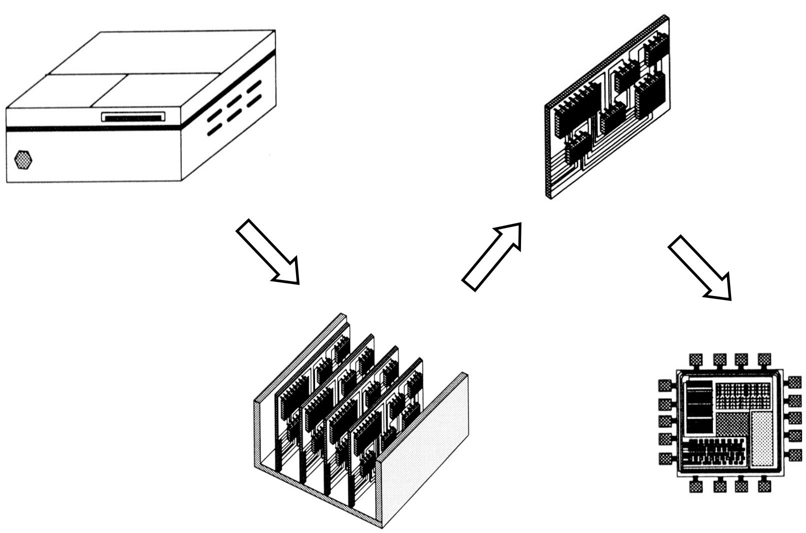
VLSI Physical Design Automation

Lecture 3. Circuit Partitioning

Prof. David Pan
dpan@ece.utexas.edu
Office: ACES 5.434

System Hierarchy

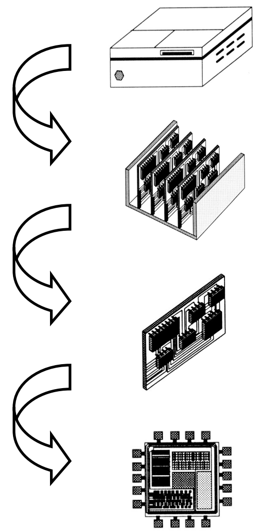


Levels of Partitioning

System Level Partitioning

Board Level Partitioning

Chip Level Partitioning



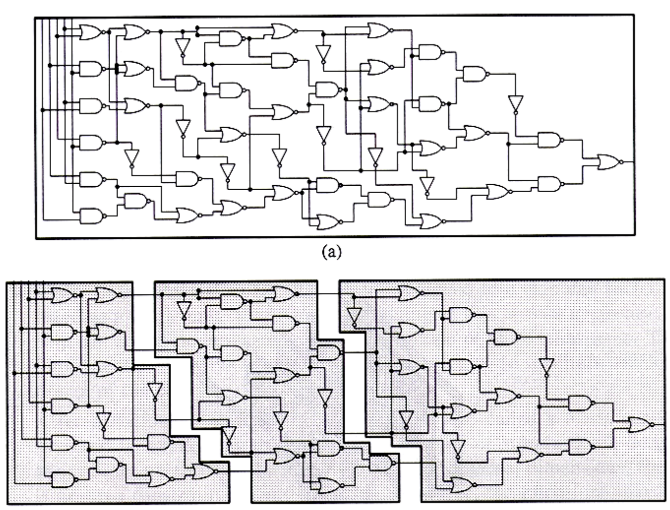
System

PCBs

Chips

Subcircuits / Blocks

Partitioning of a Circuit



(a)

(b)

Importance of Circuit Partitioning

- **Divide-and-conquer methodology**

  The most effective way to solve problems of high complexity

  E.g.: min-cut based placement, partitioning-based test generation,…
- **System-level partitioning for multi-chip designs**

  inter-chip interconnection delay dominates system performance.
- **Circuit emulation/parallel simulation**

  partition large circuit into multiple FPGAs (e.g. Quickturn), or multiple special-purpose processors (e.g. Zycad).
- **Parallel CAD development**

  Task decomposition and load balancing
- **In deep-submicron designs, partitioning defines local and global interconnect, and has significant impact on circuit performance**

…… ……

Some Terminology

Partitioning: Dividing bigger circuits into a small number of partitions (top down)

Clustering: cluster small cells into bigger clusters (bottom up).

Covering / Technology Mapping: Clustering such that each partitions (clusters) have some special structure (e.g., can be implemented by a cell in a cell library).

