ESE535:
Electronic Design Automation

Day 10: February 18, 2009

Partitioning 2

(spectral, network flow)

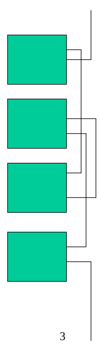
Today

- Alternate views of partitioning
- Two things we can solve optimally
  - (but don't exactly solve our original problem)
- Techniques
  - Linear Placement w/ squared wire lengths
  - Network flow MinCut

Optimization Target

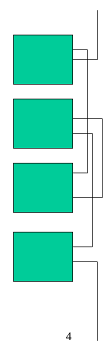
- Place cells
- In linear arrangement
- Wire length between connected cells:
  - distance=X_i - X_j
  - cost is sum of distance squared

Pick X_i's to minimize cost



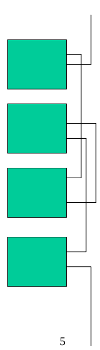
Why this Target?

- Minimize sum of squared wire distances
- Prefer:
  - **Area:** minimize channel width
  - **Delay:** minimize critical path length



Why this Target?

- Our preferred targets are discontinuous and discrete
- Cannot formulate analytically
- Not clear how to drive toward solution
  - Does reducing the channel width at a non-bottleneck help or not?
  - Does reducing a non-critical path help or not?



Spectral Ordering

Minimize Squared Wire length -- 1D layout

- Start with connection array C ($c_{i,j}$)
- "Placement" Vector X for x_i placement
- **Problem:**
  - Minimize cost = $0.5*\Sigma$ (all i,j) $(x_i - x_j)^2 c_{i,j}$
  - cost sum is X^TBX
    - B = D-C
    - D=diagonal matrix, $d_{i,i} = \Sigma$(over j) $c_{i,j}$

Preclass Netlist

- Squared wire lengths:

$$
\begin{aligned}
&(X_A - X_G)^2 \\
+&(X_B - X_G)^2 \\
+&(X_B - X_H)^2 \\
+&(X_C - X_H)^2 \\
+&(X_G - X_O)^2 \\
+&(X_H - X_O)^2
\end{aligned}
$$

C Matrix

	A	B	C	G	H	O
A				1		
B				1	1	
C					1	
G	1	1				1
H		1	1			1
O				1	1	

D Matrix

	A	B	C	G	H	O
A	1			1		
B		2		1	1	
C			1		1	
G	1	1		3		1
H		1	1		3	1
O				1	1	2

B=D-C Matrix

	A	B	C	G	H	O
A	1			-1		
B		2		-1	-1	
C			1		-1	
G	-1	-1		3		-1
H		-1	-1		3	-1
O				-1	-1	2

BX

$$
\begin{array}{c|cccccc|}
 & A & B & C & G & H & O \\
\hline
A & 1 & & & -1 & & \\
B & & 2 & & -1 & -1 & \\
C & & & 1 & & -1 & \\
G & -1 & -1 & & 3 & & -1 \\
H & & -1 & -1 & & 3 & -1 \\
O & & & & -1 & -1 & 2
\end{array}
\begin{bmatrix} X_A \\ X_B \\ X_C \\ X_G \\ X_H \\ X_O \end{bmatrix}
=
\begin{bmatrix} X_A - X_G \\ 2X_B - X_G - X_H \\ X_C - X_H \\ 3X_G - X_A - X_B - X_O \\ 3X_H - X_B - X_C - X_O \\ 2X_O - X_G - X_H \end{bmatrix}
$$

$X^T(BX)$

$$
\begin{bmatrix} X_A & X_B & X_C & X_G & X_H & X_O \end{bmatrix}
\begin{bmatrix} X_A - X_G \\ 2X_B - X_G - X_H \\ X_C - X_H \\ 3X_G - X_A - X_B - X_O \\ 3X_H - X_B - X_C - X_O \\ 2X_O - X_G - X_H \end{bmatrix}
$$

$X^T(BX)$

$X_A^2 - X_AX_G$

$+2X_B^2 - X_BX_G - X_BX_H$

$+X_C^2 - X_C X_H$

$+3X_G^2 - X_AX_G - X_BX_G - X_GX_O$

$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$

$+2X_O^2 - X_GX_O - X_HX_O$

$X^T(BX)$

$X_A^2 - X_AX_G$	$(X_A - X_G)^2$
$+2X_B^2 - X_BX_G - X_BX_H$	$+2X_B^2 - X_BX_G - X_BX_H$
$+X_C^2 - X_C X_H$	$+X_C^2 - X_C X_H$
$+3X_G^2 - X_AX_G - X_BX_G - X_GX_O$	$+2X_G^2 - X_BX_G - X_GX_O$
$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$	$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$
$+2X_O^2 - X_GX_O - X_HX_O$	$+2X_O^2 - X_GX_O - X_HX_O$

$X^T(BX)$

$(X_A - X_G)^2$	$(X_A - X_G)^2 + (X_B - X_G)^2$
$+2X_B^2 - X_BX_G - X_BX_H$	$+X_B^2 - X_BX_H$
$+X_C^2 - X_C X_H$	$+X_C^2 - X_C X_H$
$+2X_G^2 - X_BX_G - X_GX_O$	$+X_G^2 - X_GX_O$
$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$	$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$
$+2X_O^2 - X_GX_O - X_HX_O$	$+2X_O^2 - X_GX_O - X_HX_O$

Can See Will Converge To..

