Modern Physical Design:
Algorithm
Technology
Methodology
(Part II)

Stan Chow Ammocore

Andrew B. Kahng UCSD

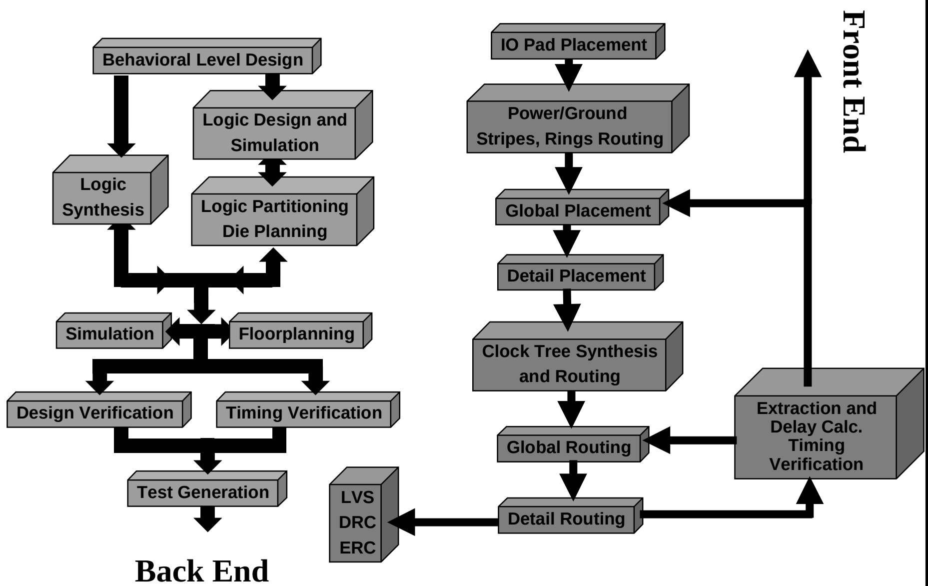
Majid Sarrafzadeh UCLA

PART II: Fundamental Physical Design Formulation and Algorithms

- **Placement**
  - Motivation
  - Formulation
  - Algorithms
  - Complexity management
  - Challenges
- **Routing**
  - Motivation
  - Formulation
  - Algorithms
  - Complexity management
  - Challenges

Front End Flow

Back End Flow



3 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Prediction and Cost Functions

4 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Prediction

- What is prediction ?
  - every system has some critical cost functions: Area, wirelength, congestion, timing etc.
  - Prediction aims at estimating values of these cost functions without having to go through the time-consuming process of full construction.
- Allows quick space exploration, localizes the search
- For example:
  - statistical wire-load models
  - Wirelength in placement

Paradigms of Prediction

- Two fundamental paradigms
  - statistical prediction
    - #of two-terminal nets in all designs
    - #of two-terminal nets with length greater than 10 in all designs
  - constructive prediction
    - #of two-terminal nets with length greater than 10 in this design
  - … and everything in between, e.g.,
    - #of critical two-terminal nets in a design based on statistical data and a quick inspection of the design in hand.
- "Absolute truth" or "I need it to make progress"
- SLIP (System Level Interconnect Prediction) community.

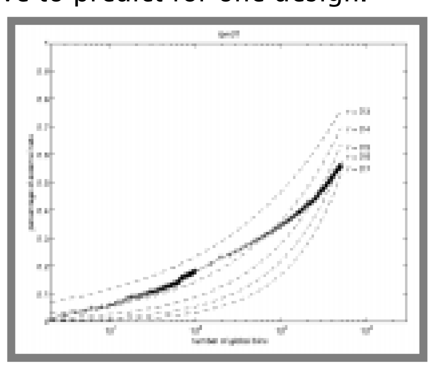
Statistical Prediction

- W. E. Donath , A. B. Kahng, J. Mehndl , et al.
- Developing theoretical/statistical/observational models for interconnect estimation.
- Basic types
  - Estimation of Global Parameters. Assumes homogenous designs.
    - Rent's rule
    - Average multiplicity of a netlist is 2.2 - 2.6
  - Design Specific. Assumes localized homogeneity
    - localized Rent's rule
    - A specific Verilog block has average multiplicity 3.2

7 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Statistical Prediction (cont)

- Shortcomings
  - circuits are not homogeneous.
  - they have been designed/defined hierarchically.
  - higher connectivity at lower levels.
  - these features are difficult to model statistically across designs and too expensive to predict for one design.
- Positive Aspects
  - very fast
  - reasonable approximation



8 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Constructive Prediction

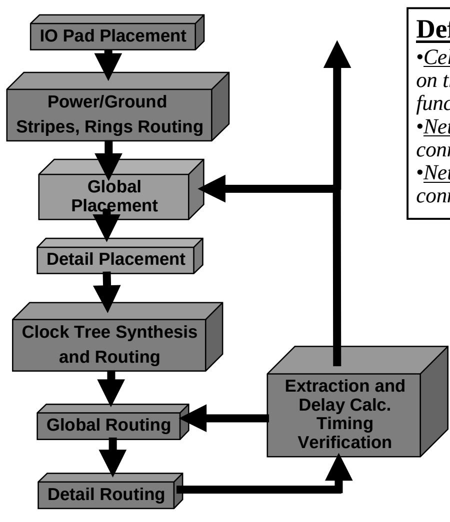
- Generally: the concept of fast algorithms
  - a quick/low-temperature annealing.
  - Final routabililty is correlated with first-level mincut?
- SLIP position statement and some recent results (a few ideas will be discussed).
- E.g., Floorplan based on a given verilog hierarchy
- E.g., Construct fast layouts to predict final timing violations, routability information, etc.

Constructive Prediction (cont)

- Usefulness
  - xKy (K means knowledgeable) type of applications which may require some critical parameters from x to be fed back to y engine.
  - Allows quick exploration.
  - The predictor itself can act as the front-end for final Construction (time spent is not really wasted).
- Shortcomings
  - Slow
  - Can we trust it? (low-temperature annealing)
  - Would it localize the search too much?

Placement Paradigms

VLSI Design Flow and Physical Design Stage



Definitions:
- *Cell: a circuit component to be placed on the chip area. In placement, the functionality of the component is ignored.*
- *Net: specifying a subset of terminals, to connect several cells.*
- *Netlist: a set of nets which contains the connectivity information of the circuit.*

Placement Problem

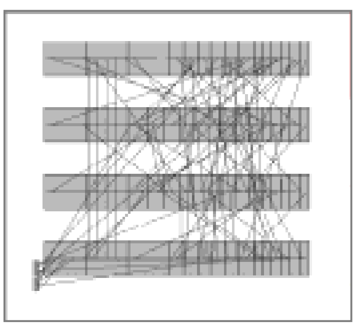
Input:
- A set of cells and their complete information (a cell library).
- Connectivity information between cells (netlist information).

Output:
A set of locations on the chip: one location for each cell.

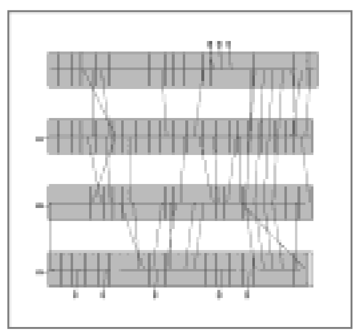
Goal:
The cells are placed to produce a routable chip that meets timing (low-power, …)

Challenge:
- The number of cells in a design is very large (> 1 million).
- The timing constraints are very tight.

Placement Problem



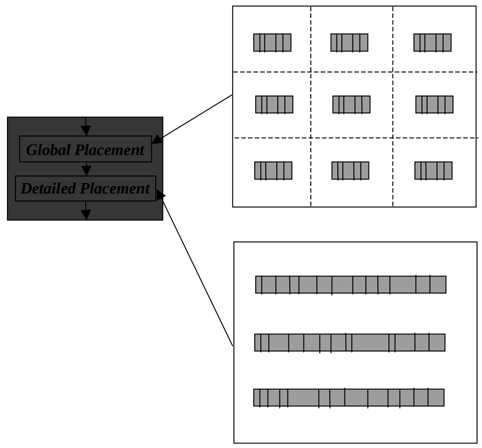
A bad placement



A good placement

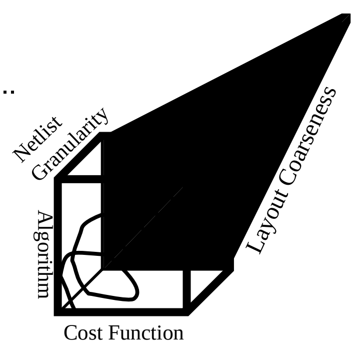
Global and Detailed Placement

In global placement, we decide the approximate locations for cells by placing cells in global bins. In detailed placement, we make some local adjustment to get the final non-overlapping placement.



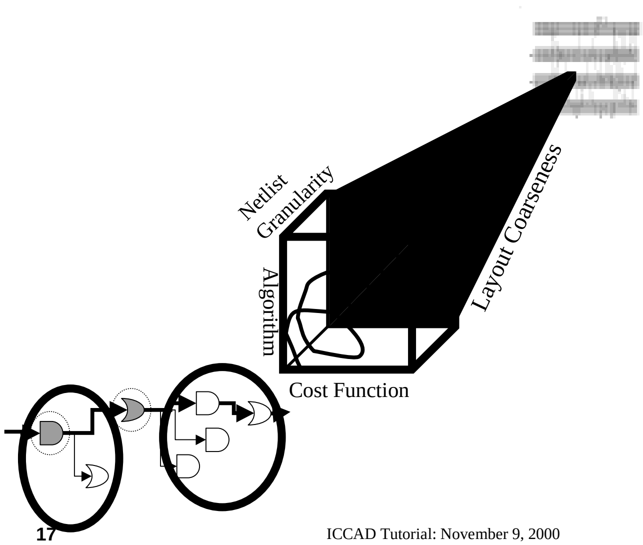
Placement Cube (4d)

- Cost Function(s) to be used
  - Cut, wirelength, congestion, crossing, ...
- Algorithm(s) to be used
  - FM, Quadratic, annealing, ….
- Granularity of the netlist
- Coarseness of the layout domain
  - 2x2, 4x4, ….



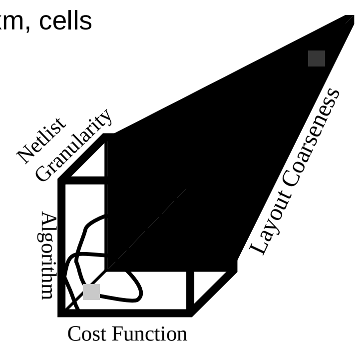
- An effective methodology picks the right mix from the above and knows when to switch from one to next.
- Today: Ad-hoc

What does the cube represent?



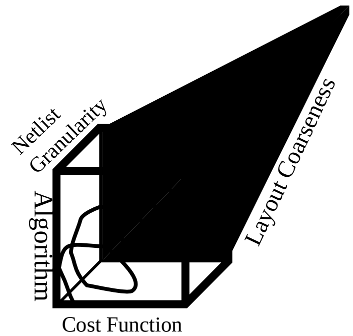
Twolf Global Placement

- Two points in the entire cube
  - Annealing, WL, mxm, clusters
  - Annealing, WL, mxm, cells



Traditional Algorithms

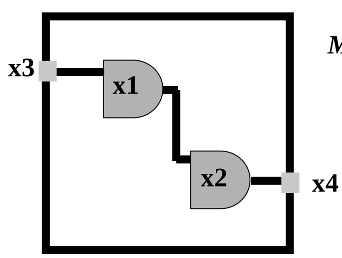
- Quadratic Placement
- Simulated Annealing
- Bi-Partitioning / Quadrisection
- Force Directed Placement
- Hybrid



19 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Quadratic Placement

☞Analytical Technique



Min **[(x1-x3)^2 + (x1-x2)^2 + (x2-x4)^2] : F**

δF/δx1 = 0;

δF/δx2 = 0;

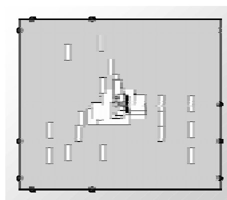
Ax = B

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad B = \begin{bmatrix} x3 \\ x4 \end{bmatrix} \quad x = \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

20 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Analytical Placement

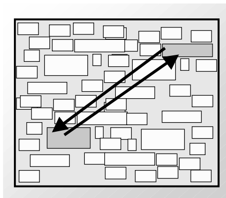
- Get a solution with lots of overlap
- What do we do with the overlap?



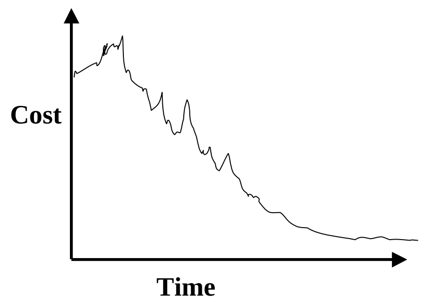
Pros and Cons of QP

- ☞Pros:
  - ☞Very Fast Analytical Solution
  - ☞Can Handle Large Design Sizes
  - ☞Can be Used as an Initial Seed Placement Engine
- ☞Cons:
  - ☞Not Suitable for Timing Driven Placement
  - ☞Not Suitable for Simultaneous Optimization of Other Aspects of Physical Design (clocks, crosstalk…)
  - ☞Gives Trivial Solutions without Pads (and close to trivial with pads)

Simulated Annealing Placement



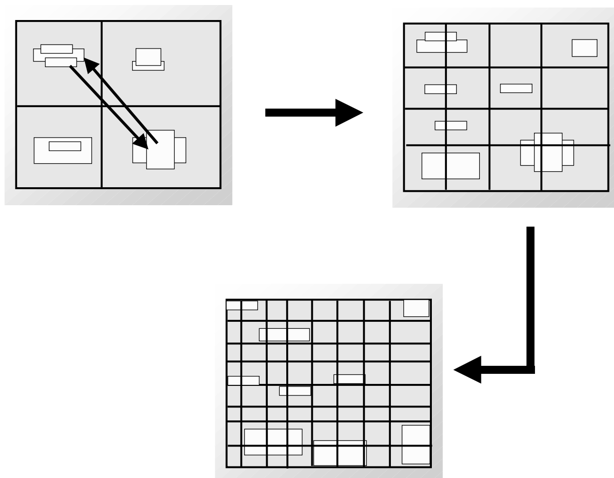
- Initial Placement Improved through Swaps and Moves
- Accept a Swap/Move if it improves cost
- Accept a Swap/Move that degrades cost under some probability conditions



Pros and Cons of SA

- Pros:
  - Can Reach Globally Optimal Solution (given “enough” time)
  - Open Cost Function.
  - Can Optimize Simultaneously all Aspects of Physical Design
  - Can be Used for End Case Placement
- Cons:
  - Extremely Slow Process of Reaching a Good Solution

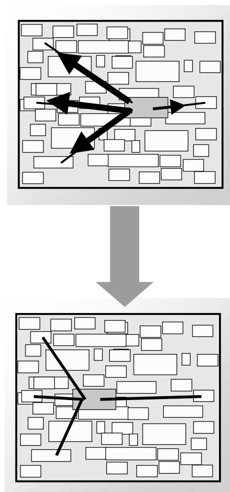
Bi-Partitioning/Quadrisection



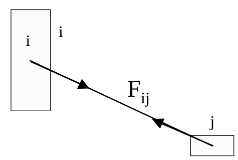
Pros and Cons of Partitioning Based Placement

- ☞Pros:
  - ☞More Suitable to Timing Driven Placement since it is Move Based
  - ☞New Innovation (hMetis) in Partitioning Algorithms have made this Extremely Fast
  - ☞Open Cost Function
  - ☞Move Based means Simultaneous Optimization of all Design Aspects Possible
- ☞Cons:
  - ☞Not Well Understood
  - ☞Lots of “indifferent” moves
  - ☞May not work well with some cost functions.

Force Directed Placement



- Cells are dragged by forces.
- Forces are generated by nets connecting cells. Longer nets generate bigger forces.
- Placement is obtained by either a constructive or an iterative method.



Pros and Cons of Force Directed Placement

- Pros:
  - Very Fast Analytical Solution
  - Can Handle Large Design Sizes
  - Can be Used as an Initial Seed Placement Engine
  - The force
- Cons:
  - Not sensitive to the non-overlapping constraint
  - Gives Trivial Solutions without Pads
  - Not Suitable for Timing Driven Placement

Hybrid Placement

- ☞Mix-matching different placement algorithms
- ☞Effective algorithms are always hybrid

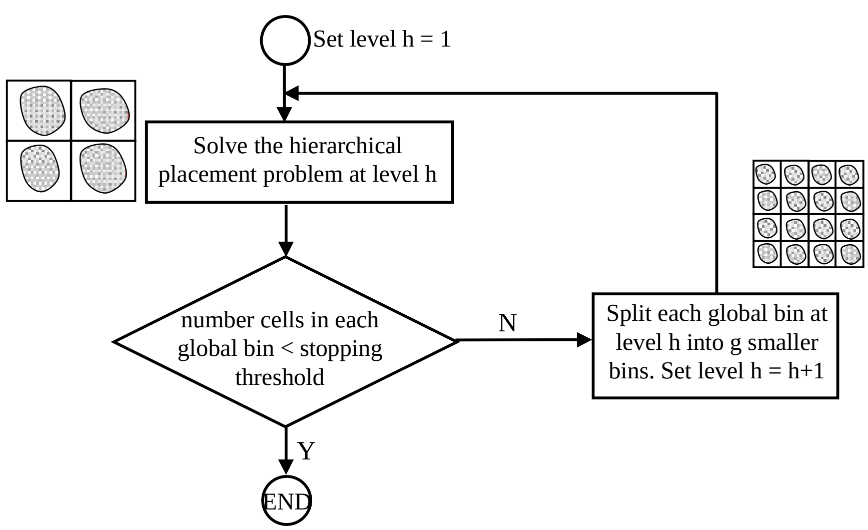
GORDIAN (quadratic + partitioning)