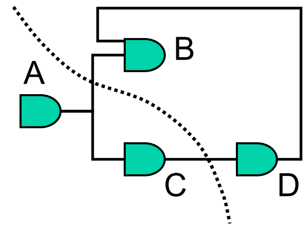
k-way Partitioning: Dividing into k partitions.

Bipartitioning: 2-way partitioning.

Bisectioning: Bipartitioning such that the two partitions have the same size.

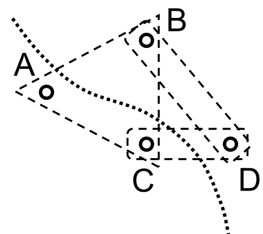
Circuit Representation

- Netlist:
  - Gates: A, B, C, D
  - Nets: {A,B,C}, {B,D}, {C,D}



- Hypergraph:
  - Vertices: A, B, C, D
  - Hyperedges: {A,B,C}, {B,D}, {C,D}

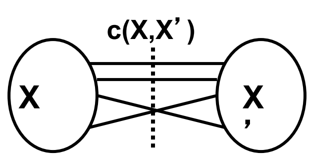
  - Vertex label: Gate size/area
  - Hyperedge label:
    Importance of net (weight)



Circuit Partitioning Formulation

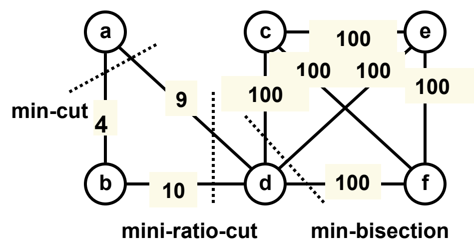
Bi-partitioning formulation:

Minimize interconnections between partitions



- **Minimum cut:** **min c(x, x')**
- **minimum bisection:** **min c(x, x') with |x|= |x' |**
- **minimum ratio-cut:** **min c(x, x') / |x||x' |**

A Bi-Partitioning Example



Min-cut size=13
Min-Bisection size = 300
Min-ratio-cut size= 19

Ratio-cut helps to identify natural clusters

Circuit Partitioning Formulation (Cont' d)

General multi-way partitioning formulation:

Partitioning a network N into N_1, N_2, …, N_k such that

- **Each partition has an area constraint**

$$\sum_{v \in N_i} a(v) \leq A_i$$

- **each partition has an I/O constraint**

$$c(N_i, N - N_i) \leq I_i$$

Minimize the total interconnection:

$$\sum_{N_i} c(N_i, N - N_i)$$

Partitioning Algorithms

- **Iterative partitioning algorithms**
- **Spectral based partitioning algorithms**
- **Net partitioning vs. module partitioning**
- **Multi-way partitioning**
- **Multi-level partitioning**
- **Further study in partitioning techniques (timing-driven …)**

Iterative Partitioning Algorithms

- **Greedy iterative improvement method**
  - **[Kernighan-Lin 1970]**
  - **[Fiduccia-Mattheyses 1982]**
  - **[krishnamurthy 1984]**
- **Simulated Annealing**
  - **[Kirkpartrick-Gelatt-Vecchi 1983]**
  - **[Greene-Supowit 1984]**

Kernighan-Lin Algorithm

"An Efficient Heuristic Procedure for Partitioning Graphs"

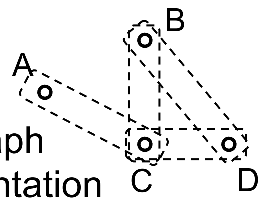
The Bell System Technical Journal

49(2):291-307, 1970

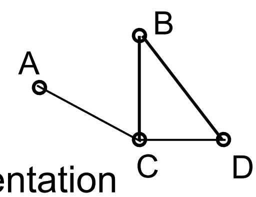
Restricted Partition Problem

- Restrictions:
  - For Bisectioning of circuit.
  - Assume all gates are of the same size.
  - Works only for 2-terminal nets.
- If all nets are 2-terminal,
  the Hypergraph is called a Graph.

Hypergraph Representation



Graph Representation



Problem Formulation

- Input: A graph with
  - Set vertices V. ($|V| = 2n$)
  - Set of edges E. ($|E| = m$)
  - Cost c_{AB} for each edge {A, B} in E.
- Output: 2 partitions X & Y such that
  - Total cost of edges cut is minimized.
  - Each partition has n vertices.

- This problem is NP-Complete!!!!!

