ESE535: Electronic Design Automation

Day 11: March 4, 2008

Placement

(Intro, Constructive)

Today

- Placement Problem
- Partitioning→Placement
- Quadrisection
- Refinement

Placement

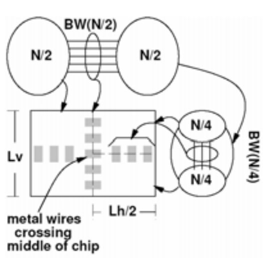
- **Problem**: Pick locations for all building blocks
  - minimizing energy, delay, area
  - really:
    - minimize wire length
    - minimize channel density

Bad Placement

- How bad can it be?
  - Area
  - Delay
  - Energy

Bad: Area

- All wires cross bisection
- $O(N^2)$ area
- good: O(N)

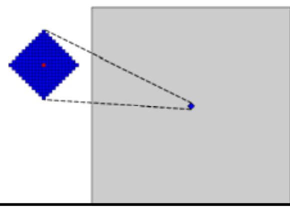


Bad: Delay

- All critical path wires cross chip
- Delay $=O(|PATH|*2*L_{side})$
  - [and L_{side} is O(N)]
- good: $O(|PATH|* L_{cell})$
- compare 50ps gates to many nanoseconds to cross chip

Clock Cycle Radius

- Radius of logic can reach in one cycle (45 nm)
  - 1 Cycle Radius = 10
    - Few hundred PEs
  - Chip side 600-700 PE
    - 400-500 thousand PEs
  - 100s of cycles to cross



Bad: Energy

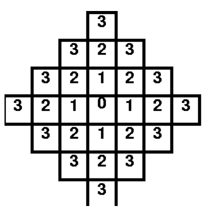
- All wires cross chip:
  $O(L_{side})$ long $\rightarrow$ $O(L_{side})$ capacitance per wire
  - Recall Area→$O(N^2)$
  - So L_{side} → O(N)

  ×O(N) wires $\rightarrow$ $O(N^2)$ capacitance

- Good:
  O(1) long wires $\rightarrow$ O(N) capacitance

Distance

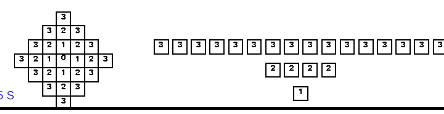
- Can we place everything close?



"Closeness"

- Try placing "everything" close

Manhattan Distance	Places	Transitive Fanin
1	4	4
2	8	16
3	12	64
⋮	⋮	⋮
n	$4n$	4^n



Illustration

- Consider a complete tree
  - nand2's, no fanout
  - N nodes
- Logical circuit depth?
- Circuit Area?
- Side Length?
- Average wire length between nand gates? (lower bound)

Another Example

- Consider a cut size F(N) > $\sqrt{N}$
- If optimally place all F(N) producers right next to bisection
  - How many cells deep is producer farthest from the bisection?
- Lower bound on wire length?

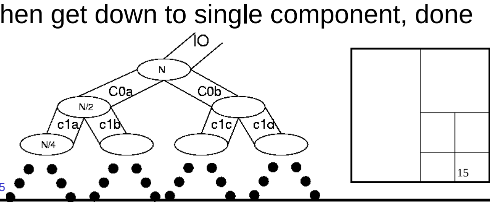
Problem Characteristics

- Familiar
  - NP Complete
  - local, greedy not work
  - greedy gets stuck in local minima

Constructive Placement

Basic Idea

- Partition (bisect) to define halves of chip
  - minimize wire crossing
- Recurse to refine
- When get down to single component, done



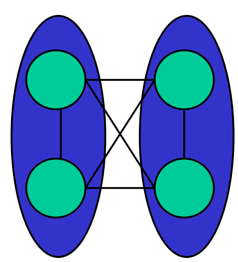
Adequate?

- Does recursive bisection capture the primary constraints of two-dimensional placement?