Initial Placement

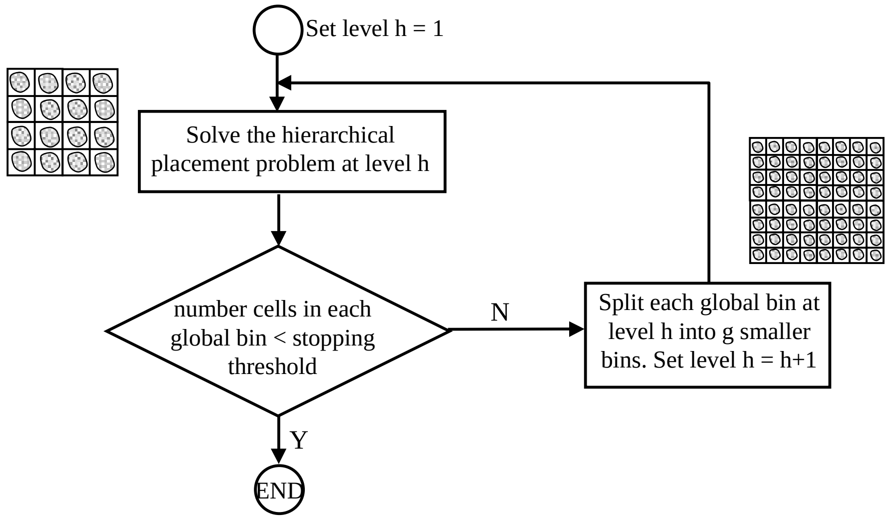
Partition and Replace

Details of a Complete Placement Tool

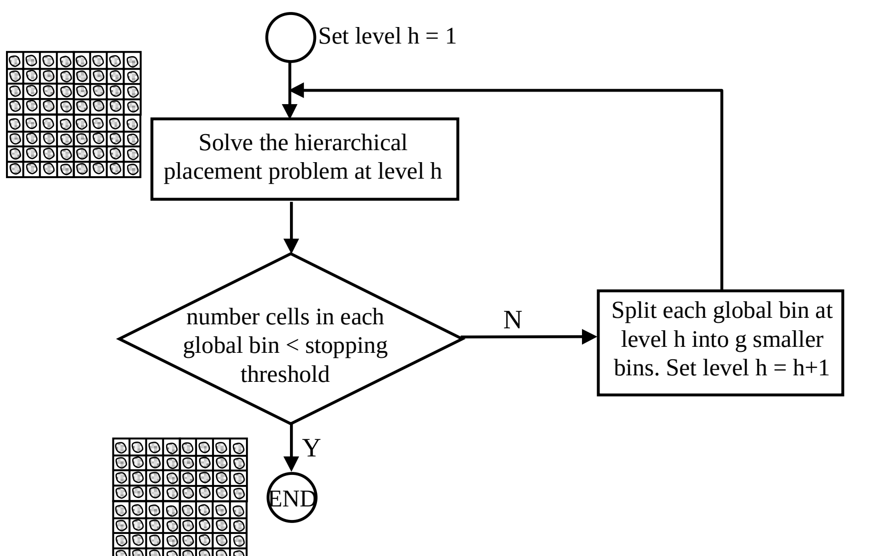
Top-down Hierarchical Placement Approach



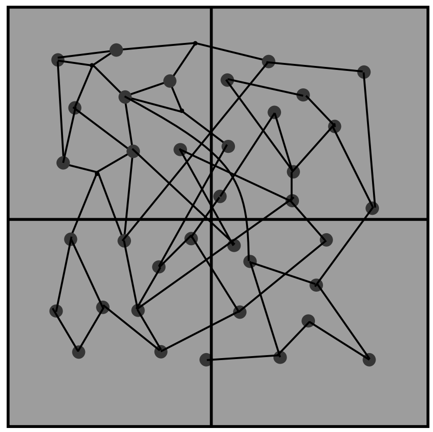
Top-down Hierarchical Placement Approach



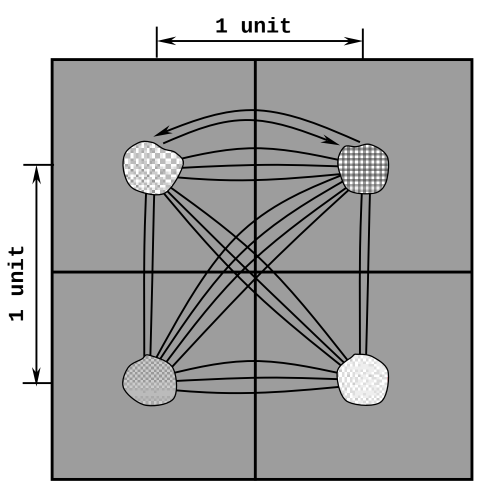
Top-down Hierarchical Placement Approach



Original Subcircuit at B_0

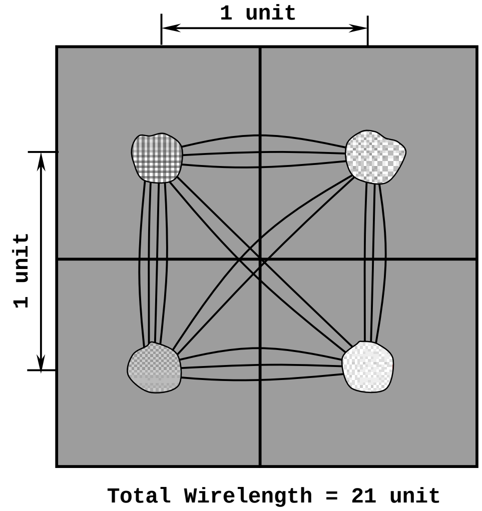


Original Subcircuit at B_0

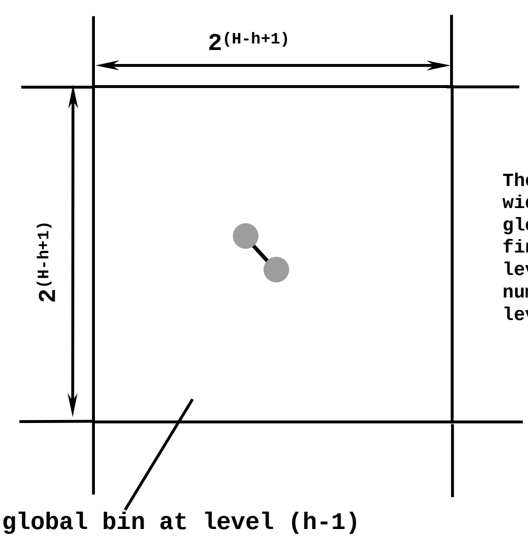


Total Wirelength = 24 unit

Original Subcircuit at B_0

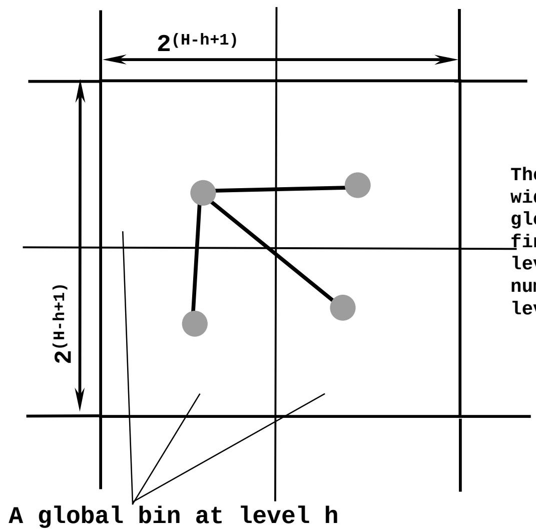


Length of a Level h Net



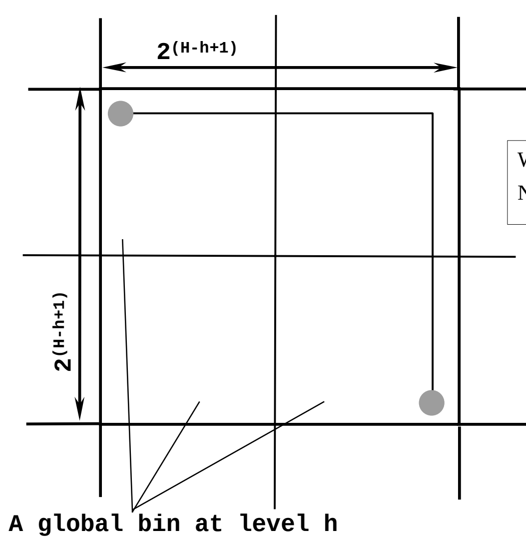
The length unit is the width/height of a global bin in the final hierarchical level (assume total number of hierarchical levels is H).

Length of a Level h Net



The length unit is the width/height of a global bin in the final hierarchical level (assume total number of hierarchical levels is H).

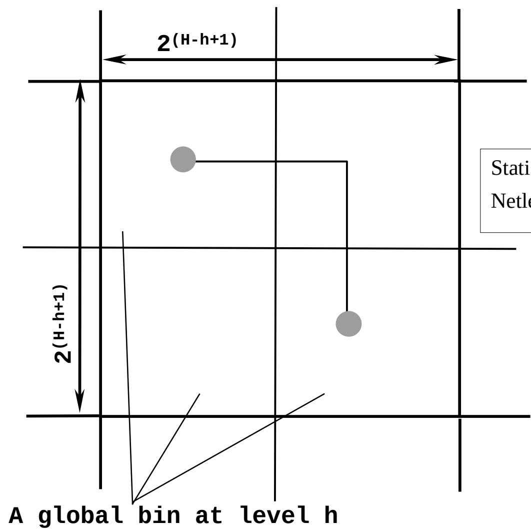
Length of a Level h Net (Worst Case)



Worst Case:

Netlength = $2^{(H-h+2)}$

Length of a Level h Net (Worst Case)



Statistical Average Case:
Netlength = $\frac{7}{9}2^{(H-h+1)}$

Wirelength Bound in a Top-Down Approach

- More nets are cut in finer levels: the number of cut nets at each level is increase by approximately 4^{1-r}, where r is the Rent's parameter of the circuit (0.3~0.8).
- The total wirelength obtained by a top-down approach is between Cut and 2H·Cut

$$Cut \leq Wirelength \leq 2H \cdot Cut = 2\log N_c Cut$$

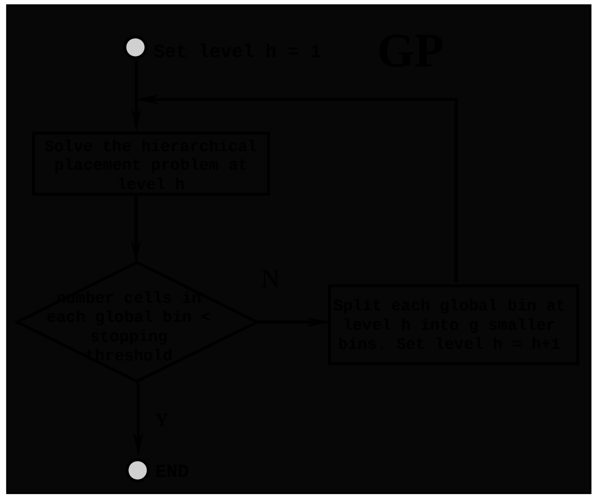
Dragon2000

START

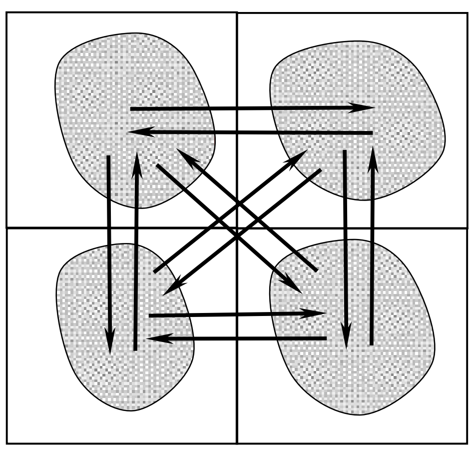
[redacted]

Detailed Placement (DP)
Greedy heuristic

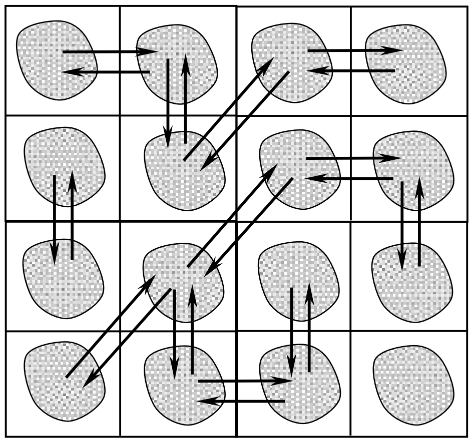
END



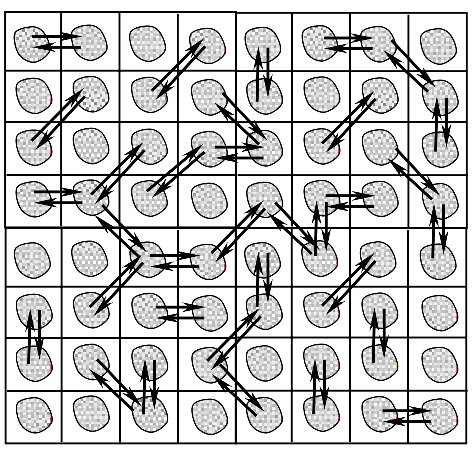
GP Phase (Start)



GP Phase



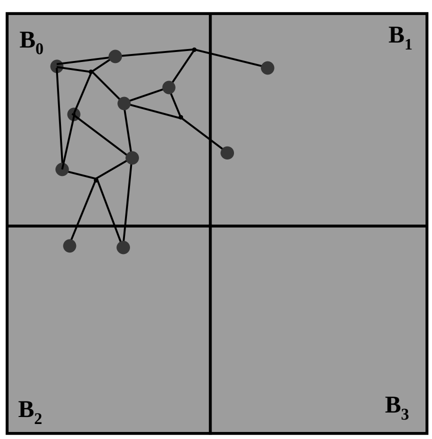
GP Phase



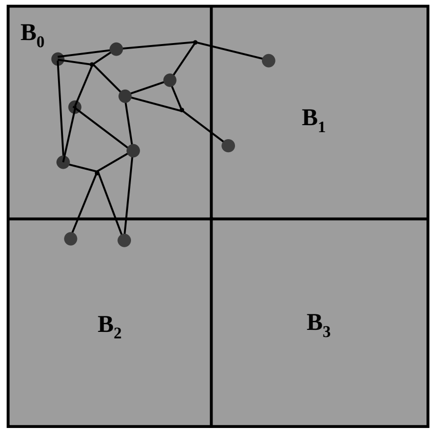
Global Placement Phase

- Start of GP: We start GP phase with four global bins (2×2 grid)
- End of GP: We stop GP when there is less than 7 cells in each global bin
- Interaction between net-cut and wirelength in GP
- Final Stage

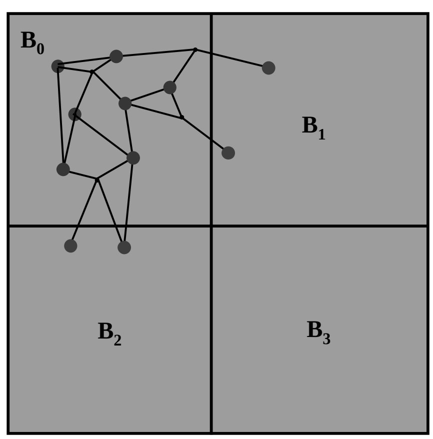
Original Subcircuit at B_0



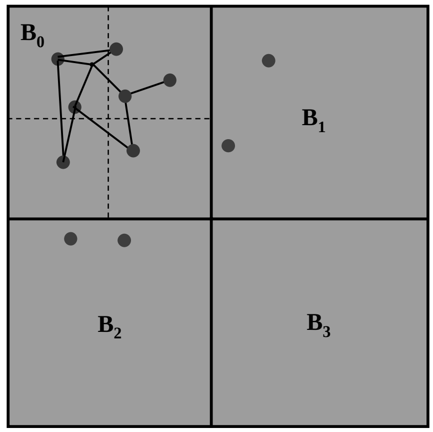
Original Subcircuit at B_0



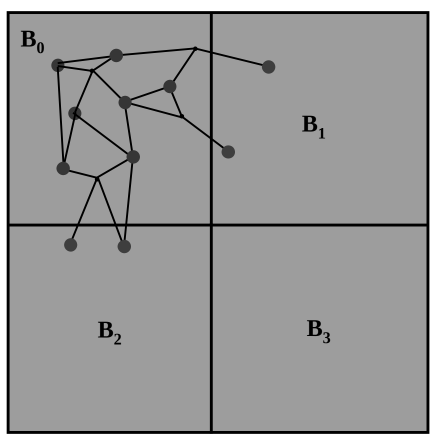
Approach A: Remove External Nets



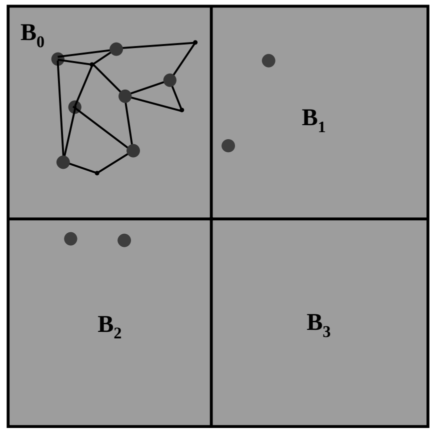
Approach A: Remove External Nets



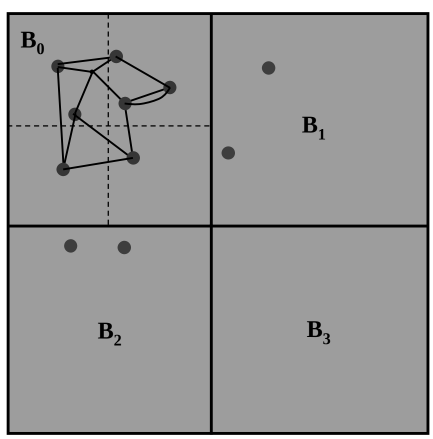
Approach B: Remove External Pins



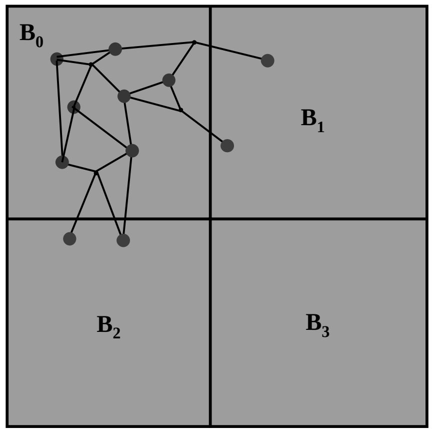
Approach B: Remove External Pins



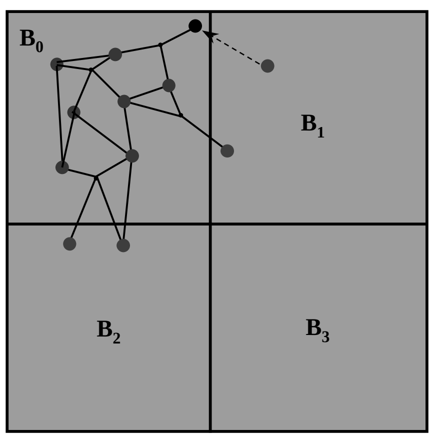
Approach B: Remove External Pins



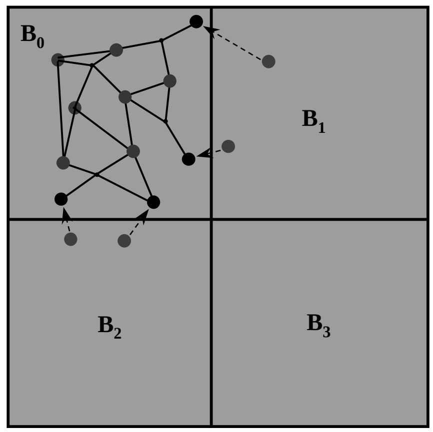
Approach C: Terminal Propagation



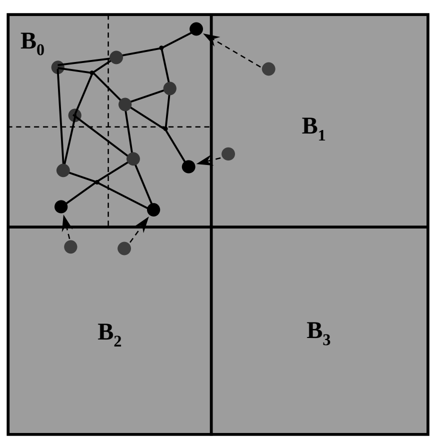
Approach C: Terminal Propagation



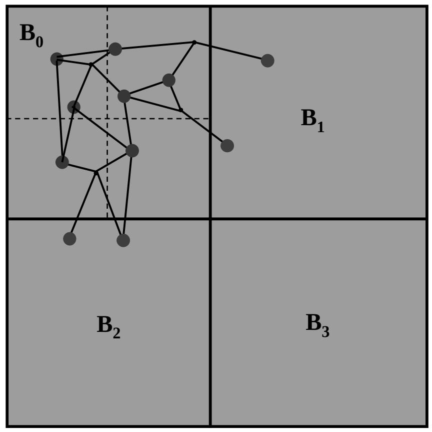
Approach C: Terminal Propagation



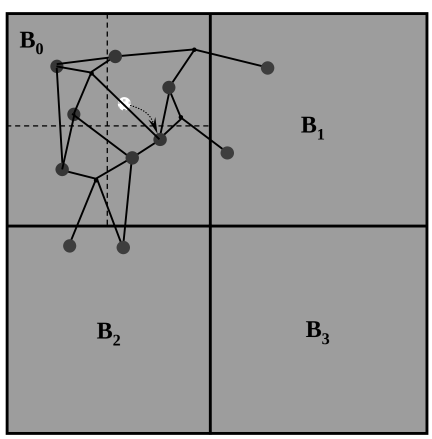
Approach C: Terminal Propagation



Approach D: Single Cell Move for Wirelength Reduction



Approach D: Single Cell Move for Wirelength Reduction

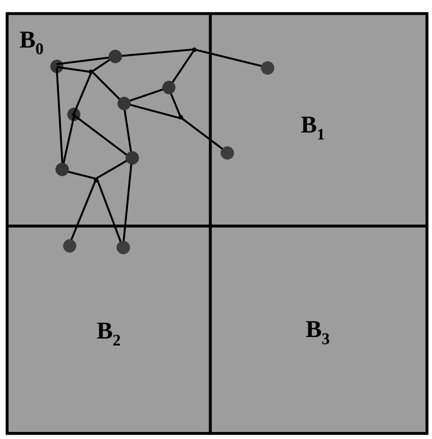


Comparison Between Four Approaches

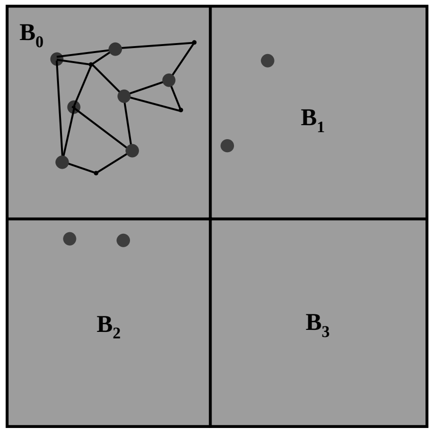
circuit	App. A	App. B	App. C	App. D
ibm01	4.79	4.71	4.98	4.81
ibm02	13.70	13.91	14.38	13.99
ibm03	13.12	12.83	13.02	12.93
ibm04	17.66	16.58	17.54	17.21
ibm05	38.94	38.21	39.32	39.12

Approach B is the best!