3/31/16 15

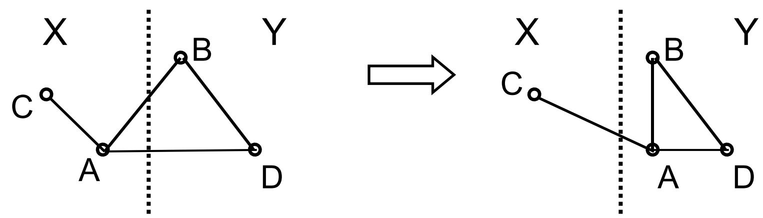
A Trivial Approach

- Try all possible bisections. Find the best one.
- If there are 2n vertices,
  \# of possibilities = $(2n)! / n!^2 = n^{O(n)}$
- For 4 vertices (A,B,C,D), 3 possibilities.
  1. X={A,B} & Y={C,D}
  2. X={A,C} & Y={B,D}
  3. X={A,D} & Y={B,C}
- For 100 vertices, 5×10^{28} possibilities.
- Need 1.59×10^{13} years if one can try 100M possbilities per second.

3/31/16 16

Idea of KL Algorithm

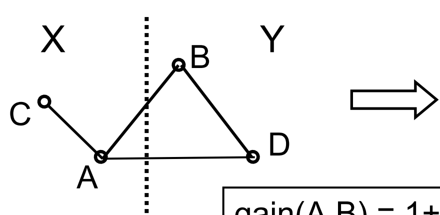
- D_A = Decrease in cut value if moving A
  - External cost (connection) E_A – Internal cost I_A
  - Moving node a from block A to block B would increase the value of the cutset by E_A and decrease it by I_A



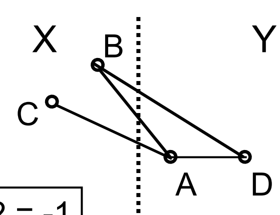
D_A = 2-1 = 1
D_B = 1-1 = 0

Idea of KL Algorithm

- Note that we want to balance two partitions
- If switch A & B, gain(A,B) = D_A+D_B-2c_{AB}
  - c_{AB} : edge cost for AB



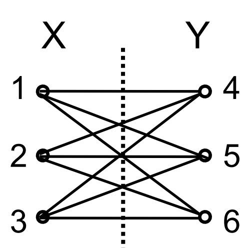
gain(A,B) = 1+0-2 = -1



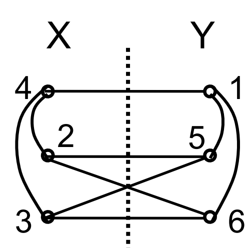
Idea of KL Algorithm

- Start with any initial legal partitions X and Y.
- A pass (exchanging each vertex exactly once) is described below:
  1. For i := 1 to n do
     From the unlocked (unexchanged) vertices,
     choose a pair (A,B) s.t. gain(A,B) is largest.
     Exchange A and B. Lock A and B.
     Let g_i = gain(A,B).
  2. Find the k s.t. $G=g_1+...+g_k$ is maximized.
  3. Switch the first k pairs.
- Repeat the pass until there is no improvement (G=0).

Example



Original Cut Value = 9



Optimal Cut Value = 5

A good step-by-step example in SY book

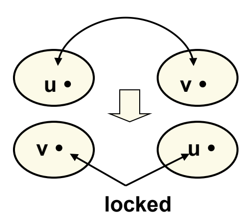
Time Complexity of KL

- For each pass,
  - $O(n^2)$ time to find the best pair to exchange.
  - n pairs exchanged.
  - Total time is $O(n^3)$ per pass.
- Better implementation can get $O(n^2 \log n)$ time per pass.
- Number of passes is usually small.

Recap of Kernighan-Lin's Algorithm

- **Pair-wise exchange of nodes to reduce cut size**
- **Allow cut size to increase temporarily within a pass**

Compute the gain of a swap
Repeat
Perform a feasible swap of max gain
Mark swapped nodes "locked";
Update swap gains;
Until no feasible swap;
Find max prefix partial sum in gain sequence $g_1, g_2, \ldots, g_m$
Make corresponding swaps permanent.



- **Start another pass if current pass reduces the cut size (usually converge after a few passes)**

A Useful Survey Paper

- Charles Alpert and Andrew Kahng, “Recent Directions in Netlist Partitioning: A Survey”, Integration: the VLSI Journal, 19(1-2), 1995, pp. 1-81.

- Next lecture: more on partitioning