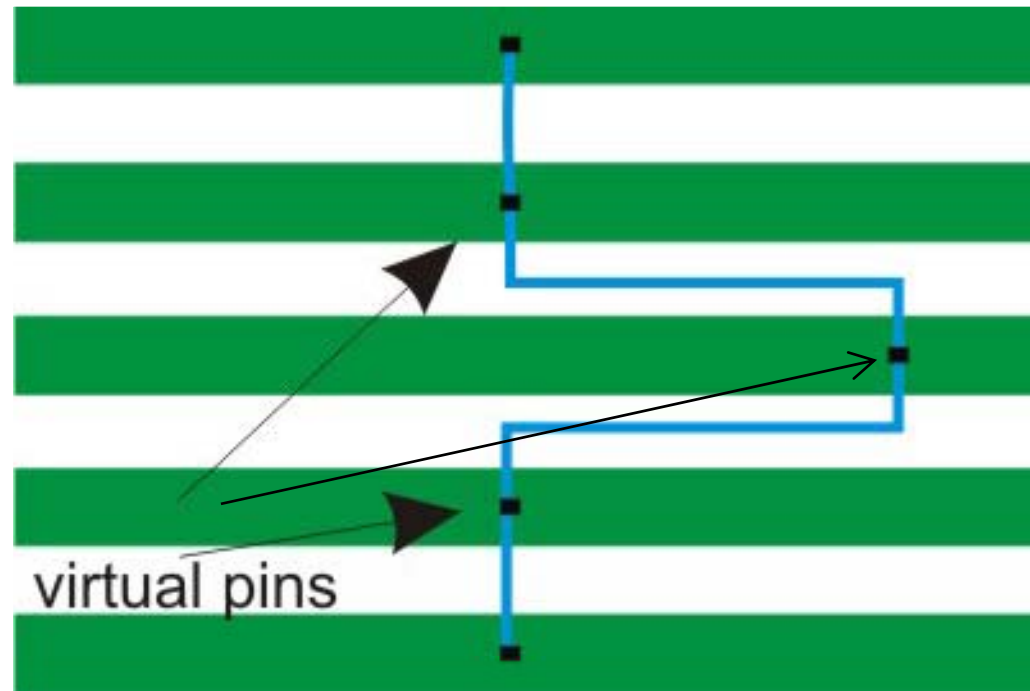
# Itools Global Routing Algorithm

```
globe(void) {  
    setup_routing_regions() ;  
    create_prerouting() ;  
    instantiate_power_ground_networks() ;  
    process_scan_chains() ;  
    process_clocktrees() ;  
    steiner() ;  
    initialize_gr_cost() ;  
    switchable_segment_opt() ;  
    area_minimization() ;  
    freeway_assign() ;  
    feedthru_assign() ;  
    for( c = 1 ;c <= reassign_limitG ; c++){  
        cell_swap_opt() ;  
        detour_minimization() ;  
    } /* end for( loop_cnt = 1... */  
    switchable_segment_opt() ;  
    area_minimization() ;  
    congestion_minimization() ;  
    vertical_constraint_min() ;  
}
```

# Area Minimization

```
void area_minimization(GRAPHPTR region_graph,const int MLIMIT)
{
  for( count=1; count <= area_passesG; count++ ){
    // Compact & find longest path but don't move cells.
    compact_fixed_area() ;
    // Determine set of regions where area opt is warranted.
    xpaths = find_M_longestPath(region_graph,ICCOMPACT_X,MLIMIT) ;
    xcritical = assign_critical_regions( xpaths, ICCOMPACT_X ) ;
    ypaths = find_M_longestPath(region_graph,ICCOMPACT_Y,MLIMIT) ;
    ycritical = assign_critical_regions( ypaths, ICCOMPACT_Y ) ;
    // Reroute paths on the critical path.
    for( p = 1 ; p <= xcritical.num ; p++ ){
      path_p = xcritical.paths[p] ;
      reroute_critical_nets(path_p,ICCOMPACT_X) ;
    }
    for( q = 1 ; q <= ycritical.num; q++ ){
      path_p = ycritical.paths[q] ;
      reroute_critical_nets(path_p,ICCOMPACT_Y) ;
    }
  } /* end for( count=1; count <= area_passeS... */
  // Move the cells and rebuild the tile database and graph.
  compact_variable_area() ;
} /* end area_minimization() */
```

# Detour Minimization



# Detour Minimization

```
void detour_minimization(void)
{
    obstacles = build_feed_obstacles() ;
    for( row = 1 ; row <= numRowsG ; row++ ){
        comb_a_row(row,obstacles) ;
    }
    free_feed_obstacles() ;
} /* end detour_minimization() */
void comb_a_row(int row,TILEPLANEPTR obstacles)
{
    // A detour is a C or inverted C.
    detours = find_set_of_detours(row) ;
    // Remove net segments containing the detour
    // making the virtual pin now unused.
    remove_detour_feeds(detours) ;
    // Determine the set of unused virtual pin locations.
    feeds = unused_feeds(row,obstacles) ;
    // Use linear assignment to assign detour
    // segment to virtual pins.
    new_cost = reassign_feeds(row,detours,feeds) ;
} /* end comb_a_row() */
```

# Summary

- Global routers should do detail placement
- Global routers should aid detail routers
  - Minimize cycles in constraint graphs
  - Long range alignment vertical constraints
- Simultaneous routing of signals, power and clocks
- Variable die and constrained die methodologies to remove congestion
- Integral component to floorplanning