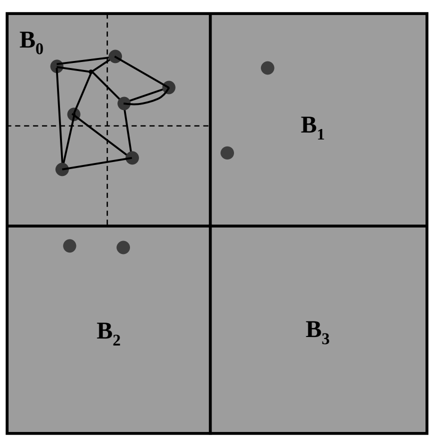
Approach B: Remove External Pins



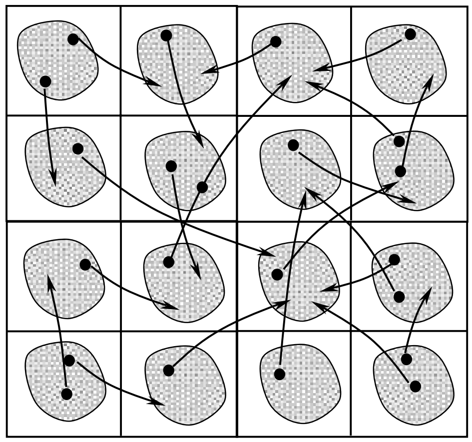
Approach B: Remove External Pins



Approach B: Remove External Pins



Final Stage of GP (wirelength reduction by single cell moves)



Effect of the Final Stage of GP

Ckts	w/o final stage	w/ final stage	% impr.
ibm01	4.99	4.70	5.8%
ibm02	14.71	13.76	6.5%
ibm03	13.56	12.74	6.0%
ibm04	17.07	15.79	7.5%
ibm05	42.19	38.57	8.6%
avg			6.88%

Detailed Placement Heuristics

- Simulated Annealing (SA) has been widely used at DP stage in other placement tools.
- SA is very slow. Thus DP is the most time consuming part in some commercial placement tools (DP in iTools consumes more than 80% of the total running time).
- We use a greedy heuristic to shorten the runtime of DP. The quality of our GP stage assures we have an excellent final layout.

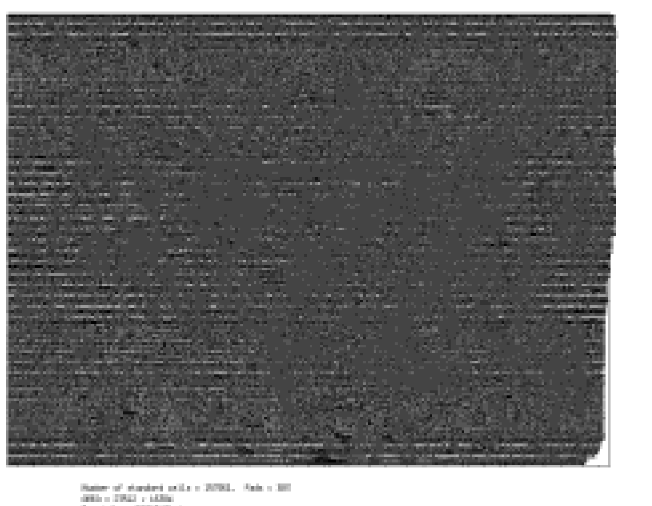
Placement Results for Circuits Larger Than 60k Cells

Ckts	iTools1.4.0		Dragon		Comparison	
	WL	time	WL	time	impr.	spdup
ibm11	39.76	18251	40.82	10301	-2.6%	1.8×
ibm12	69.56	18075	70.38	14198	-1.2%	1.3×
ibm13	49.11	22577	51.02	15456	-3.9%	1.5×
ibm14	118.8	43057	118.0	31894	0.7%	1.4×
ibm15	130.6	54262	130.8	22808	0.0%	2.4×
ibm16	163.8	70320	168.8	39001	-3.0%	1.8×
ibm17	256.6	72094	255.1	38752	0.5%	1.9×
ibm18	191.7	75363	189.6	39603	1.1%	1.9×
golem3	85.44	24380	77.56	8422	9.2%	2.9×
ave	(average for circuit > 60k cells.)				0.1%	1.9×
ave*	(average for circuit > 100k cells.)				1.4%	2.1×

Placement Results for MCNC Benchmark Circuits

Ckts	TW7.0	TUM	iTools1.4.0		Dragon	
	WL	WL	WL	time	WL	time
ind2	13.53	14.6	12.30	1537	12.88	1461
in3	42.84	45.1	40.13	3154	42.33	2849
avqs	5.41	4.91	4.84	1915	5.17	1420
avql	5.86	5.38	5.19	2043	5.25	1984
golem3	90.39	-	85.44	24380	77.56	8422

Output Layout from Dragon2000 (ibm15, 157k cells)



Conclusion (Wirelength Reduction)

- We studied properties of net-cut and wirelength objective
- Net-cut should be used as an effective shortcut to minimize wirelength in top-down hierarchical placement
- We developed Dragon2000 placement tool which integrate net-cut and wirelength together
- Dragon2000 can produce good layout for large industrial circuits

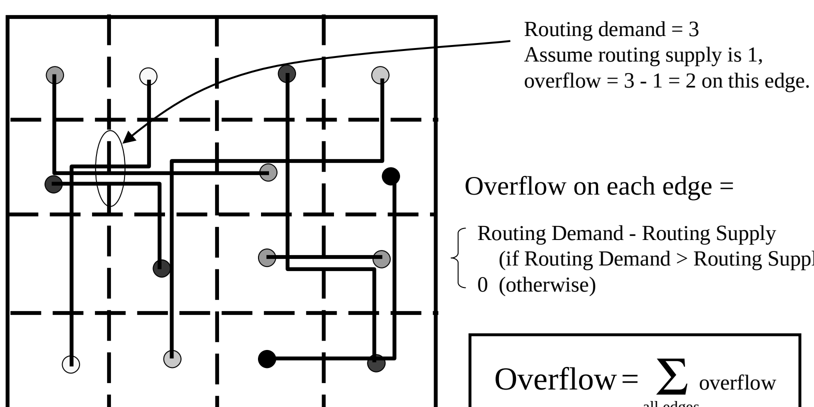
Congestion Minimization During Placement

Motivation (part I)

- Traditional placement problem is to minimize interconnection length (wirelength)
- A valid placement has to be routable
- Congestion is important because it represents routability (lower congestion implies better routability)
- There is not enough research work on the congestion minimization problem yet

73 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Definition of Congestion



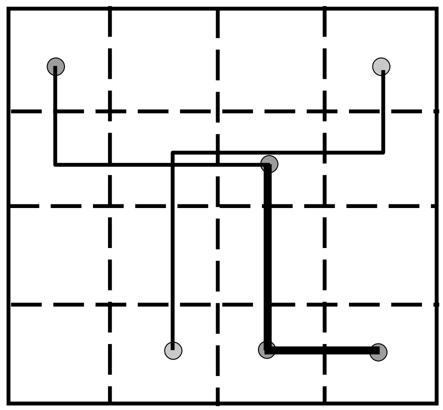
Overflow on each edge =

- Routing Demand - Routing Supply (if Routing Demand > Routing Supply)
- 0 (otherwise)

$$\text{Overflow} = \sum_{\text{all edges}} \text{overflow}$$

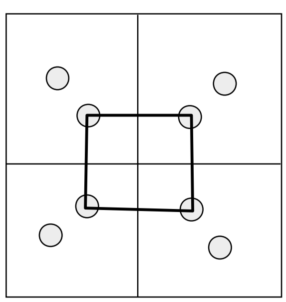
74 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Correlation between Wirelength and Congestion

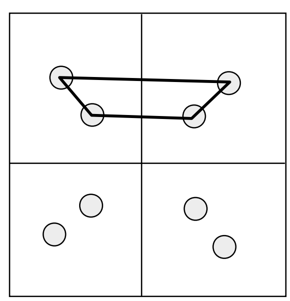


Total Wirelength = Total Routing Demand

Wirelength ≠ Congestion

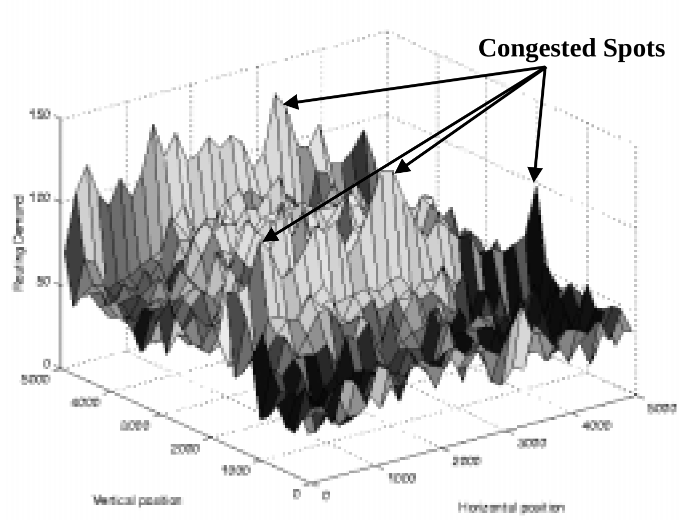


A congestion minimized placement



A wirelength minimized placement

Congestion Map of a Wirelength Minimized Placement



Post Processing to Reduce Congestion

Reduce congestion globally by minimizing the traditional wirelength

Post process the wirelength optimized placement using the congestion objective

Summary

- Among a variety of cost functions and methods for congestion minimization (including several currently used in industry), wirelength alone followed by a post processing congestion minimization works the best and is one of the fastest.
- Cost functions such as a hybrid length plus congestion (commonly believed to be very effective) do not work very well.

Multi-center Congestion Estimation and Minimization During Placement

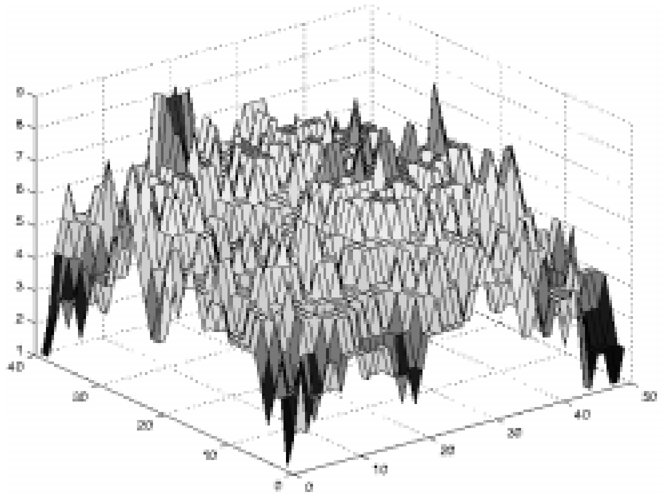
Motivation (part II)

- Congestion is globally consistent with wirelength
- Congestion locally compete with wirelength
- It is good to perform congestion optimization at a later stage
- At a later stage of placement, layout cannot be changed much in order to satisfy various performance constraints
- We need a method to reduce congestion by making only local changes to the layout

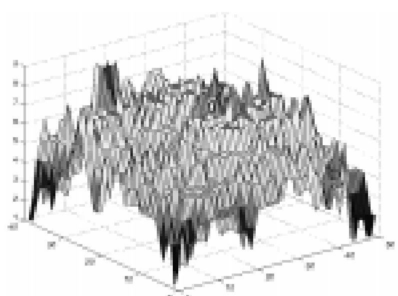
Our Goal and Approaches

- Goal: Reduce congestion in placement while keeping the change of placement bounded
- Approach:
  - Study congestion distribution in layout
  - Find a way to estimate congestion within a local region
  - Perform congestion reduction within local regions.

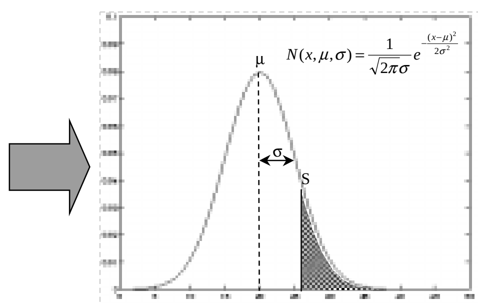
Congestion Map of a Wirelength Minimized Placement (biomed)



Congestion Distribution vs. Normal Distribution

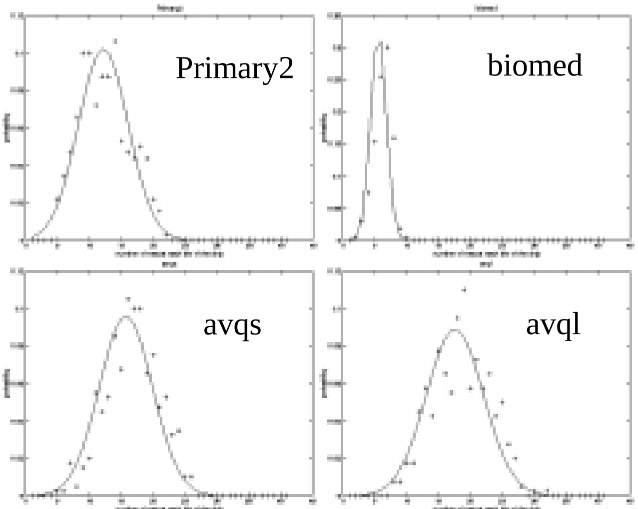


Congestion map in a layout

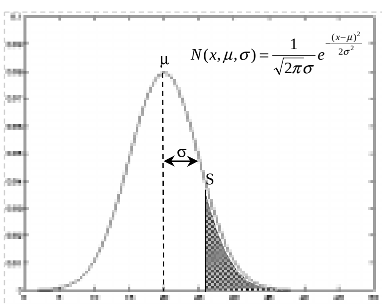


Normal distribution curve $N(x, \mu, \sigma)$: shows percentage of global bins which has x wires.

