

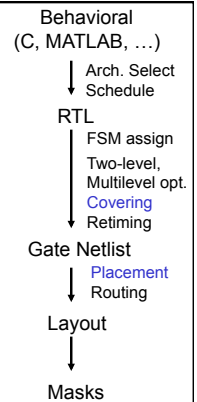
ESE535: Electronic Design Automation

Day 25: April 20, 2011
Dual Objective
Dynamic Programming



Today

- Cover and Place
 - Linear
 - GAMA
 - Optimal Tree-based
- Area and Time
 - covering for
 - ...and linear placement
- Two Dimensional Cover and Place
 - Lily



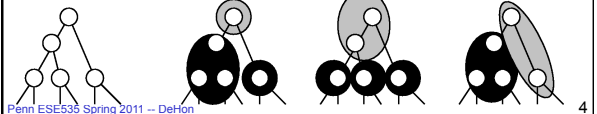
Covering Review

- Use dynamic programming to optimally cover trees
 - problem decomposable into subproblems
 - optimal solution to each are part of optimal
 - no interaction between subproblems
 - small number of distinct subproblems
 - single optimal solution to subproblem
- Break DAG into trees then cover optimally

Covering Basics

Basic Idea:

- Assume have optimal solution to all subproblems smaller than current problem
- Try all ways of implementing current root
 - each candidate solution is new gate + previously solve subtrees
- Pick best
 - (smallest area, least delay, least power)



Placement

- How do we integrate placement into this covering process?

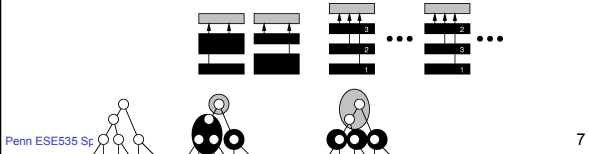
GaMa - Linear Placement

- **Problem:** cover and place datapaths in rows of FPGA-like cells to minimize area, delay
- Datapath width extends along one dimension (rows)
- Composition is 1D along other dimension (columns)
- Always covering row at a time

[Callahan/FPGA'98]

Basic Strategy

- Restrict each subtree to a contiguous set of rows
- Build up placement for subtree during cover
- When consider cover, also consider all sets of arrangements of subtrees
 - effectively expands library set



Penn ESE535 Sp

7

Simultaneous Placement Benefits

- Know real delay (including routing) during covering
 - make sure critical logic uses fastest inputs
 - ...shortest paths
- Know adjacency
 - can use special resources requiring adjacent blocks
 - Carry chains, direct connections

Penn ESE535 Spring 2011 -- DeHon

8

GaMa Properties

- Operates in time linear in graph size
 - $O(|\text{rule set}| \times |\text{graph nodes}|)$
- Finds area-optimum for restricted problem
 - trees with contiguous subtrees
- As is, may not find delay optimum

Penn ESE535 Spring 2011 -- DeHon

9

GaMa Example

- $Y = A \times B + (\text{abs}(C) + \text{ilog2}(D))$

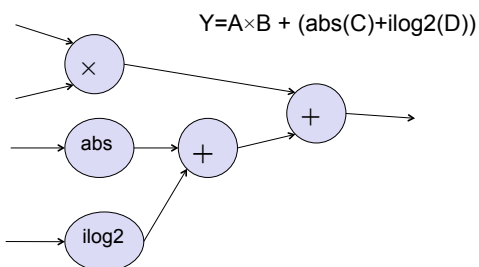
Distance	Delay
0—3 rows	0
4—7 rows	1
8+ rows	2

add	A=1, D=1	
multiply	A=8, T=4	A=6, T=6
abs	A=3, T=2	A=2, T=3
ilog2	A=4, T=4	A=3, T=5