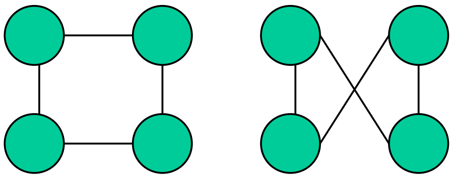
Problems

- Greedy, top-down cuts
  - maybe better pay cost early?
- Two-dimensional problem
  - (often) no real cost difference between H and V cuts
- Interaction between subtrees
  - not modeled by recursive bisect

Interaction



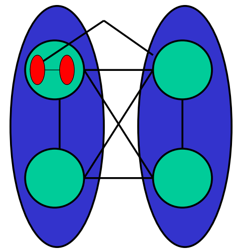
Example



Ideal split
(not typical)

"Equivalent" split
ignoring external constraints
Practically -- makes all H cuts also be V cuts

Interaction

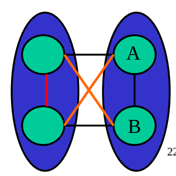


Problem

- Need to keep track of where things are
  - outside of current partition
  - include costs induced by above
- …but don't necessarily know where things are
  - still solving problem

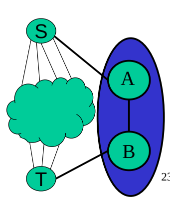
Improvement: Ordered

- Order operations
- Keep track of existing solution
- Use to constrain or pass costs to next subproblem



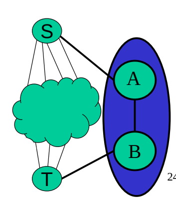
Improvement: Ordered

- Order operations
- Keep track of existing solution
- Use to constrain or pass costs to next subproblem
- Flow cut
  - use existing in src/sink
  - A nets = src, B nets = sink



Improvement: Ordered

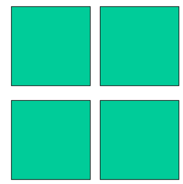
- Order operations
- Keep track of existing solution
- Use to constrain or pass costs to next subproblem
- Flow cut
  - use existing in src/sink
  - A nets = src, B nets = sink
- FM: start with fixed, unmovable nets for side-biased inputs

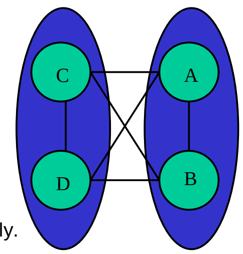


Improvement: Constrain

- Partition once
- Constrain movement within existing partitions
- Account for both H and V crossings
- Partition next
  - (simultaneously work parallel problems)
  - easy modification to FM

Constrain Partition





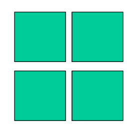
Solve AB and CD concurrently.

Improvement: Quadrisect

- Solve more of problem at once
- Quadrisection:
  - partition into 4 bins simultaneously
  - keep track of costs all around

Quadrisect

- Modify FM to work on multiple buckets
- k-way has:
  - k(k-1) buckets
  - |from|×|to|
  - quad→ 12
- reformulate gains
- update still O(1)

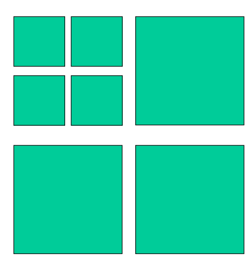


Quadrisect

- Cases (15):
  - (1 partition) → 4
  - (2 part) → 6 = (4 choose 2)
  - (3 part) → 4 = (4 choose 3)
  - (4 part) → 1

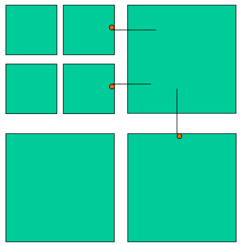
Recurse

- Keep outside constraints
  - (cost effects)
- Don't know detail place
- Model as at center of unrefined region



Option: Terminal Propagation

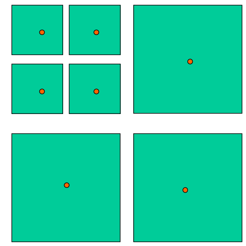
- Abstract inputs as terminals
- Partition based upon
- Represent cost effects on placement/refinement decisions



31

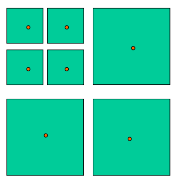
Option: Refine

- Keep refined placement
- Use in cost estimates



32

Problem

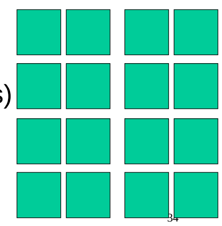


- Still have ordering problem
- Earlier subproblems solved with weak constraints from later
  - (cruder placement estimates)
- Solved previous case by flattening
  - …but in extreme give up divide and conquer