Congestion Distribution In the Layout



Congestion Estimation Within A Local Region



$$\text{Overflow} = A\int_S^\infty \frac{1}{\sqrt{2\pi}\sigma}(x-S)e^{-\frac{(x-\mu)^2}{2\sigma^2}dx}$$

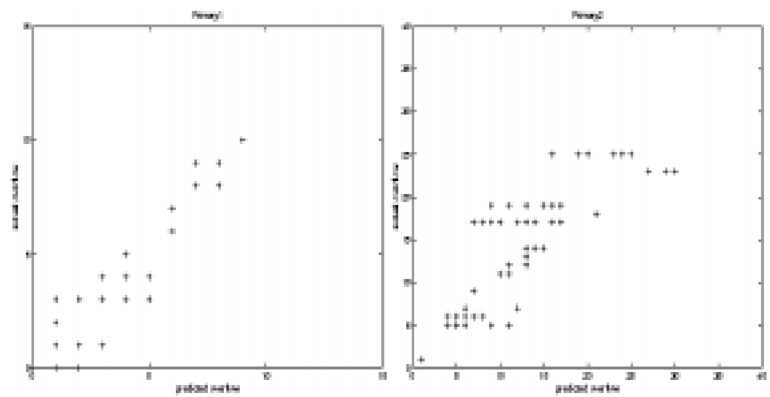
Let $t=\frac{x-\mu}{\sqrt{2}\sigma}, S_t=\frac{S-\mu}{\sqrt{2}\sigma}$

$$\text{Overflow} = \frac{\sigma A}{\sqrt{2\pi}}\int_{S_t}^\infty e^{-t^2}dt^2 + \frac{(\mu-S)A}{\sqrt{\pi}}\int_{S_t}^\infty e^{-t^2}dt$$

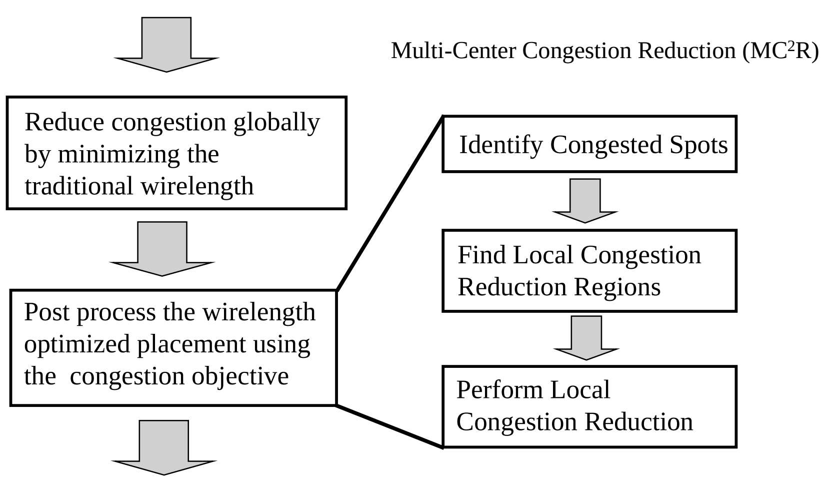
Using standard function $\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^x e^{-t^2}dt$

$$\text{Overflow}(A,S,\mu,\sigma)=\frac{A\sigma}{\sqrt{2\pi}}e^{-\frac{(S-\mu)^2}{2\sigma^2}} + A(\mu-S)\sqrt{2}(1-\Phi(\frac{S-\mu}{\sqrt{2}\sigma}))$$

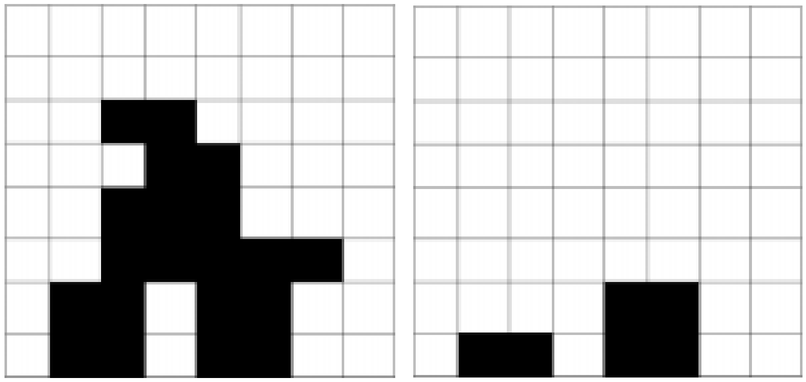
Experimental Validation of Our Congestion Estimation Method



Post Processing and MC^2R

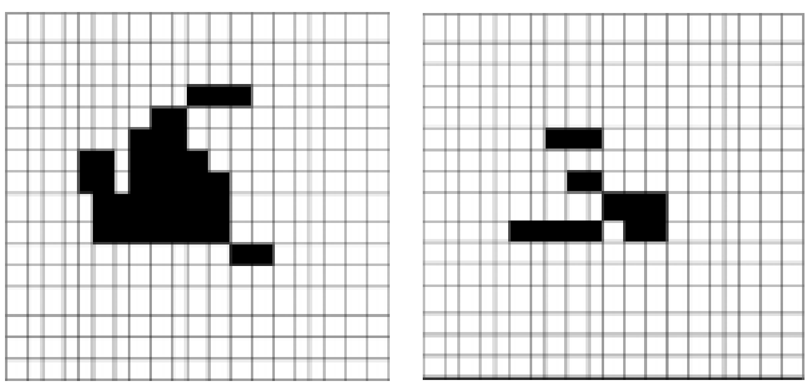


Binary Congestion Map for Primary1



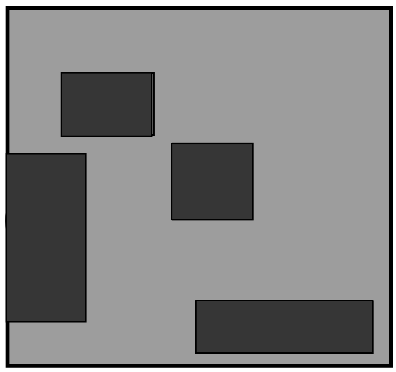
(a) Routing supply = 24 nets/edge (b) Routing supply = 26 nets/edge

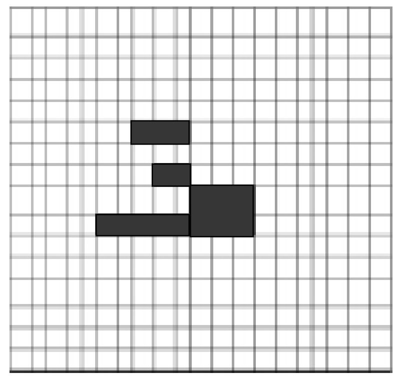
Binary Congestion Map for Primary2



(a) Routing supply = 24 nets/edge (b) Routing supply = 26 nets/edge

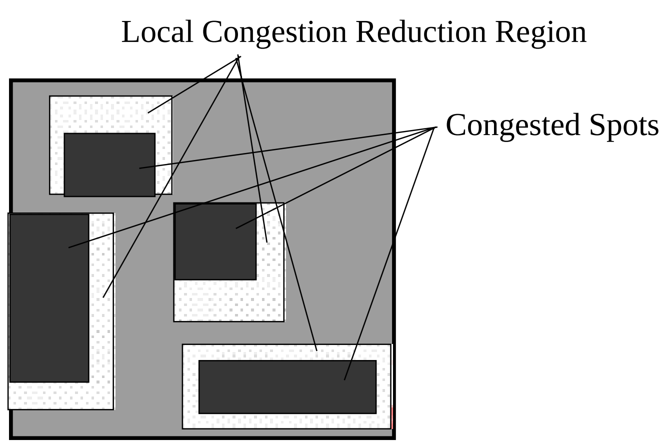
Identify Congested Spots



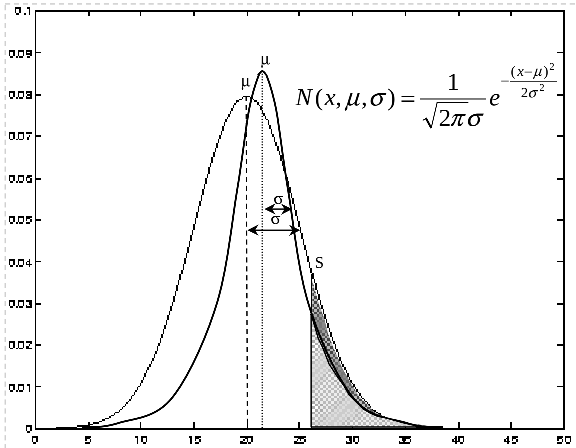


Congested spots are always rectangles.

Local Congestion Reduction Region

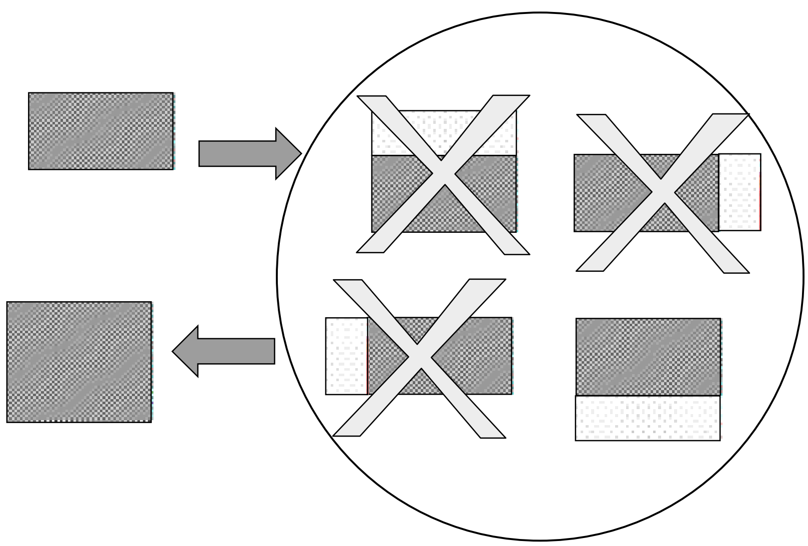


Congestion Estimation Within A Local Region



After performing congestion optimization, μ is expected to increase and σ is expected to decrease. We approximate the increase and the decrease as a constant ratio of the original value.

Expand the Current Evaluating Region



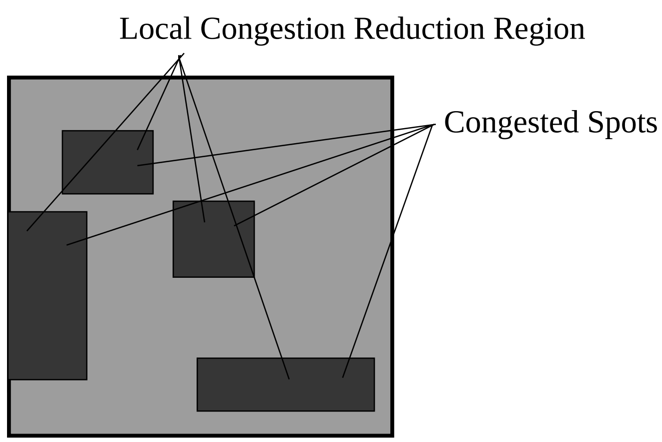
Stopping Criterion for the Expansion

- The estimated routability of the new region is better than an expected value.
- The new region has worse estimated routability than the original region. It means that the neighborhood is more congested than the original evaluating region.
- The new region occupies the whole layout area so that there is no more space to expand.

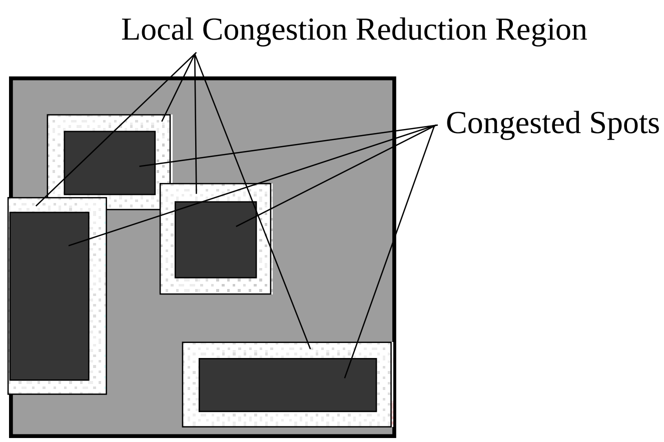
Four Expansion Schemes:

- ***Flexible Expansion*** (Our new scheme)
- Constant Expansion
- Zero Expansion
- Full Expansion

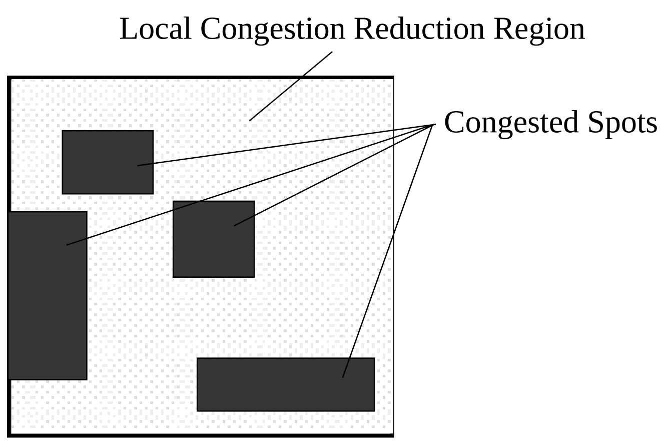
Zero Expansion Scheme



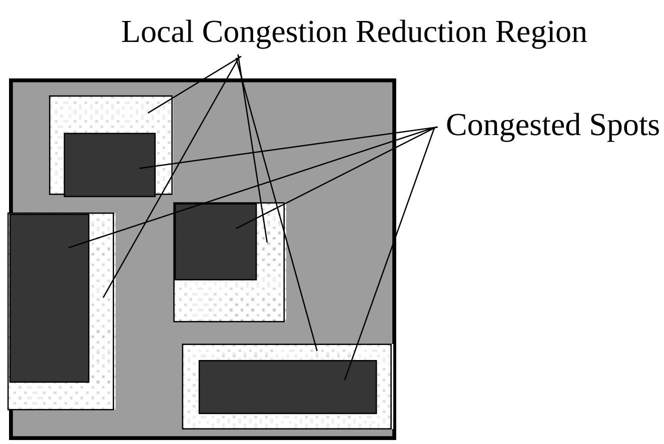
Constant Expansion Scheme



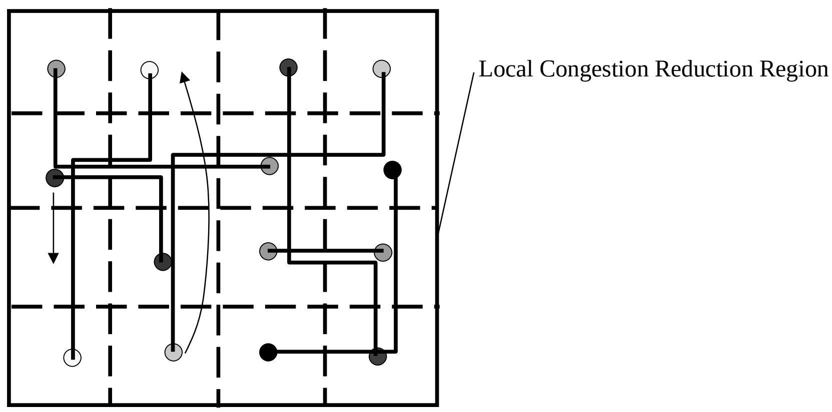
Full Expansion Scheme



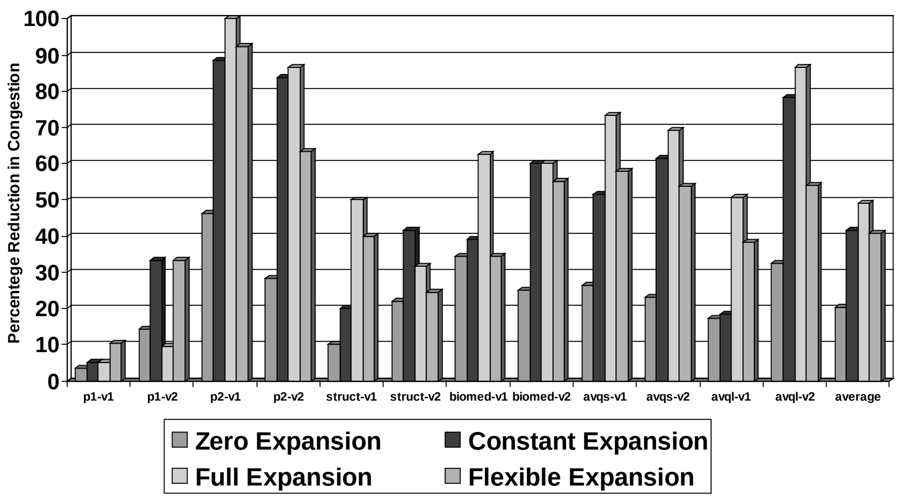
Flexible Expansion Scheme



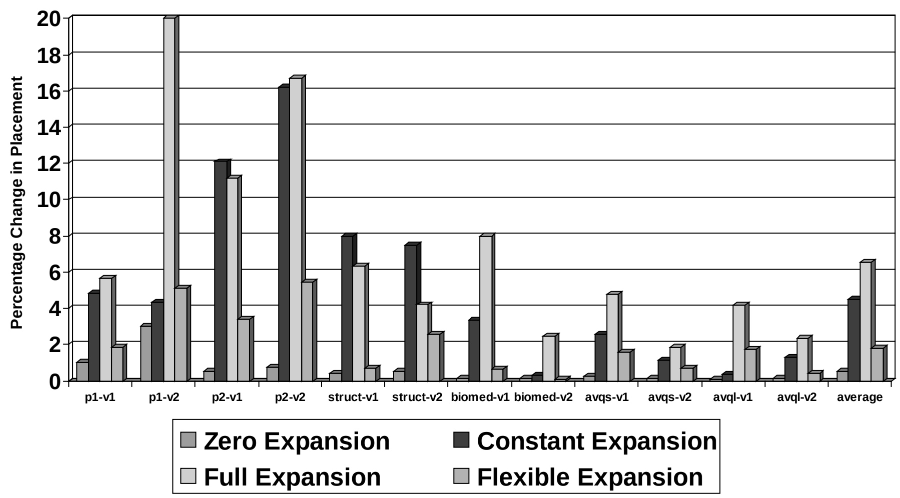
Perform Local Congestion Reduction



Congestion Reduction Comparison



Change in Placement Comparison



103 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Conclusion (Congestion Reduction)

- Wirelength minimization can minimize congestion globally. A post processing congestion minimization following wirelength minimization works the best to reduce congestion in placement
- We use a normal distribution approximation to estimate congestion within a local region
- We propose a flexible expansion scheme to locate the local congestion reduction region
- Experimental results show that our scheme can achieve significant congestion reduction while keeping change of the layout bounded

104 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

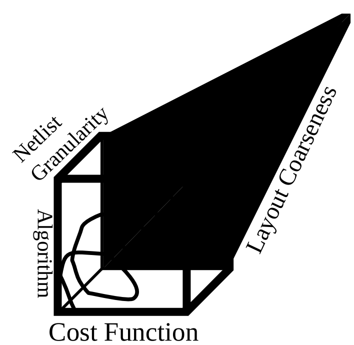
Cost Functions for Placement

- The final goal of placement is to achieve routability and meet timing constraints
- Constraints are very hard to use in optimization, thus we use cost functions (e.g., Wirelength) to predict our goals.
  - We will show what happens when you try constraints directly
  - The main challenge is a technical understanding of various cost functions and their interaction.

105 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

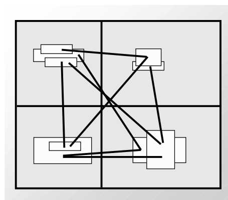
Cost Functions for Placement

- Net-cut
- Linear wirelength
- Quadratic wirelength
- Congestion
- Timing
- Coupling
- Other performance related cost functions
- Undiscovered: crossing



106 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

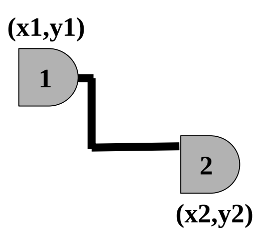
Net-cut Cost for Global Placement



☞The net-cut cost is defined as the number of external nets between different global bins

☞Minimizing net-cut in global placement tends to put highly connected cells close to each other.

Linear Wirelength Cost

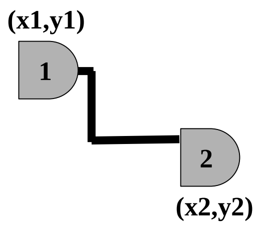


The linear length of a net between cell 1 and cell 2 is

$$l_{12} = |x1-x2| + |y1-y2|$$

The linear wirelength cost is the summation of the linear length of all nets.

Quadratic Wirelength Cost



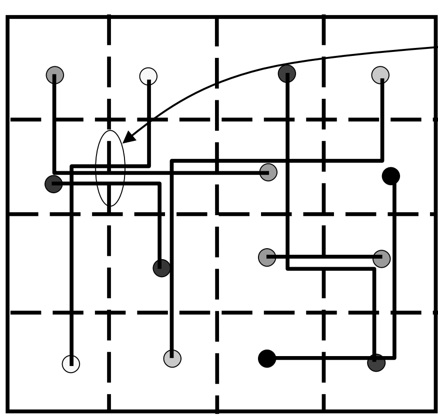
The quadratic length of a net between cell 1 and cell 2 is

$$l_{12} = (x1-x2)^2 + (y1-y2)^2$$

The quadratic wirelength cost is the summation of the quadratic length of all nets.

109 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Congestion Cost



Routing demand = 3
Assume routing supply is 1,
overflow = 3 - 1 = 2 on this edge.