Penn ESE535 Spring 2011 -- DeHon

10

Example: Cover for Delay



Penn ESE535 Spring 2011 -- DeHon

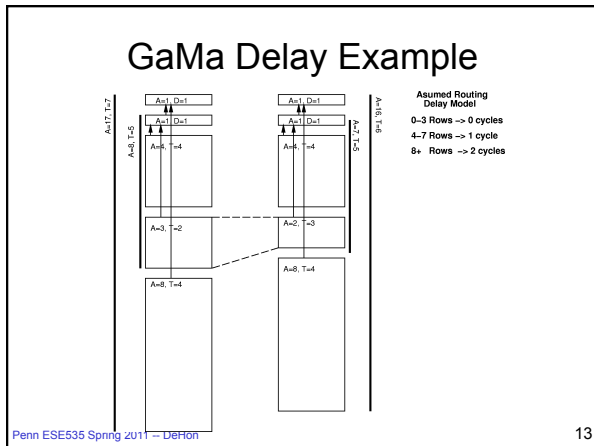
11

How Change?

add	A=1, D=1	
multiply	A=8, T=4	A=6, T=6
abs	A=3, T=2	A=2, T=3
ilog2	A=4, T=4	A=3, T=5

Penn ESE535 Spring 2011 -- DeHon

12



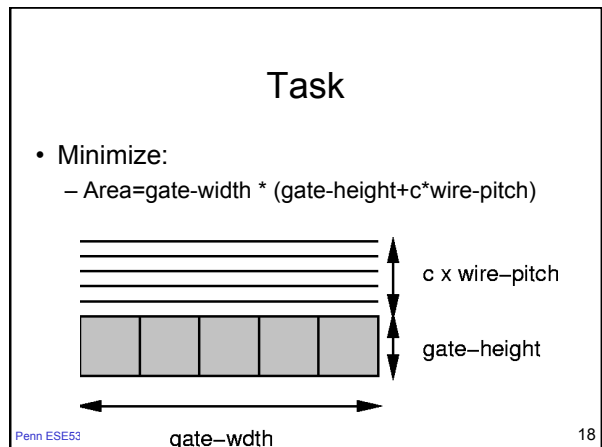
- ### GaMa Delay Problem
- Area can affect delay
 - Doesn't know when to pick worse delay to reduce area
 - make non-critical path subtree slower/smaller
 - so overall critical path will be close later
 - Only tracking single objective
 - Fixable as next technique demonstrates
- Penn ESE535 Spring 2011 -- DeHon 14

- ### GaMa Results
- Comparable result quality (area, time) to running through Xilinx tools
 - Placement done in **seconds** as opposed to minutes to hours for Xilinx
 - simulated annealing, etc.
 - not exploiting datapath regularity
- Penn ESE535 Spring 2011 -- DeHon 15

GaMa Questions?

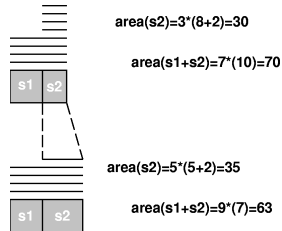
Penn ESE535 Spring 2011 -- DeHon 16

- ### Simultaneous Mapping and Linear Placement of Trees
- **Problem:** cover and place standard cell row minimizing area
 - **Area:** cell width and cut width
 - **Technique:** combine DP-covering with DP-tree layout
- [Lou+Salek+Pedram/ICCAD'97]
 Penn ESE535 Spring 2011 -- DeHon 17



Composition Challenge

- Minimum area solution to subproblems does **not** necessarily lead to minimum area solution:



Penn ESE535 Spring 2011 -- DeHon

19

Minimize Area

- Two components of area:
 - gate-area
 - cut-width
- Unclear during mapping when need
 - a smaller gate-area
 - vs. a smaller cut-width
 - at the expense of (local) cell area
 - (same problem as area vs. delay in GaMa)

Penn ESE535 Spring 2011 -- DeHon

20

Strategy

- Recognize that these are incomparable objectives
 - neither is strictly superior to other
 - keep all solutions
 - discard only inferior (dominated) solutions