33

Iterate

- After solve later problems
- Relax solution
- Solve earlier problems again with refined placements (cost estimates)
- Repeat until converge



34

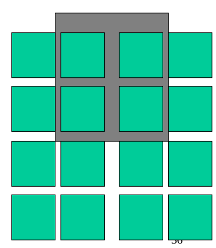
Iteration/Cycling

- General technique to deal with phase-ordering problem
  - what order do we perform transformations, make decisions?
  - How get accurate information to everyone
- Still basically greedy

35

Refinement

- Relax using overlapping windows
- Deal with edging effects
- Huang&Kahng claim 10-15% improve
  - cycle
  - overlap



36

Possible Refinement

- Allow unbalanced cuts
  - most things still work
  - just distort refinement groups
  - allowing unbalance using FM quadrisection looks a bit tricky
  - gives another 5-10% improvement

Penn ESE535 Spring 2008 -- DeHon 37

Runtime

- Each gain update still O(1)
  - (bigger constants)
  - so, FM partition pass still O(N)
- O(1) iterations expected
- assume O(1) overlaps exploited
- O(log(N)) levels
- Total: O(N log(N))
  - very fast compared to typical annealing
    - (annealing next time)

Penn ESE535 Spring 2008 -- DeHon 38

Quality: Area

Case	GORD-L	DOMINO	QUAD	Impr. GOR-L	Impr. DOMI
	MSTx100				
prim1	10500	10059	10208	2.8%	-1.5%
prim2	45994	43705	44478	3.3%	-1.8%
ind2	436300	417264	380194	12.9%	8.9%
ind3	1121000	1048673	970068	13.5%	7.5%
fract	400	383	380	5.0%	0.8%
C1908	1858	1767	1830	1.5%	-3.6%
C5315	6220	5922	6185	0.6%	-4.4%
C6288	8794	8339	8312	5.5%	0.3%
s1423	2334	2208	2265	3.0%	-2.6%
s1488	2680	2558	2470	7.8%	3.4%
s5378	8609	8182	8208	4.7%	-0.3%
s9234	14848	14023	13848	6.7%	1.3%
s13207	31284	29995	28161	9.9%	6.1%
s15850	37020	35591	33625	9.2%	5.5%
struct	4160	3967	4196	-0.9%	-5.8%
biomed	34677	33712	33787	2.6%	-0.2%
avq_s	95648	92355	95867	-0.2%	-3.8%
avq_l	100650	97825	101930	-1.3%	-4.2%
Impr.				4.8%	0.3%

[Huang&Kahng/ISPD1997]

Penn ESE535 Spring 2008 -- DeHon 39

Quality: Delay

- Weight edges based on criticality
  - Periodic, interleaved timing analysis

Case	Measure	Max Intrinsic Path Delay	TW7.0	Timing-QUAD
fract	Delay	10.6	17.9	18.1
	MSTx100		349	347
struct	Delay	40.0	78.8	79.3
	MSTx100		5130	5103
avq_s	Delay	37.3	61.4	60.9
	MSTx100		46763	47153

Penn ESE535 Spring 2008 -- DeHon 40

Uses

- Good by self
- Starting point for simulated annealing
  - speed convergence
- With synthesis (both high level and logic)
  - get a quick estimate of physical effects
  - (play role in estimation/refinement at larger level)
- Early/fast placement
  - before willing to spend time looking for best
- For fast placement where time matters
  - FPGAs, online placement?

Penn ESE535 Spring 2008 -- DeHon 41

Summary

- Partition to minimize cut size
- Additional constraints to do well
  - Improving constant factors
- Quadrisection
- Keep track of estimated placement
- Relax/iterate/Refine

Penn ESE535 Spring 2008 -- DeHon 42

Admin

- Reading for Wednesday
- Assignment 4 out
  - Placement (and clustering) for timing

Big Ideas:

- Potential dominance of interconnect
- Divide-and-conquer
- Successive Refinement
- Phase ordering: estimate/relax/iterate