Overflow on each edge =

- Routing Demand - Routing Supply (if Routing Demand > Routing Supply)
- 0 (otherwise)

$$\text{Congestion Overflow} = \sum_{\text{all edges}} \text{overflow}$$

110 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Cost Functions for Placement

- ☞Various cost functions (and a mix of them) have been used in practice to model/estimate routability and timing
- ☞We have a good “feel” for what each cost function is capable of doing
- ☞We need to understand the interaction among cost functions

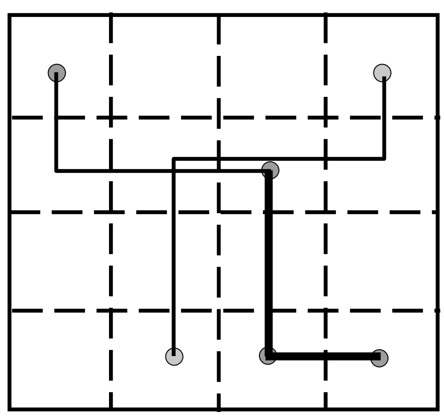
Cut vs. Wirelength

- ☞Globally: In a 2x1 bin, wirelength is the same as cut

Congestion Minimization and Congestion vs Wirelength

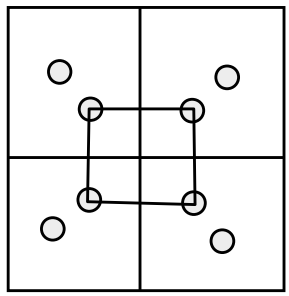
- ☞Congestion is important because it closely represents routability (especially at lower-levels of granularity)
- ☞Congestion is not well understood
- ☞Ad-hoc techniques have been kind-of working since congestion has never been severe
- ☞It has been observed that length minimization tends to reduce congestion.
- ☞Goal: Reduce congestion in placement (willing to sacrifice wirelength a little bit).

Correlation between Wirelength and Congestion

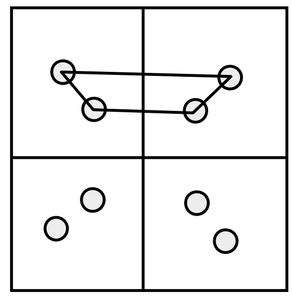


Total Wirelength = Total Routing Demand

Wirelength ≠ Congestion

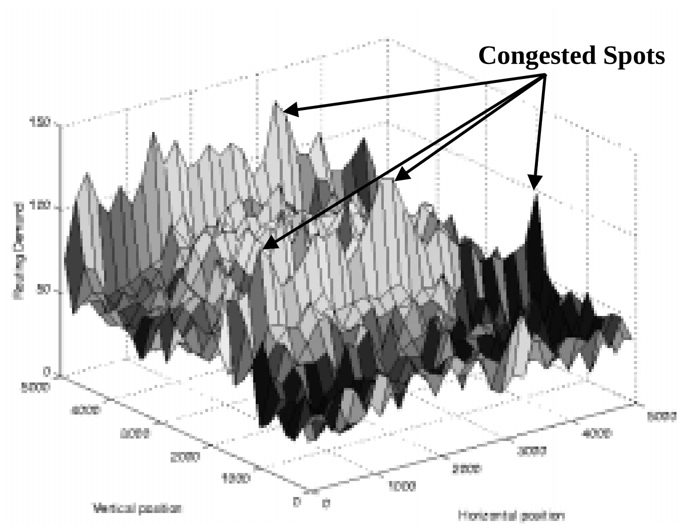


A congestion minimized placement



A wirelength minimized placement

Congestion Map of a Wirelength Minimized Placement

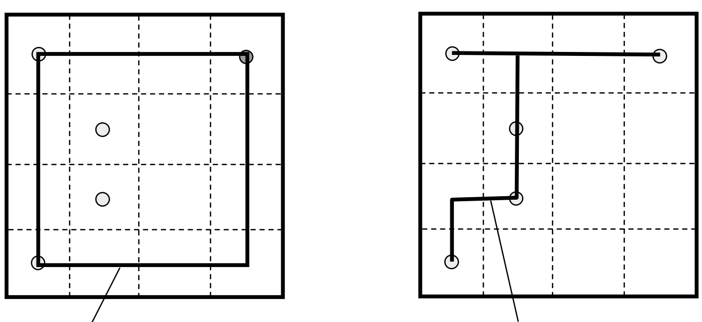


Different Routing Models for modeling congestion

- Bounding box router: fast but inaccurate.
- Real router: accurate but slow.
- A bounding box router can be used in placement if it produces correlated routing results with the real router.

- Note: For different cost functions, answer might be different (e.g., for coupling, only a detailed router can answer).

Different Routing Models



A bounding box routing model

A MST+shortest_path routing model

Correlation Test Between Different Routers

Circuit	RoutModel	A	B	C	D	E	F
Primary1	BBox	14	36	26	27	40	30
	Real	27	9	7	4	5	4
Primary2	BBox	562	163	594	680	147	631
	Real	331	63	378	407	73	378
struct	BBox	949	459	1086	1091	665	1119
	Real	92	294	121	142	414	154
biomed	BBox	4098	2522	7458	7335	3790	6711
	Real	188	48	706	760	180	474

Evaluate overflow value using different routers.

(A, B, C, D, E and F are six independent placements)

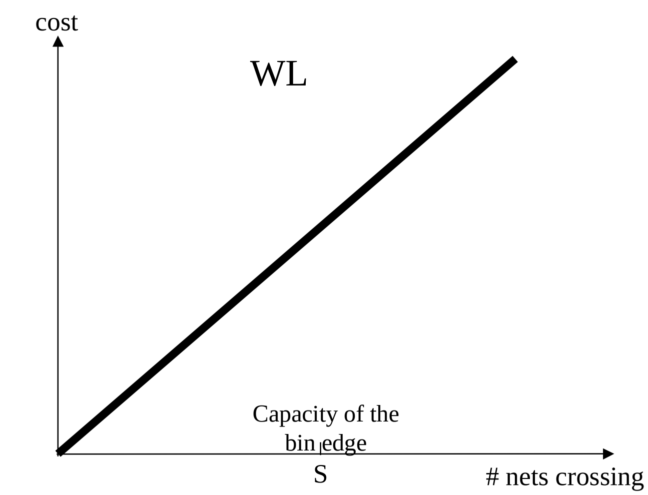
Conclusion:

Bounding box router cannot be used in placement to evaluate congestion.

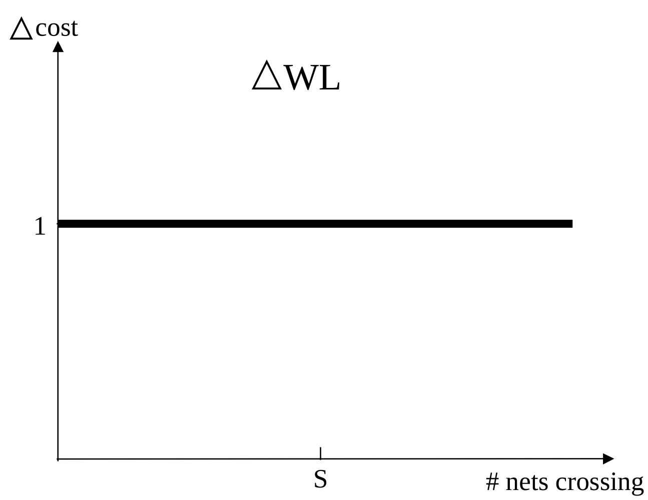
Objective Functions Used in Congestion Minimization

- WL: Standard total wirelength objective.
- Ovrflw: Total overflow in a placement (a direct congestion cost).
- Hybrid: (1- α)WL + α Ovrflw
- QL: A quadratic plus linear objective.
- LQ: A linear plus quadratic objective.
- LkAhd: A modified overflow cost.
- (1- α_T)WL + α_T Ovrflw: A time changing hybrid objective which let the cost function gradually change from wirelength to overflow as optimization proceeds.

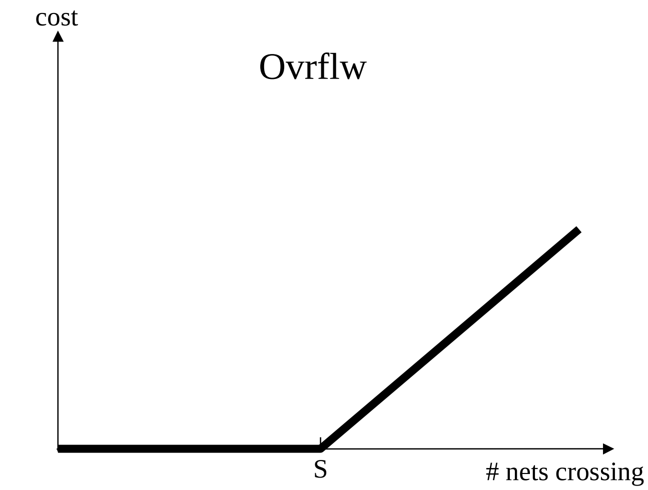
Wirelength Cost



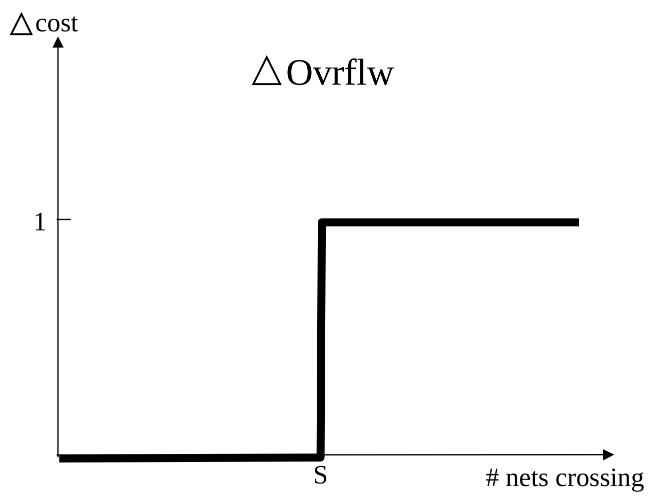
Delta Wirelength Cost



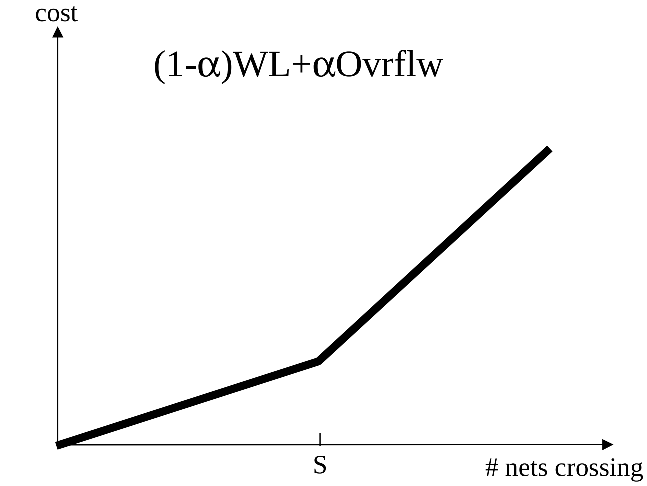
Overflow Cost



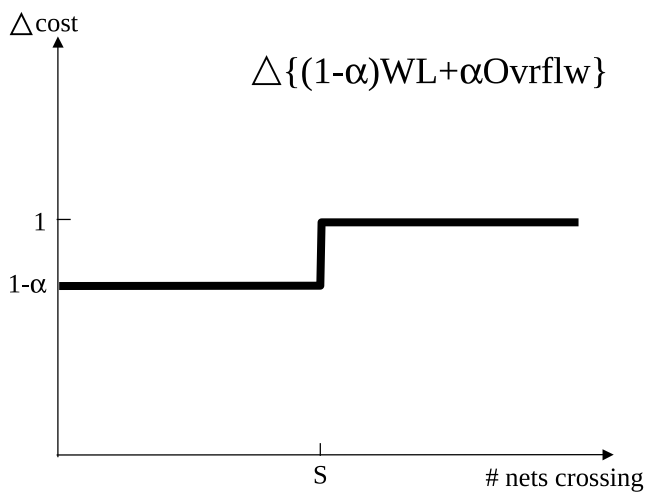
Delta Overflow Cost



Hybrid Cost



Delta Hybrid Cost



Look-ahead overflow cost

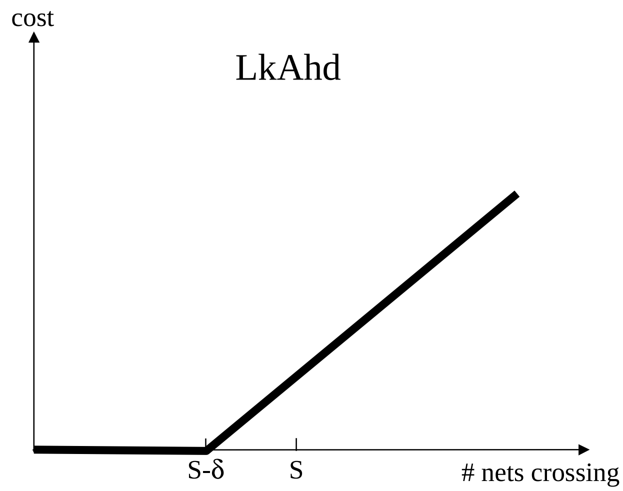
Overflow on each edge = { Routing Demand - Routing Supply (if Routing Demand > Routing Supply); 0 (otherwise)

$$\text{Total Overflow} = \sum_{\text{all edges}} \text{overflow}$$

Overflow with look-ahead = { Routing Demand - (Routing Supply - δ) (if Routing Demand > Routing Supply - δ); 0 (otherwise)

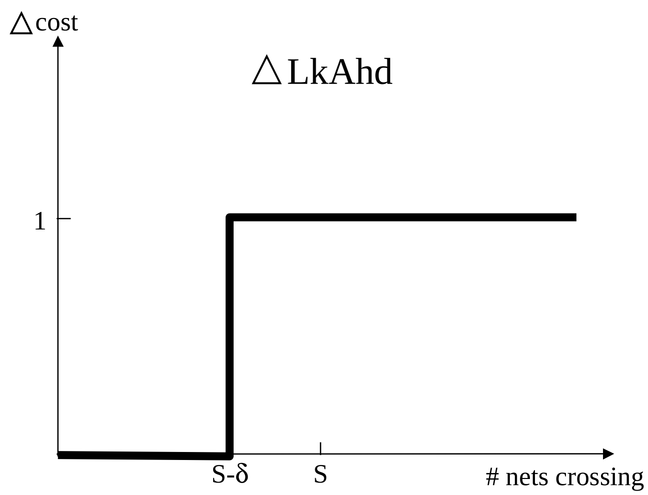
127 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Look-ahead Cost

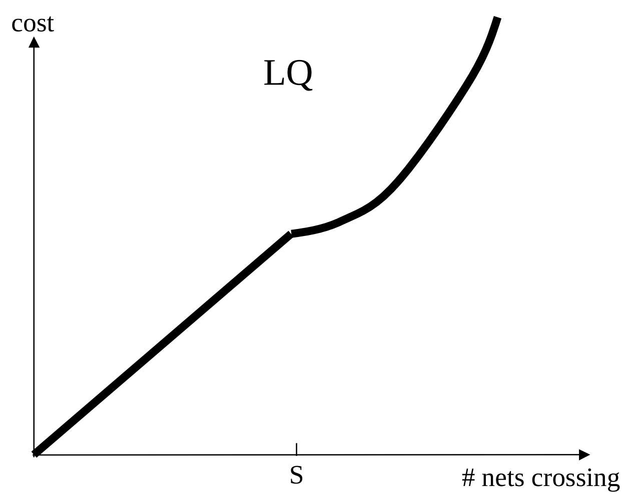


128 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

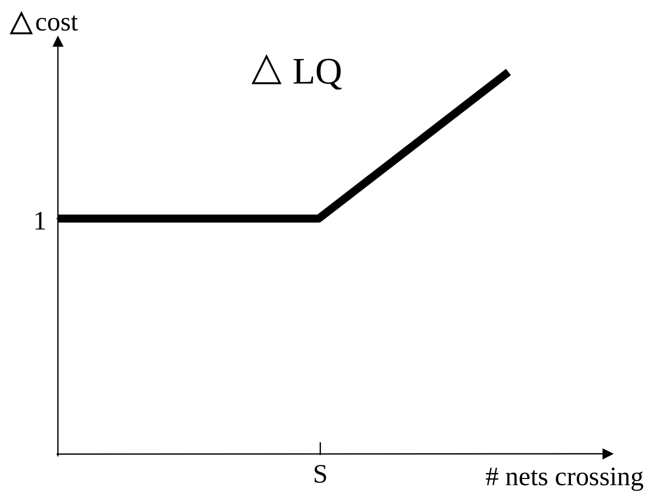
Delta Look-ahead Cost



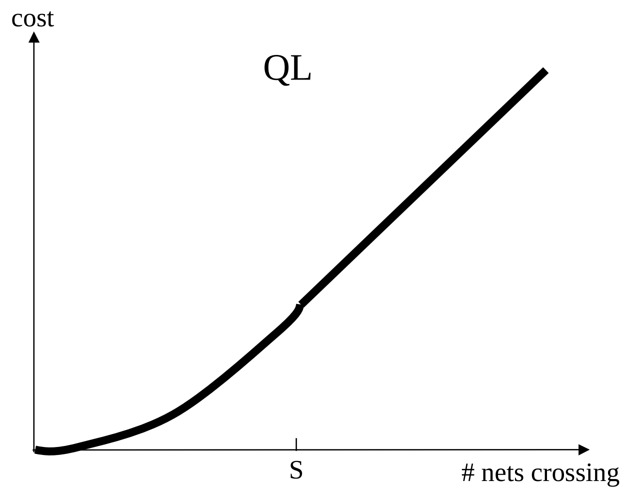
Linear-Quadratic Cost



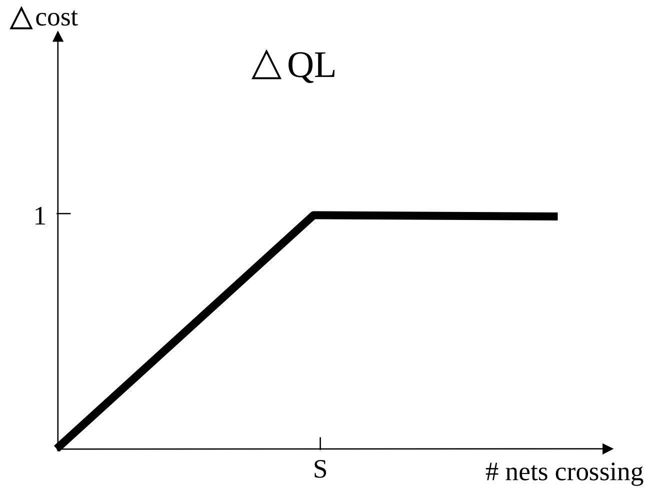
Delta Linear-Quadratic Cost



Quadratic-Linear Cost



Delta Quadratic-Linear Cost



Comparison Between Different Objectives

	wirelength	overflow	runtime
WL	27885	3011	643
Ovrflw	57992	20400	116050
$0.8WL + 0.2Ovrflw$	53289	20982	51001
$0.6WL + 0.4Ovrflw$	56993	23399	53398
$0.5WL + 0.5Ovrflw$	58016	23768	50074
$0.4WL + 0.6Ovrflw$	59434	24954	49283
$0.2WL + 0.8Ovrflw$	62450	27063	49884
$(1 - \alpha_I)WL + \alpha_I Ovrflw$	65233	29486	47300
LkAhd	70346	32367	43523
QL	65532	27738	47426
LQ	67786	30846	48212

Comparison between different objectives for circuit biomed.