Penn ESE535 Spring 2011 -- DeHon

21

Dominating/Inferior Solutions

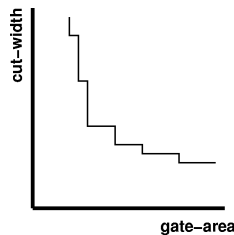
- A solution is **dominated** if there is another solution strictly superior in all objectives
 - $A=3, T=2$ $A=2, T=3$
 - neither dominates
 - $A=3, T=3$ $A=3, T=2$ $A=2, T=3$
 - $A=3, T=3$ is inferior, being dominated by either of the other two solutions

Penn ESE535 Spring 2011 -- DeHon

22

Non-Inferior Curve

- Set of dominators defines a curve



This is a recurring theme---often prune work using dominator curve

Penn ESE535 Spring 2011 -- DeHon

23

Strategy

- Keep curve of non-inferior area-cut points
- During DP
 - build a new curve for each subtree
 - by looking at solution set intersections
 - cross product set of solutions from each subtrees feeding into this subtree

Penn ESE535 Spring 2011 -- DeHon

24

Consequences

- More work per graph point
 - keeping and intersecting many points
- Theory: points^(fanin) × gates
- Points ≤ range of solutions in smallest dimension
- e.g. points ≤ number of different cut-widths

Algorithm: Tree Cover+Place

- For each tree node from leaves
 - For each gate cover
 - For each non-inferior point in fanin-subtrees
 - compute optimal tree layout
 - keep non-inferior points (cutwidth, gate-area)
- Optimal Tree Layout
 - Yannakakis/JACM v32n4p950, Oct. 1985

Time Notes

- Computing Optimal Tree layout:
 $O(N \log(N))$
- Per node: $O(\text{cutwidth}^{\text{fanin}} * N * \log(N))$
- Loose bound
 - possible to tighten?
 - less points and smaller “N” in tree for earlier subproblems
 - higher fanin → less depth → more use of small “N” for linear layout problems

Empirical Results

- Claim: 20% area improvement

Area: Questions

Covering for Area and Delay (no placement)

- Previously saw was hard to do DP to
 - simultaneously optimize for area and delay
 - properly generate area-time tradeoffs
- **Problem:**
 - whether or not needed a fast path
 - not clear until saw speed of siblings

Strategy

- Use same technique as just detailed for
 - gate-area + cutwidth
- *I.e.* -- at each tree cover
 - keep all non-inferior points
 - (effectively the full area-time curve)
 - as cover, intersect area-time curves to generate new area-time curve
- When get to a node
 - can pick smallest implementation for a child node that does not increase critical path

Penn ESE535 Spring 2011 -- DeHon

31

Points to Keep

- Usually small variance in times
 - if use discrete model like LUT delays, only a small number of different times
 - if use continuous model, can get close to optimum by discretizing and keeping a fixed set
- Similarly, small total variance in area
 - e.g. factor of 2-3
 - discretizing, gets close w/out giving up much
- Discretized: run in time linear in N
 - assuming bounded fanin gates

Penn ESE535 Spring 2011 -- DeHon

32

GaMa -- Optimal Delay

- Use this technique in GaMa
 - solve delay problem
 - get good area-delay tradeoffs
 - GARP has a discrete timing model
 - so already have small spread
 - for conventional FPGA
 - will have to discretize

Penn ESE535 Spring 2011 -- DeHon

33

Rework GaMa Example

- $Y = A \times B + (\text{abs}(C) + \text{ilog2}(D))$

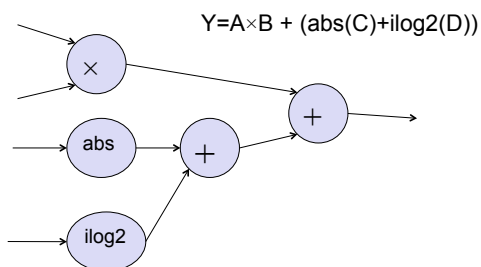
Distance	Delay
0—3 rows	0
4—7 rows	1
8+ rows	2

add	A=1, D=1	
multiply	A=8, T=4	A=6, T=6
abs	A=3, T=2	A=2, T=3
ilog2	A=4, T=4	A=3, T=5

Penn ESE535 Spring 2011 -- DeHon

34

Example



Penn ESE535 Spring 2011 -- DeHon

35

Area & Delay: Questions?