- Squared wire lengths:
  $(X_A - X_G)^2$
  $+(X_B - X_G)^2$
  $+(X_B - X_H)^2$
  $+(X_C - X_H)^2$
  $+(X_G - X_O)^2$
  $+(X_H - X_O)^2$

$(X_A - X_G)^2 + (X_B - X_G)^2$

$+X_B^2 - X_BX_H$

$+X_C^2 - X_C X_H$

$+X_G^2 - X_GX_O$

$+3X_H^2 - X_BX_H - X_CX_H - X_HX_O$

$+2X_O^2 - X_GX_O - X_HX_O$

Trying to Minimize

- Squared wire lengths:
  $(X_A - X_G)^2$
  $+(X_B - X_G)^2$
  $+(X_B - X_H)^2$
  $+(X_C - X_H)^2$
  $+(X_G - X_O)^2$
  $+(X_H - X_O)^2$
- Which we know is also X^TBX
- Make all X_i's same?
- …but, we probably need to be in unique positions.

Spectral Ordering

- Add constraint: $X^TX=1$
  - prevent trivial solution all x_i's $=0$
- Minimize cost$=X^TBX$ w/ constraint
  - minimize $L=X^TBX-\lambda(X^TX-1)$
  - $\partial L/\partial X=2BX-2\lambda X=0$
  - $(B-\lambda I)X=0$
  - $X \rightarrow$ Eigenvector of B
  - cost is Eigenvalue λ

Spectral Solution

- Smallest eigenvalue is zero
  - Corresponds to case where all x_i's are the same ➔uninteresting
- **Second smallest** eigenvalue (eigenvector) is the solution we want

Eigenvector for B

B Matrix

	A	B	C	G	H	O
A	1			-1		
B		2		-1	-1	
C			1		-1	
G	-1	-1		3		-1
H		-1	-1		3	-1
O				-1	-1	2

X_A		0.6532815
X_B		1.1157741E-14
X_C	=	-0.6532815
X_G		0.27059805
X_H		-0.27059805
X_O		1.9342212E-14

Spectral Ordering

- X (x_i's) continuous
- use to order nodes
  - real problem wants to place at discrete locations
  - this is one case where can solve ILP from LP
    - Solve LP giving continuous x_i's
    - then move back to closest discrete point

Eigenvector for B

Order?

X_A		0.6532815
X_B		1.1157741E-14
X_C	=	-0.6532815
X_G		0.27059805
X_H		-0.27059805
X_O		1.9342212E-14

Eigenvector for B

A
G
O
B
H
C

X_A		0.6532815
X_B		1.1157741E-14
X_C	=	-0.6532815
X_G		0.27059805
X_H		-0.27059805
X_O		1.9342212E-14

Spectral Ordering Option

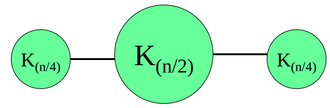
- With iteration, can reweigh connections to change cost model being optimized
  - linear
  - (distance)$^{1.X}$
- Can encourage "closeness"
  - by weighting connection between nodes
    - Making $c_{i,j}$ larger
  - (have to allow some nodes to not be close)

Spectral Partitioning

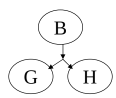
- Can form a basis for partitioning
- Attempts to cluster together connected components
- Create partition from ordering
  - *E.g.* Left half of ordering is one half, right half is the other

Spectral Ordering

- Midpoint bisect isn't necessarily best place to cut, consider:

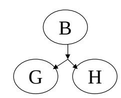


Fanout

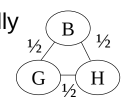


- How do we treat fanout?
- As described assumes point-to-point nets
- For partitioning, pay price when cut something once
  - *I.e.* the accounting did last time for KLFM
- Also a discrete optimization problem
  - Hard to model analytically

Spectral Fanout



- Typically:
  - Treat all nodes on a single net as fully connected
  - Model links between all of them
  - Weight connections so cutting in half counts as cutting the wire
  - Threshold out high fanout nodes
    - If connect to too many things give no information



Spectral Partitioning Options

- Can bisect by choosing midpoint
  - (not strictly optimizing for minimum bisect)
- Can relax cut critera
  - min cut w/in some δ of balance
- Ratio Cut
  - minimize (cut/|A||B|)
    - idea tradeoff imbalance for smaller cut
      - more imbalance $\rightarrow$smaller |A||B|
      - so cut must be much smaller to accept
- Circular bisect/relaxed/ratio cut
  - wrap into circle and pick two cut points
  - How many such cuts?