Comparison Between Different Objectives

	wirelength	overflow	runtime
WL	26918	151	269
Ovrflw	80425	6391	9116
$0.8WL + 0.2Ovrflw$	79918	9406	17103
$0.6WL + 0.4Ovrflw$	81704	9149	17108
$0.5WL + 0.5Ovrflw$	84586	9660	17145
$0.4WL + 0.6Ovrflw$	89734	10883	17167
$0.2WL + 0.8Ovrflw$	96108	12052	17517
$(1-\alpha_T)WL + \alpha_T Ovrflw$	100869	13055	17761
LkAhd	77823	5613	9267
QL	66086	4231	11600
LQ	75090	6298	10284

Comparison between different objectives for circuit Primary2.

Comparison Between Different Objectives

	wirelength	overflow	runtime
WL	9110	159	5839
Ovrflw	718451	130410	93381
$0.8WL + 0.2Ovrflw$	651406	117992	89283
$0.6WL + 0.4Ovrflw$	655704	118569	93330
$0.5WL + 0.5Ovrflw$	658943	118994	89081
$0.4WL + 0.6Ovrflw$	660134	119084	90385
$0.2WL + 0.8Ovrflw$	661199	119243	90469
$(1-\alpha_T)WL + \alpha_T Ovrflw$	698035	126173	60884
LkAhd	711535	128970	61417
QL	669985	120612	59896
LQ	718701	130538	61840

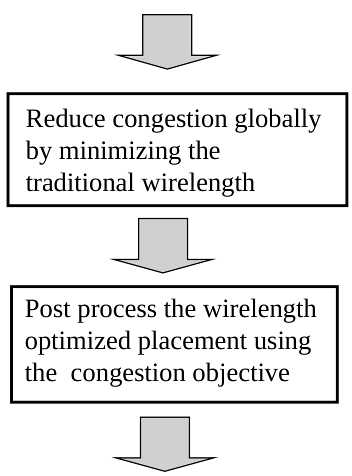
Comparison between different objectives for circuit avqs.

Comparison Between Different Objectives

	wirelength	overflow	runtime
WL	107261	802	7934
Ovrflw	879751	160520	113085
$0.8WL + 0.2Ovrflw$	832858	153260	110778
$0.6WL + 0.4Ovrflw$	838492	159306	119350
$0.5WL + 0.5Ovrflw$	839052	159465	113754
$0.4WL + 0.6Ovrflw$	842840	153849	117805
$0.2WL + 0.8Ovrflw$	849358	159374	110485
$(1-\alpha_T)WL + \alpha_T Ovrflw$	859994	156729	72723
LkAhd	881915	161172	71997
QL	840739	152345	72526
LQ	879860	160625	72593

Comparison between different objectives for circuit avql.

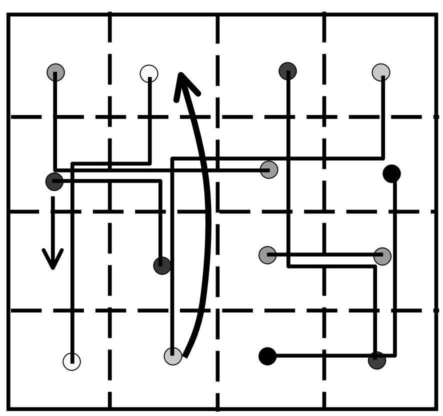
Post Processing to Reduce Congestion



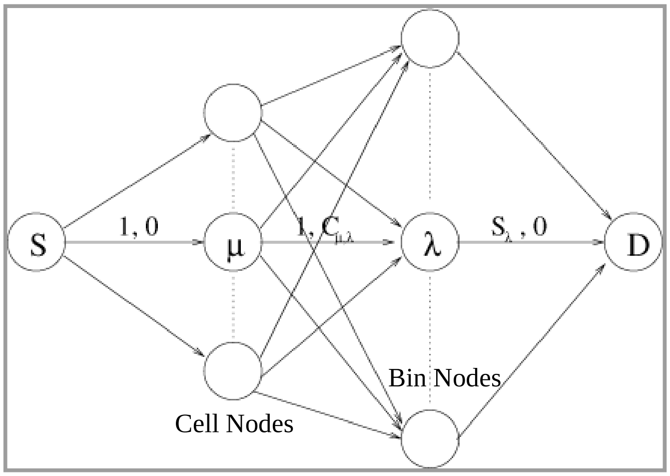
Post Processing Heuristics

- Greedy cell-centric algorithm: Greedily move cells around and greedily accept moves.
- Flow-based cell-centric algorithm: Use a flow-based approach to move cells.
- Net-centric algorithm: Move nets with bigger contributions to the congestion first.

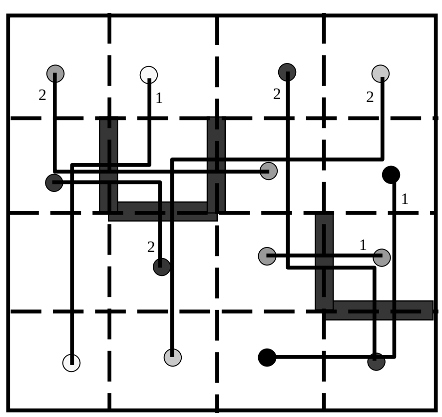
Greedy Cell-centric Heuristic



Flow-based Cell-centric Heuristic



Net-centric Heuristic

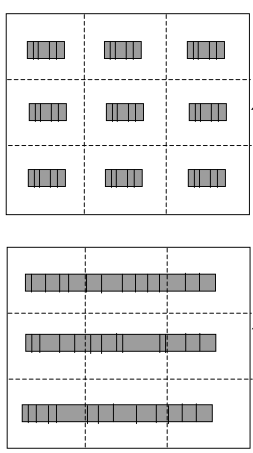


Post Processing Results

TestCase	before PP	cell-centric	flow-based	net-centric	% Imp. net-centric vs. before PP
highway2	12	7	7	7	41.7%
fract	16	14	14	14	12.5%
Primary1	34	9	17	4	88.2%
Primary2	151	56	65	49	67.5%
struct	88	52	39	47	46.5%
biomed	3011	2646	*	2610	12.1%
avqs	159	124	*	116	27.0%
avql	802	753	*	747	6.9 %
ave.					36.9%

(* out of memory)

From Global Placement to Detailed Placement



Global Placement: Assuming all the cells are placed at the centers of global bins.

Detailed Placement: Cells are placed without overlapping.

Correlation Between Global and Detailed Placement

TestCase	WL_g	CON_g	WL_d	CON_d
highway2	12	8	18	13
fract	16	14	24	23
Primary1	140	125	151	141
Primary2	710	586	917	867
struct	150	110	261	227
biomed	667	1115	605	1084
avqs	180	149	258	214
avql	898	791	1032	909

- **WL_g: Wirelength optimized global placement.**
- **CON_g: Wirelength optimized detailed placement.**
- **WL_d: Congestion optimized global placement.**
- **CON_d: Congestion optimized detailed placement.**

Conclusion: Congestion at detailed placement level is correlated with congestion at global placement level. Thus reducing congestion in global placement helps reduce congestion in final detailed placement.

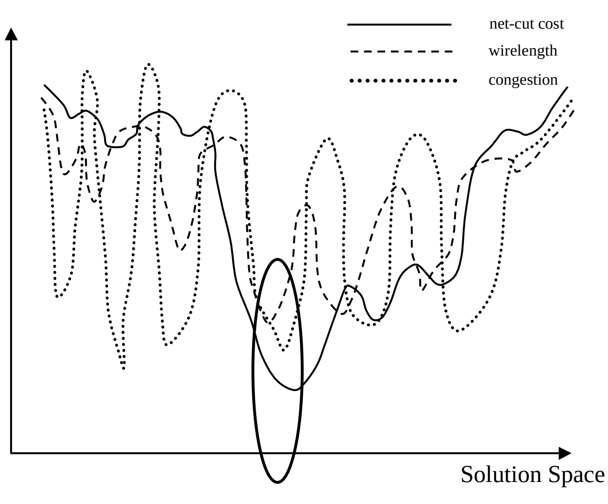
Conclusion

- Wirelength minimization can minimize congestion globally. A post processing congestion minimization following wirelength minimization works the best to reduce congestion in placement.
- We tested a number of congestion-related cost functions including a hybrid length plus congestion (commonly believed to be very effective). Experiments prove that they do not work very well.
- Net-centric post processing techniques are very effective to minimize congestion.
- Congestion at the global placement level, correlates well with congestion of detailed placement.

Summary: Relationships Between the Three Cost Functions

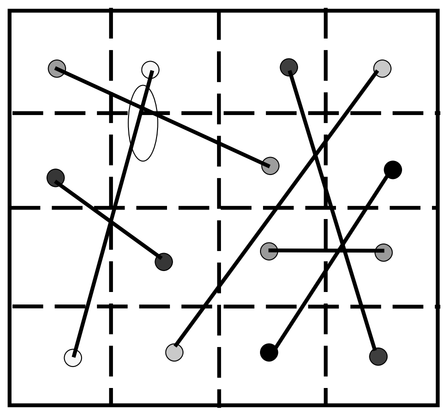
- ☞The net-cut objective function is more smooth than the wirelength objective function
- ☞The wirelength objective function is more smooth than the congestion objective function
- ☞Local minimas of these three objectives are in the same neighborhood.

Shapes of Cost Functions

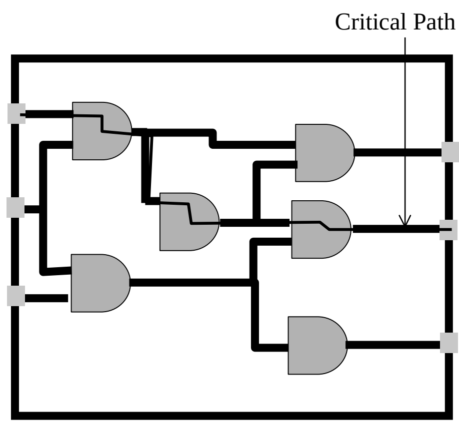


Crossing: A routability estimator?

- Replace each crossing with a "gate"
- A planar netlist
- Easy to place



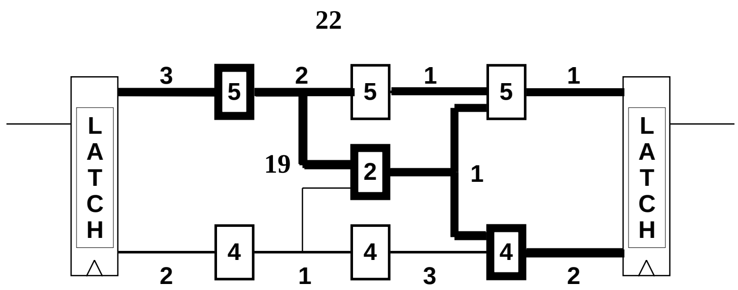
Timing Cost



☞ Delay of the circuit is defined as the longest delay among all possible paths from primary inputs to primary outputs.

☞ Interconnection delay becomes more and more important in deep sub-micron regime.

Timing Analysis



How do we get the delay numbers on the gate/interconnect?

151 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

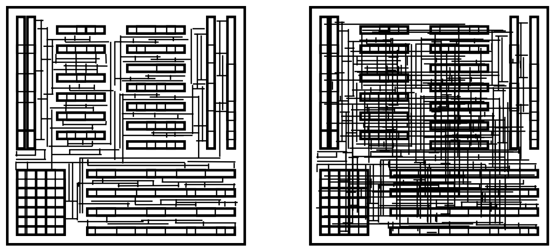
Approaches

- Budgeting
  - In accurate information
  - Fast
- Path Analysis
  - Most accurate information
  - Very slow
- Path analysis with infrequent path substitution
  - Somewhere in between

152 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

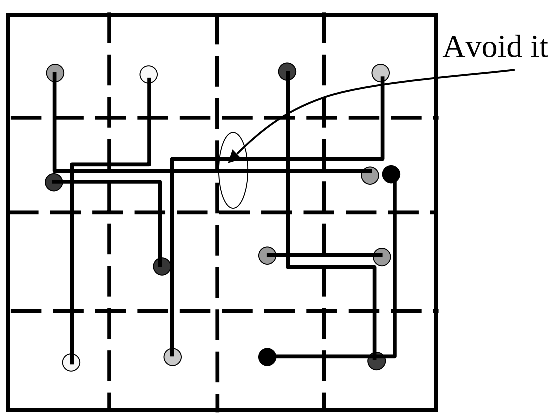
Timing Metrics

- How do we assess the change in a delay due to a potential move during physical design?
- Whether it is channel routing or area routing, the problem is the same
  - translate geometrical change into delay change



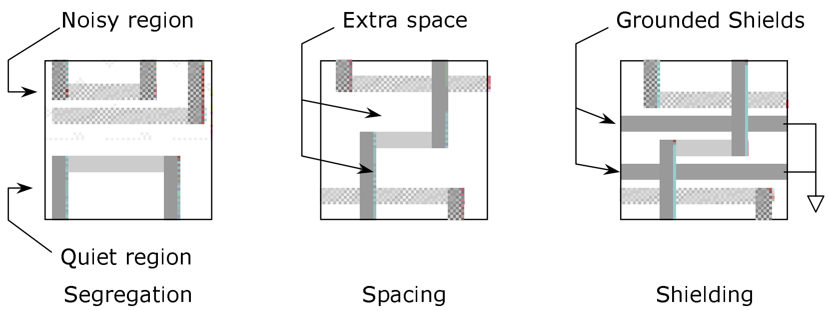
Others costs: Coupling Cost

- Hard to model during placement
- Can run a global router in the middle of placement
- Even at the global routing level it is hard to model it



Coupling Solutions

- Once we have some metrics for coupling, we can calculate sensitivities, and optimize the physical design...

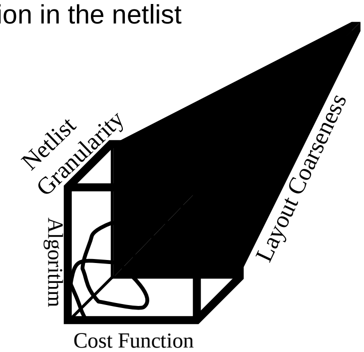


Other Performance Costs

- ☞Power usage of the chip.
  - ☞Weighted nets
  - ☞Dual voltages (severe constraint on placement)
- ☞Very little known about these cost functions and their interaction with other cost functions
- ☞Fundamental research is needed to shed some light on the structure of them

Netlist Granularity:
Problem Size and Solution Space Size

- The most challenging part of the placement problem is to solve a huge system within given amount of time
- We need to effectively reduce the size of the solution space and/or reduce the problem size
  - Netlist clustering: Edge extraction in the netlist



157 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

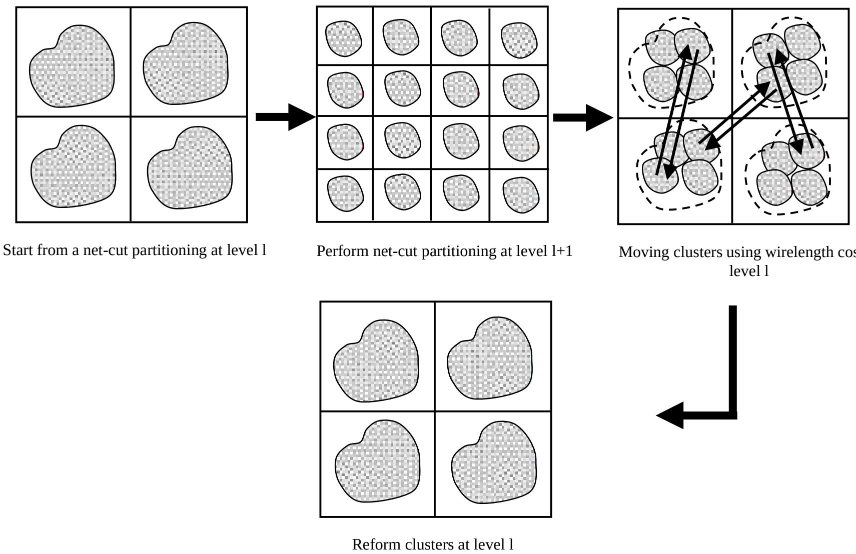
Clustering (net-cut vs. wirelength)

- Big clusters should be formed based on net-cut cost
- Small clusters should take wirelength into account.
- According to the target size of the clusters, we should be able to choose the appropriate cost function for clustering

- Reuse a partitioner:

158 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

+1 level clustering (net-cut and wirelength)



159 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

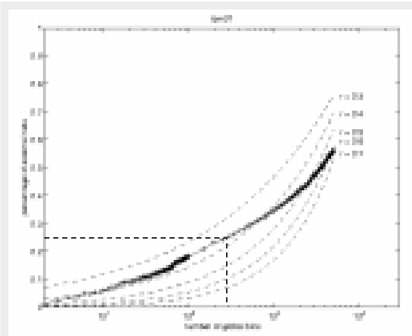
+1 Level Clustering Heuristics

- ☞+1 level A: Use hMetis to get the net-cut optimized cell clusters at level h+1. Then perform the wirelength optimization at level h: Flat hMetis
- ☞+1 level B: Use hMetis to get the net-cut optimized placement at level h. Then use hMetis to partition the subcircuit in each global bin into clusters. Then perform the wirelength optimization at level h: hierarchical (two-level) hMetis
- ☞+1 level C: Use hMetis to get the net-cut optimized placement at the first level h_1. Then use hMetis to keep partitioning until we reach level h+1. Then do clustering at level h+1 and perform the wirelength optimization back at level h. hierarchical (multi) hMetis

160 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

technique	2 × 2		4 × 4		8 × 8		16 × 16		32 × 32	
	WL	runtime	*WL*	runtime	*WL*	runtime	*WL*	runtime	*WL*	runtime
WL.fast	1472	501	1517	542	2125	511	3588	490	3505	542
WL.slow	629	15709	703	16535	972	3834	879	14962	924	14846
cut opt.	384	499	596	542	847	668	1082	961	1339	1458
+1level A	409	262	737	244	1047	300	1313	384	1739	561
+1level B	385	281	649	244	876	282	1051	326	1204	524
+1level C	384	254	790	251	858	267	1028	323	1436	463

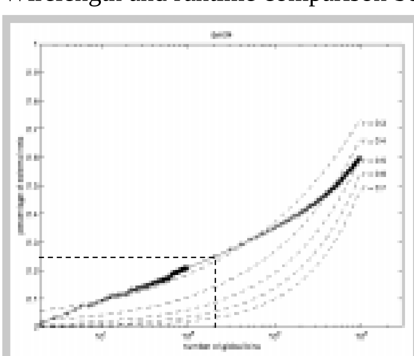
Wirelength and runtime comparison between different approaches for ibm01



Percentage of external nets vs. number of global bins: cut-only is good early-on, cut+WL later

technique	2 × 2		4 × 4		8 × 8		16 × 16		32 × 32	
	WL	runtime	*WL*	runtime	*WL*	runtime	*WL*	runtime	*WL*	runtime
WL.fast	8635	570	11537	2433	12917	1460	15138	1065	15103	1080
WL.slow	6418	3921	8139	9088	6946	9844	7008	23068	7292	17156
cut opt.	5749	1377	6730	1409	7190	1658	7670	2552	8297	2276
+1level A	5816	884	6806	751	7731	824	7976	986	8820	1365
+1level B	5762	886	6714	700	7241	766	7436	846	7939	1123
+1level C	5789	882	6979	801	7437	754	7752	790	9543	1036