Penn ESE535 Spring 2011 -- DeHon

36

Covering and Linear Placement for Area and Delay

- Have both
 - cut-width + gate-area affects
 - delay tradeoff
- Result
 - have three objectives to minimize
 - cut-width
 - gate-area
 - gate-delay

[Lou+Salek+Pedram/ICCAD'97]

Penn ESE535 Spring 2011 -- DeHon

37

Strategy

- Repeat trick:
 - keep non-inferior points in three-space
 - <cut-width,gate-area,delay>
 - Intersect spaces to compute new cover spaces
 - May really need to discretize points to limit work

Penn ESE535 Spring 2011 -- DeHon

38

Note

- Delay calculation:
 - assumes delay in gates and fanout
 - fanout effect makes heuristic
 - maybe iterate/relax?
 - ignores distance
- “Optimal” tree layout algorithm being used
 - is optimal with respect to cut-width
 - **not optimal** with respect to critical path wire length

Penn ESE535 Spring 2011 -- DeHon

39

Empirical Results

- Mapping for delay:
 - 20% delay improvement
 - achieving effectively same area
 - (of alternative, not of self targeting area)

Penn ESE535 Spring 2011 -- DeHon

40

Placement Area-Delay Questions?

Penn ESE535 Spring 2011 -- DeHon

41

Two Dimensions?

- Both so far, one-dimensional
- One-dimensional
 - nice layout restrictions
 - simple metric for delay
 - simple metric for area
- How extend to two dimensions?

Penn ESE535 Spring 2011 -- DeHon

42

2D Cover and Place

- **Problem:** cover and place in 2D to minimize area (delay)
- Area: gate area + “wirelength” area
- Delay: gate delay + estimated wire delay

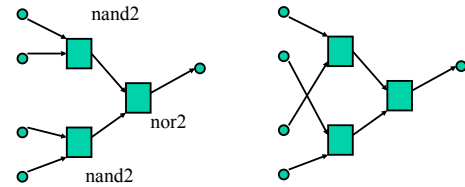
[Pedram+Bhat/DAC'91]

Penn ESE535 Spring 2011 -- DeHon

43

Example

- Covering wrt placement matters



$$\text{nor2}(\text{nand2}(A,B), \text{nand2}(C,D)) = \text{AND}(A,B,C,D)$$

Penn ESE535 Spring 2011 -- DeHon

44

Strategy

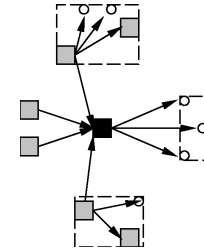
- Relax placement during covering
- Initially place unmapped using constructive placement (Day 9)
- Cover via dynamic programming
- When cover a node,
 - fanins already visited
 - calculate new placement
 - Center of Mass
- Periodically re-calculate placement
- Use estimated/refined placements to get area, delay

Penn ESE535 Spring 2011 -- DeHon

45

Incremental Placement

- Place newly covered nodes so as to minimize wire lengths (critical path delay?)



Penn ESE535 Spring 2011 -- DeHon

46

Empirical Results

- In 1 μm
 - 5% area reduction
 - 8% delay reduction
- Not that inspiring
 - ...but this was in the **micron** era
 - probably have a bigger effect today

Penn ESE535 Spring 2011 -- DeHon

47

2D Place and Cover Questions?

Penn ESE535 Spring 2011 -- DeHon

48

Summary

- Can consider placement effects while covering
- Many problems can't find optimum by minimizing single objective
 - delay (area effects)
 - area (cutwidth effects)
- Can adapt DP to solve
 - keep all non-inferior points
 - can keep polynomial time
 - if very careful, primarily increase constants

Admin

- Reading on web
- Projects due Monday
- Course evaluations online

Big Ideas:

- Simultaneous optimization
- Multi-dimensional objectives
 - dominating points (inferior points)
 - use with dynamic programming
- Exploit stylized problems can solve optimally
- Phase Ordering: estimate/iterate