Spectral vs. FM

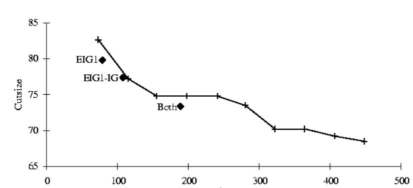
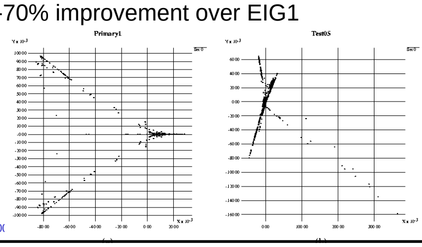


Figure 5. Graphs of cutsizes for different numbers of runs of our optimized version of KLFM versus the spectral initialization approaches. Values shown are the geometric means of the results for the 9 test circuits (all but industry3).

From Hauck/Boriello '96

Improving Spectral

- More Eigenvalues
  - look at clusters in n-d space
    - **But:** 2 eigenvectors is not opt. solution to 2D placement
  - 5--70% improvement over EIG1



Spectral Note

- Unlike KLFM, attacks **global** connectivity characteristics
- Good for finding “natural” clusters
  - hence use as clustering heuristic for multilevel algorithms
- After doing spectral
  - Can often improve incrementally using KLFM pass

Max Flow

MinCut

MinCut Goal

- Find maximum flow (mincut) between a source and a sink
  - no balance guarantee

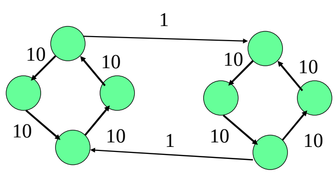
MaxFlow

- Set all edge flows to zero
  - F[u,v]=0
- While there is a path from s,t
  - (breadth-first-search)
  - for each edge in path f[u,v]=f[u,v]+1
  - f[v,u]=-f[u,v]
  - When c[v,u]=f[v,u] remove edge from search
- O(|E|*cutsize)
- [Our problem simpler than general case CLR]

Technical Details

- For min-cut in graphs,
  - Don’t really care about directionality of cut
  - Just want to minimize wire crossings
- Fanout
  - Want to charge discretely …cut or not cut
- Pick start and end nodes?

Directionality

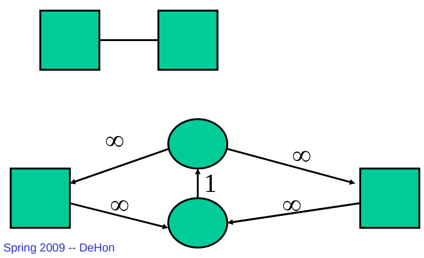


For logic net: cutting a net is the same regardless of which way the signal flows

37

Penn ESE525 Spring 2009 -- DeHon

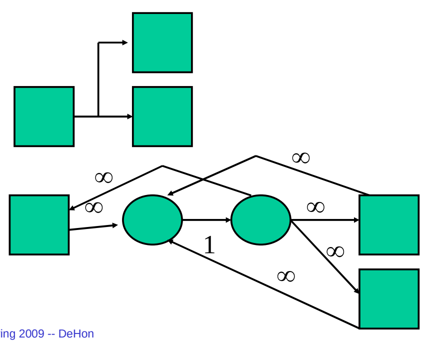
Directionality Construct



38

Penn ESE525 Spring 2009 -- DeHon

Fanout Construct



39

Penn ESE525 Spring 2009 -- DeHon

Extend to Balanced Cut

- Pick a start node and a finish node
- Compute min-cut start to finish
- If halves sufficiently balanced, done
- else
  - collapse all nodes in smaller half into one node
  - pick a node adjacent to smaller half
  - collapse that node into smaller half
  - repeat from min-cut computation

FBB -- Yang/Wong ICCAD'94

40

Penn ESE525 Spring 2009 -- DeHon

Observation

- Can use residual flow from previous cut when computing next cuts
- Consequently, work of multiple network flows is only O(|E|*final_cut_cost)

41

Penn ESE525 Spring 2009 -- DeHon

Picking Nodes

- Optimal:
  - would look at all s,t pairs
    - Just for first cut is merely N-1 “others”
      - …N/2 to guarantee something in second half
    - Anything you pick **must** be in separate halves
    - Assuming there is a perfect/ideal bisection
      - If pick randomly, probability different halves: 50%
      - Few random selections likely to yield s,t in different halves
  - would also look at all nodes to collapse into smaller
  - could formulate as branching search

42

Penn ESE525 Spring 2009 -- DeHon

Picking Nodes

- Randomly pick
  - (maybe try several starting points)
- With small number of adjacent nodes,
  - could afford to branch on all

Admin

- No class Monday 23rd
- Class on Wed. 25th
  - Assignment 2B due
  - Hardcopy handout for reading
- Assignment 3 online soon
  - Already?

Big Ideas

- Divide-and-Conquer
- Techniques
  - flow based
  - numerical/linear-programming based
  - Transformation constructs
- Exploit problems we can solve optimally
  - Mincut
  - Linear ordering