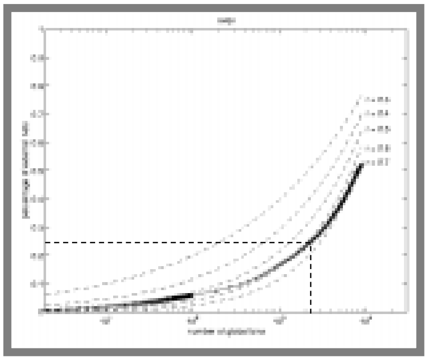
Wirelength and runtime comparison between different approaches for ibm04



Percentage of external nets vs. number of global bins

technique	2 × 2 WL	2 × 2 runtime	4 × 4 WL	4 × 4 runtime	8 × 8 WL	8 × 8 runtime	16 × 16 WL	16 × 16 runtime	32 × 32 WL	32 × 32 runtime
WL.fast	1325	814	1472	716	1452	742	1472	724	1581	654
WL.slow	1077	1672	1173	1739	1064	1579	1350	1712	905	3241
cut opt.	226	355	319	363	409	411	496	424	656	533
+1level A	272	219	486	186	763	207	1017	260	1456	280
+1level B	310	217	406	179	516	196	656	215	765	262
+1level C	253	212	405	188	495	179	699	227	778	265

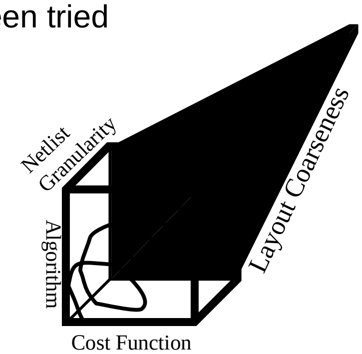
Wirelength and runtime comparison between different approaches for avqs



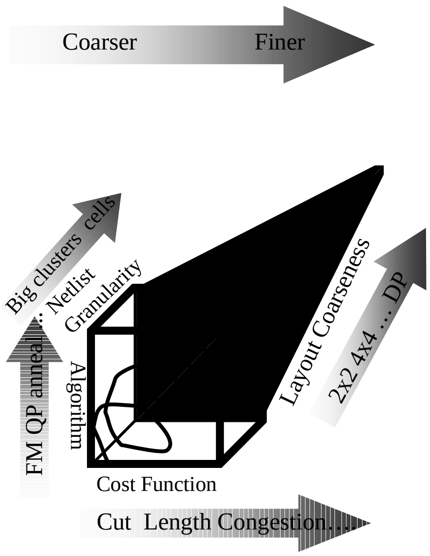
Percentage of external nets vs. number of global bins

Layout Coarsening

- Reduce Solution Space
- Edge extraction in the solution space
- Only simple things have been tried
  - GP, DP (Twolf)
  - 2x1, 2x2, ….
- Coarsen only “easy” parts



Summary of the Placement Cube



165 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

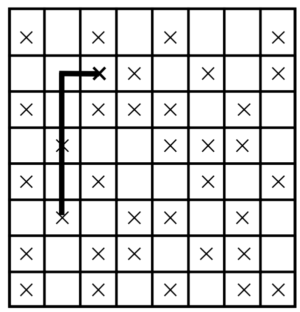
Incremental Placement

- ☞Given an optimal placement for a given netlist, how to construct optimal placements for netlists modified from the given netlist.
- ☞Very little research in this area.
  - ☞Different type of incremental changes (in one region, or, all over)
  - ☞Methods to use
  - ☞How global should the method be
- ☞An extremely important problem.

166 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

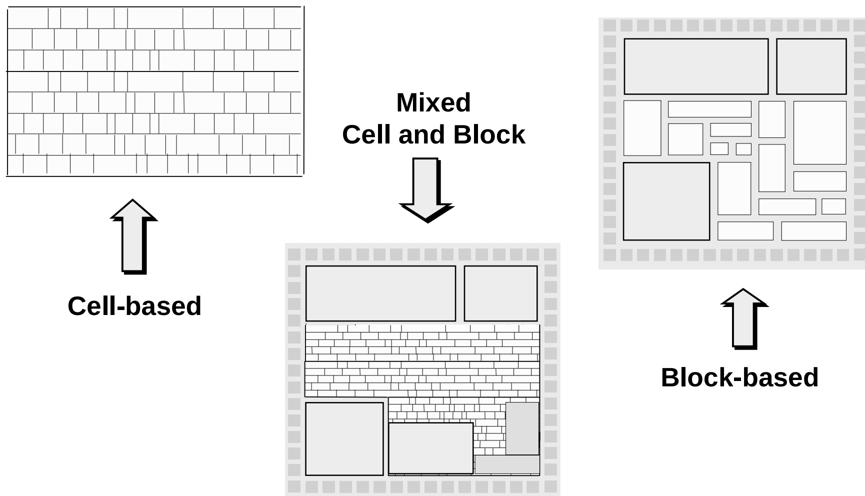
Incremental Placement

- A placement move changes the interconnect capacitance and resistance of the associated net
- A net topology approximation is required to estimate these changes

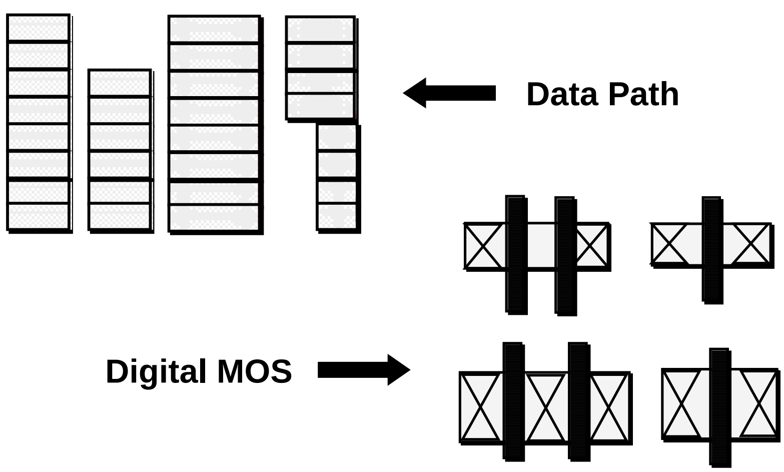


Routing Algorithms

Routing Applications



Routing Applications



ASIC Requirements

- Automation
- Fast turn around time
  - OK to trade some performance and die size

Structured Custom Requirements

- Circuit or chip performance
- User control with fine-grain rules
- Interactively complete or modify routing
- Potential modeling issues in library data

Algorithms

- Global routing
  - Guide the detail router in large design
  - May perform quick initial detail routing
  - Commonly used in cell-based design, chip assembly, and datapath
  - Also used in floorplanning and placement
- Detail routing
  - Connect all pins in each net
  - Must understand most or all design rules
  - May use a compactor to optimize result
  - Necessary in all applications

173 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Global Routing Objectives

- Minimize wire length
- Balance congestion
- Timing driven
- Noise driven
- Keep bus together

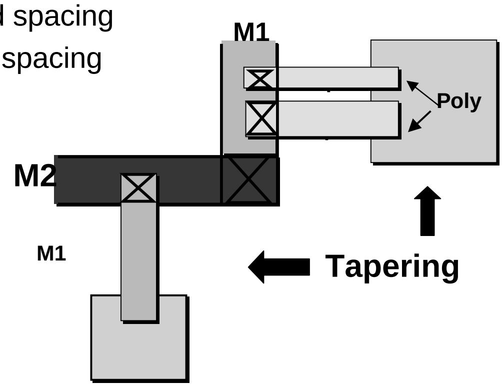
174 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Global Routing Algorithms

- Steiner tree
- Channel based
- Maze routing

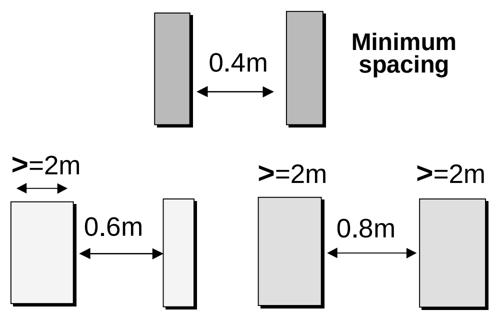
Detail Routing Objectives

- Routing completion
- Width and spacing rule
  - Minimum width and spacing
  - Variable width and spacing
    - Connection
    - Net
    - Class of nets
  - Tapering



Detail Routing Objectives

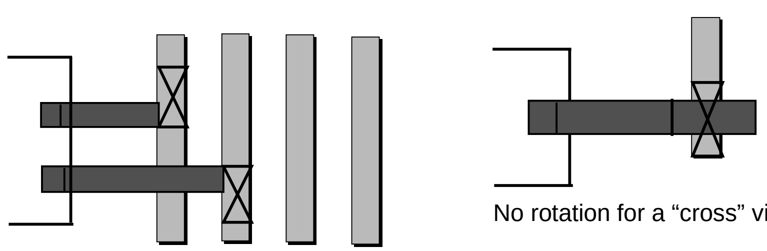
- Width and spacing rule



Width-based Spacing

Detail Routing Objectives

- Via selection
  - Via array based on wire size or resistance
  - Rectangular via rotation and offset

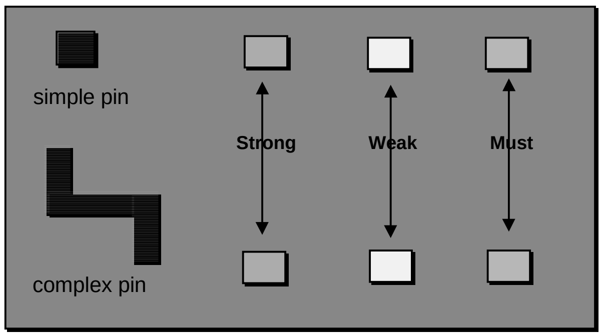


Rotate and offset horizontal vias

No rotation for a "cross" via

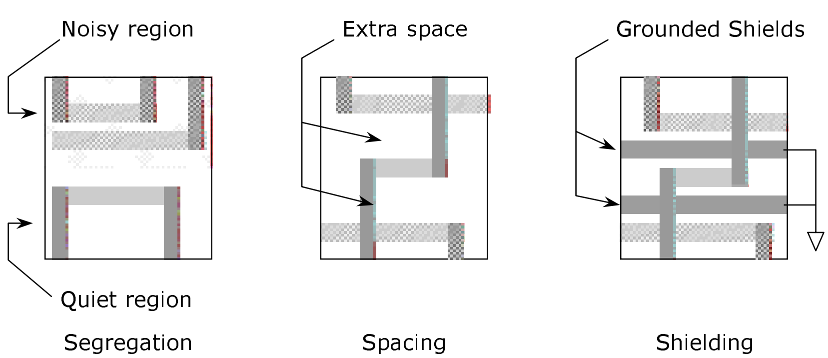
Detail Routing Objectives

- Understand complex pin & equivalent pin modeling



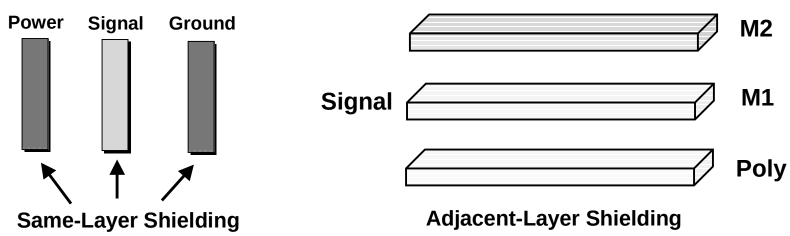
Detail Routing Objectives

- Noise-driven



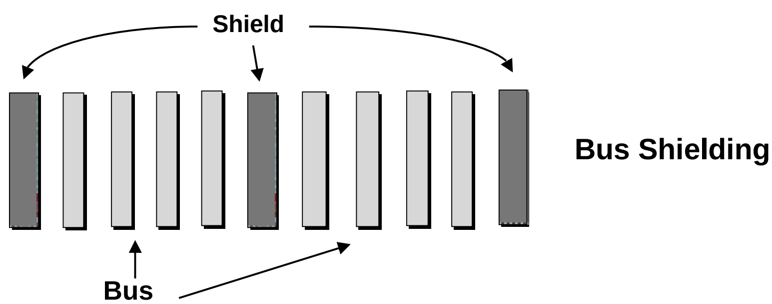
Detail Routing Objective

- Shielding
  - Same-layer shielding
  - Adjacent-layer shielding



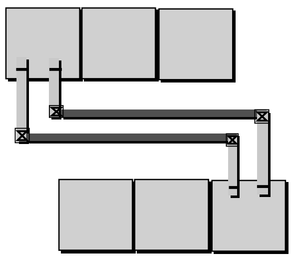
Detail Routing Objective

- Shielding
  - Bus shielding
  - Bus interleaving

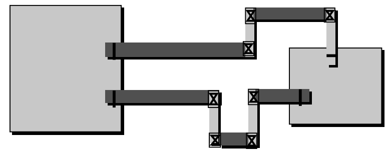


Detail Routing Objectives

- Differential pair routing
- Balanced length or capacitance



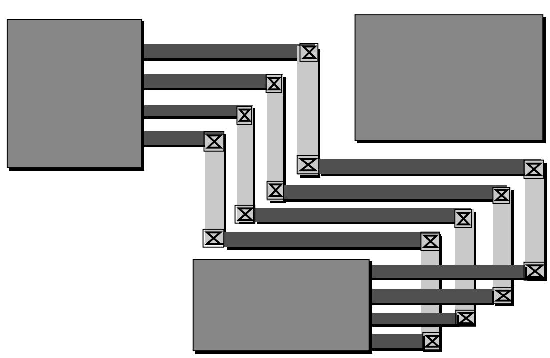
Differential



Balanced length

Detail Routing Objectives

- **Bus Routing**



Detail Routing Objectives

- Process antenna rule
- Phase shift mask

185 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

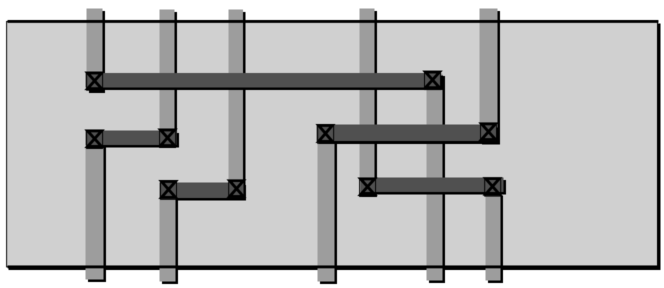
Detail Routing Algorithms

- Channel routing
- Switch box routing
- Maze routing
- Line probe routing
- Shape-based routing
- Fixed die Vs variable die
- Gridded Vs gridless

186 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

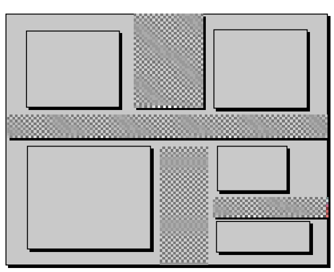
Channel Routing

- Share tracks to reduce channel height
- Resolve vertical constraints
- Left edge, dog leg
- Suitable for channeled design
- Good for 2 or 3 routing layers

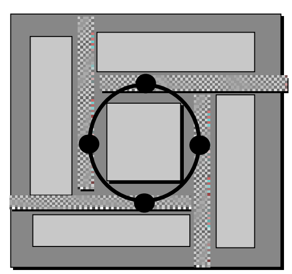


Channel Definition

- Form channels among blocks or between rows
- Route channels in order
- Require sliceable design



Sliceable Channel

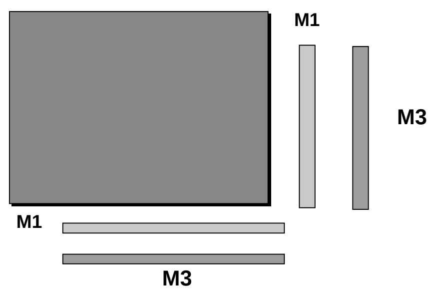


Loop

Channel Definition

- Good for 2 and 3 layer design
- Changing layer direction in 3-layer design improves die size
- Suitable when most routing is performed between blocks

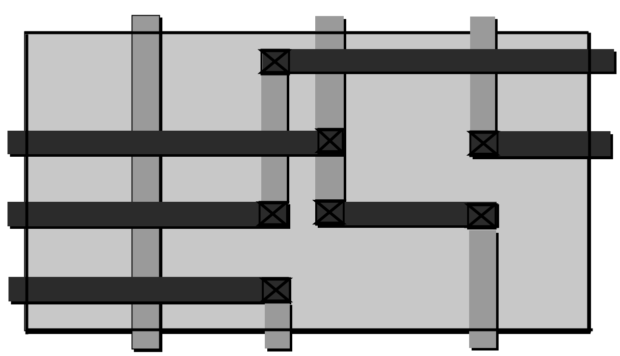
Changing Layer Direction reduces channel height



189 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Switch Box Routing

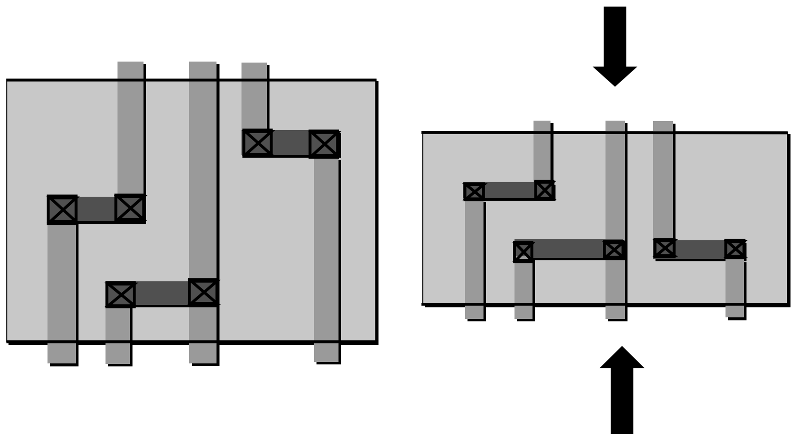
- Used in channel junction
- More constrained than channel routing
- Difficult with 3 or more layers



190 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

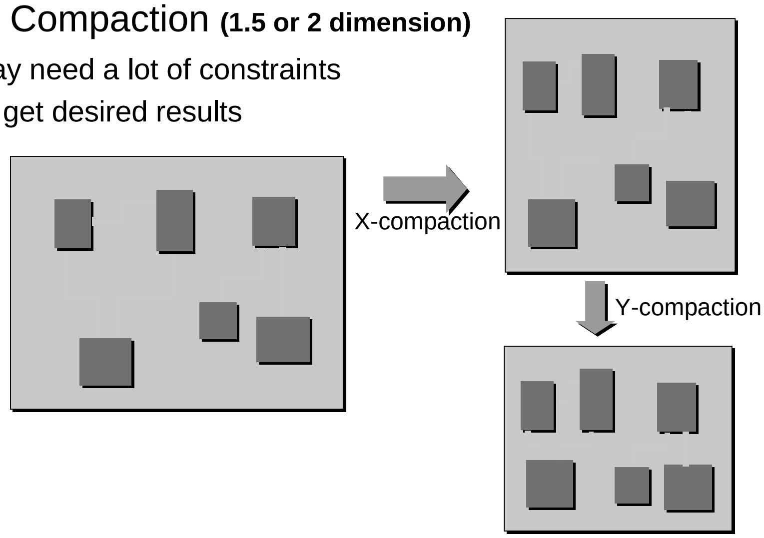
Compaction

- **Channel Compaction** (one dimension)



Compaction

- Area Compaction **(1.5 or 2 dimension)**
  - May need a lot of constraints to get desired results



Maze Routing

- Point by point routing of nets
- Route from source to sink
- Objective is to route all nets according to some cost function
- Most often, cost function tries to minimize congestion, route length, coupling, etc

Maze routing algorithm

- Initialize priority queue Q, source S and sink T
- Place S in Q
- Get the lowest cost point X from Q, put neighbors of X in Q
- Repeat last step until lowest cost point X is equal to the sink T
- Rip and reroute nets

Rip and Reroute

- After all nets are routed, rip and reroute will select a number of nets based on a cost function to reroute
- Our maze router focuses on minimizing congestion, therefore the rip and reroute finds nets that are routed through congested areas, removes net's routing and reroutes the net
- Rip and reroute is very important. It greatly improves the solution

Overflow

- The main objective of our maze router is to reduce overflow

Overflow Definition

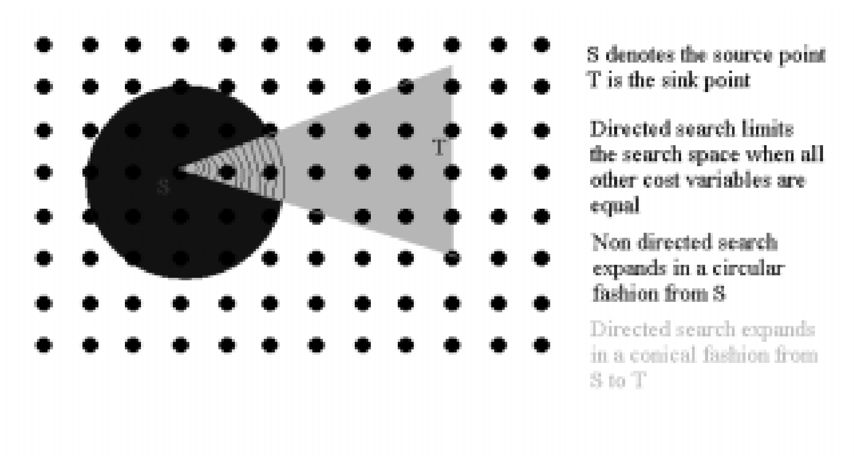
- Edge overflow = 0; if num_nets less than or equal to the capacity
- Edge overflow = num_nets – capacity; if num_nets is greater than capacity
- Overflow = Σ (edge overflows) over all edges

Maze routing cost function

- Points can be popped from queue according to a multivariable cost function
- cost = function(overflow,coupling,wire length, …)
- By adding a distance to sink variable to the cost function, you get a directed search
- Directed search allows the maze router to explore the points around the direct path from source to sink first

197 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Directed search



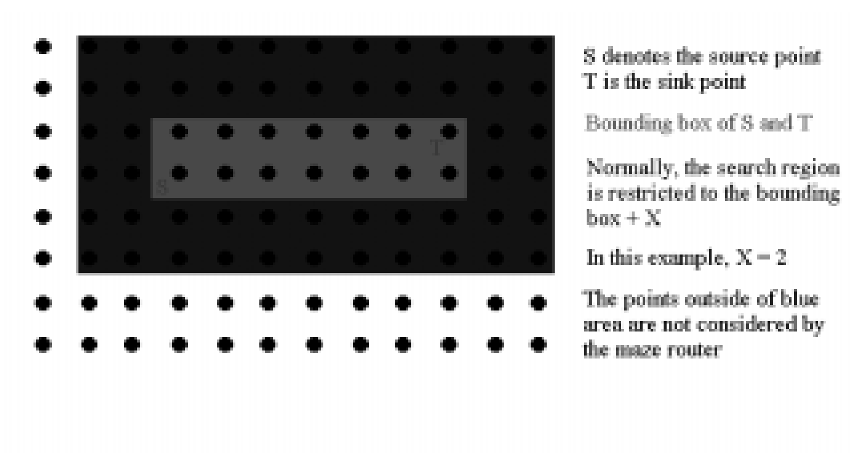
198 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Limiting the search region

- Since the majority of nets are routed within the bounding box defined by S and T, you can limit the number of points that the maze router will search to those within the bounding box
- This allows the maze router to finish sooner with little to no negative impact on the final routing cost
- Intuitively, you can see how this will decrease the runtime since the router will not consider points which are not likely to be on the route path. As stated before, any point outside the bounding box is unlikely to appear on the routing path

199 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

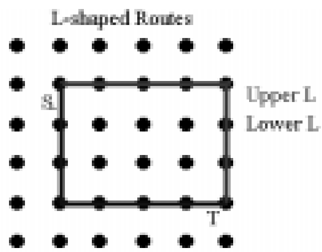
Restricting the search region

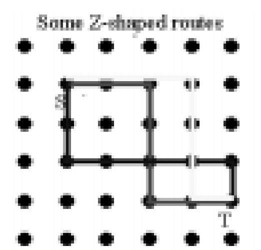


200 ICCAD Tutorial: November 9, 2000 SC/ABK/MS

Routing patterns

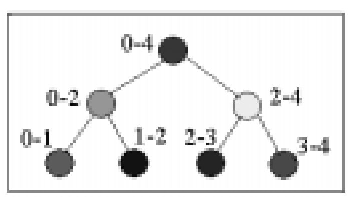
- Idea: restrict the routing of a net to certain basic templates
- Basic templates are l-shaped (1 bend) or z-shaped (2 bends) routes between a source and sink
- Templates allow fast routing of nets since you only consider certain edges and points

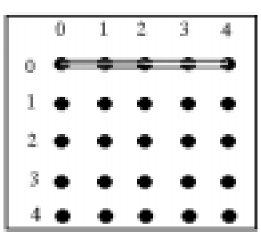




Segment Trees

- Store routing segments in binary trees for fast of segments and congestion
- Routing area is divided into m horizontal trees and n vertical segment trees where m and n are the width and height, respectively, of the routing area





The above segment tree represents the horizontal track 0

Example: red node denotes a segment from (0,0) to (4,0)

Segment Trees

- The number of segments of any node can be retrieved in log n time where n is the length of the routing track
- Segment trees give you a quick global view at the routing of the nets
- Allows to you route long z or l-shaped nets much faster than traditional grid approach
- On a 100x100 grid, segment trees will route an l-shaped net faster if the net has a bounding box perimeter greater than 40
- Therefore, we want to route long nets with segment trees if it yields a cost similar to that of traditional maze routing

Experimental Results

- In our experiments, we tried using these templates for routing the n
- Our results showed that you can route x% of the small nets in l-shaped fashion without los congestion costs
- Unfortunetly, you can not route the largest nets using l-shaped routing without a dram increase in congestion (as compared to the congestion when you maze route every n Therefore, segment trees provide no advantage for l-shaped routi
- Below, the overflow results when the x% largest nets are locked and l-shape rout

overflow costs - maze routing with x% 1-bend routing large nets -> small nets. Split into 2 terminals by mst							
filename	0%	5%	increase	5% ratio	10%	increase	10% ratio
p1.3	383	409	26	1.0678851	418	35	1.091383812
p2.3	675	874	199	1.2948148	947	272	1.402962963
avqs.2	3148	3870	722	1.229352	4150	1002	1.318297332
biomed.2	22	148	126	6.7272727	236	214	10.72727273
p1.2	70	104	34	1.4857143	105	35	1.5
p2.2	117	196	79	1.6752137	247	130	2.111111111
struct	233	280	47	1.2017167	331	98	1.420600858
biomed	3024	3307	283	1.0935847	3309	285	1.094246032
avqs	115	264	149	2.2956522	313	198	2.72173913
total	7787	9452	1665	n/a	10056	2269	n/a

Results

- The overflow results when the smallest x% (determined by bounding box) of the nets are l-shape routed. The remaining (1-x)% nets are maze routed. The smallest x% nets are locked i.e. they are not available for rip and reroute.
- The total time to route is reduced since l-shaped routing completes faster than maze routing and the locked nets allow rip and reroute to finish quickly
- When x% = 0, this is pure maze routing. When x% = 100, every net is l-shape routed
- An intuitive explanation: Since most of the routing of a net is done within it's bounding box, the nets with a small bounding box have little freedom with their routing. Thus, one of the l-shaped routes will be a good solution. On the other hand, a large bounding box gives more freedom to find a low cost path and the restrictive l-shaped route only considers two paths. Therefore, the l-shaped route is likely a bad routing choice.

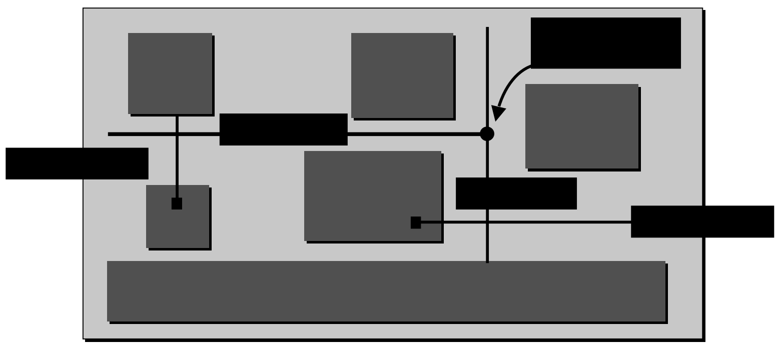
filename/x%	0	5	10	15	20	30	50	80	90	100
p1.3	379	379	379	379	379	376	376	380	378	460
p2.3	665	665	665	665	665	665	665	686	712	1099
avqs.2	2960	2960	3033	3033	3012	3016	2926	3170	2907	5224
biomed.2	15	15	15	15	15	15	15	15	16	644
p1.2	71	71	71	71	71	71	74	74	71	154
p2.2	109	109	109	109	109	109	109	115	108	315
struct	210	210	210	212	212	213	227	309	408	618
biomed	2994	2994	2994	2994	2994	2994	2985	2965	2951	3454

Coupling

- As fabrication sizes get smaller, coupling plays a larger role in timing
- Therefore, we want to minimize the number of long nets that are close to each other (on same route track)
- Segment trees keep this information
- Ways of using segment trees to reduce coupling during global routing?

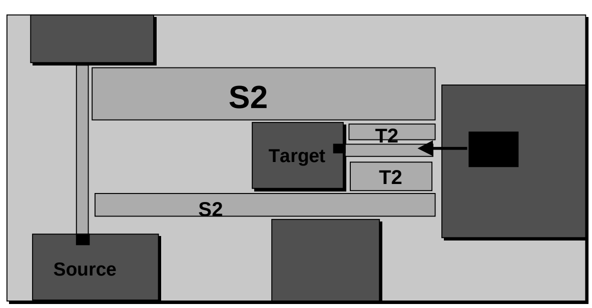
Line Probe Routing

- **Fast**
- **Handles large designs**
- **Routing may be incomplete**



Shape-based Routing

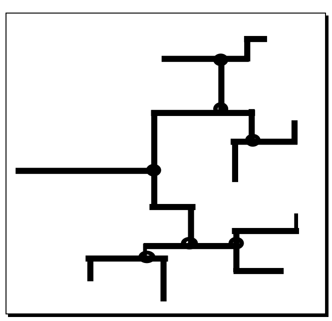
- Evolve from maze routing
- Gridless: look at actual size of each shape
- Each shape may have its spacing rule
- Good for designs with multiple width/spacing rules and other complex rules
- Slower than gridded router



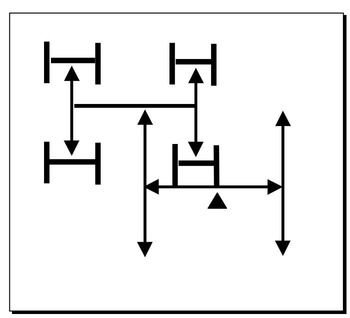
Incremental Routing

- Re-route with minor local adjustment
- Need rip-up and reroute capability
- Difficult to confine perturbation when compactor is used

Clock Routing



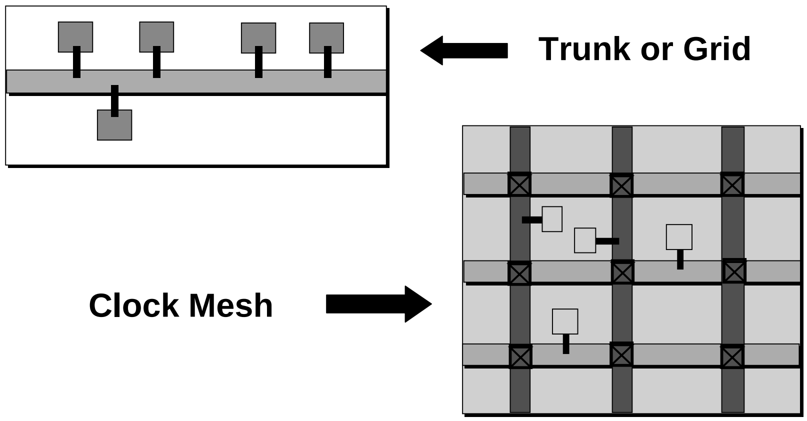
Balanced Tree



H-Tree

Clock Routing

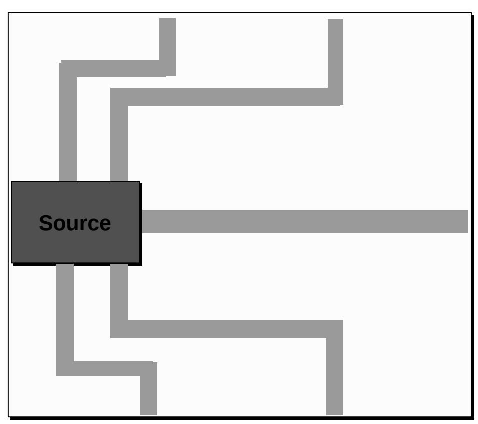
- Multiple Clock Domains



Power Routing

- Power Mesh
- Power Ring
- Star Routing

Star Routing

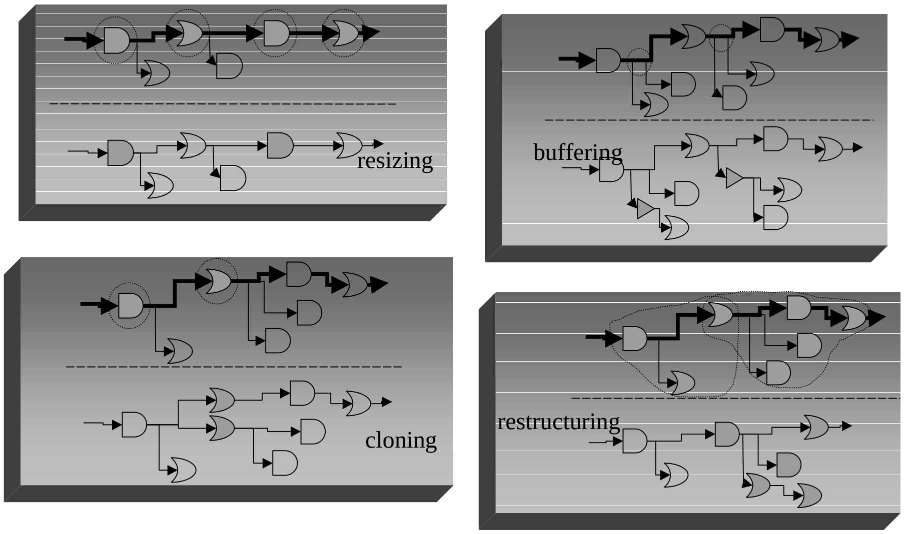


Conclusion

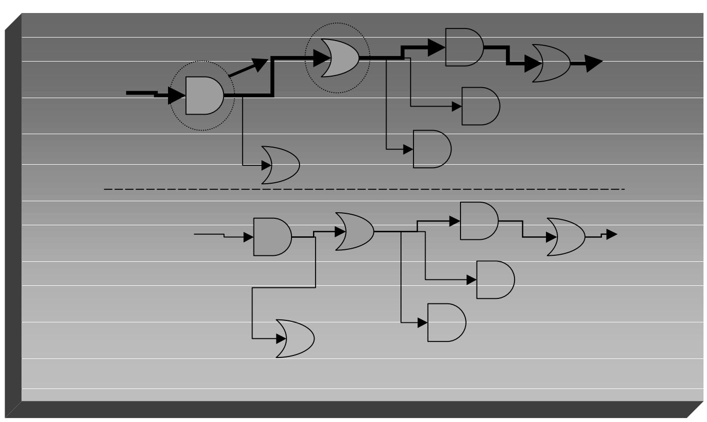
- Various routing algorithms for different applications
- Generally maze routing algorithms and derivatives are good for handling complex requirements
- Growing chip capacity and ever-changing process technology are major challenges to the router

Placynthesis Algorithms

Some Placynthesis Moves

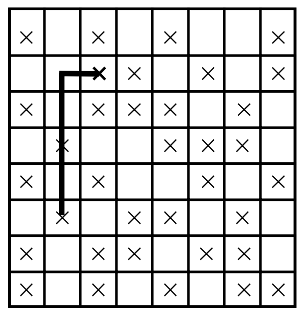


More Placynthesis Moves

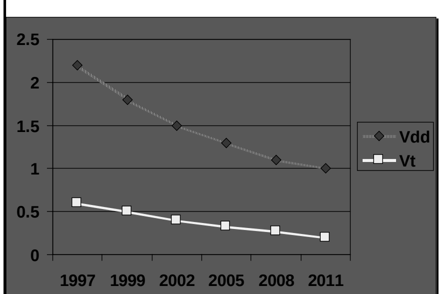


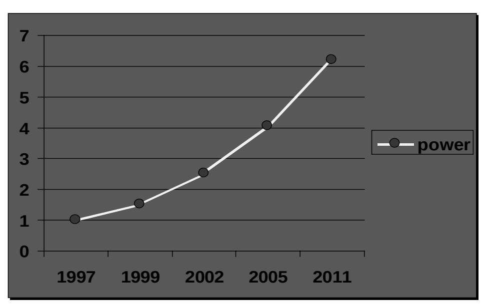
Iterative Placement

- A placement move changes the interconnect capacitance and resistance of the associated net
- A net topology approximation is required to estimate these changes



Many other Design Metrics: Power Supply and Total Power

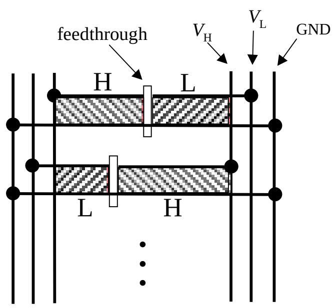




Source: The Incredible Shrinking Transistor, Yuan Taur, T. J. Watson Research Center, IBM, IEEE Spectrum, July 1999

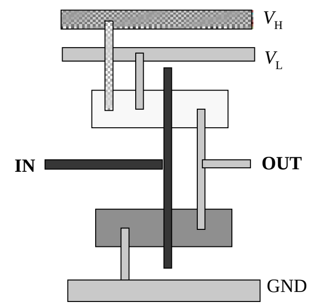
Dual Voltages: A harder problem

- Layout synthesis with dual voltages: major geometric constraints



H -- High Voltage Block
L -- Low Voltage Block

Layout Structure



Cell Library with Dual Power Rails

Conclusion

- Deep Sub-micron (DSM) problems are here and are real
- There are so many problems that we do not understand.
- Innovation (in algorithms, methodology, tools, etc) needed in all facets.