

ESE535: Electronic Design Automation

Day 17: March 30, 2009
Clustering
(LUT Mapping, Delay)



Penn ESE535 Spring 2009 -- DeHon

Background Question

- Experience with FPGAs?
- What provides programmable logic in an FPGA?

2

Penn ESE535 Spring 2009 -- DeHon

Today

- How do we map to LUTs?
- What happens when delay dominates?
- Lessons...
 - for non-LUTs
 - for delay-oriented partitioning

3

Penn ESE535 Spring 2009 -- DeHon

LUT Mapping

- **Problem:** Map logic netlist to LUTs
 - minimizing area
 - minimizing delay
- Old problem?
 - Technology mapping? (Last Monday)
 - How big is the library?
 - 2^{2^K} gates in library

4

Penn ESE535 Spring 2009 -- DeHon

Simplifying Structure

- K-LUT can implement any K-input function

5

Penn ESE535 Spring 2009 -- DeHon

Cost Function

- **Delay:** number of LUTs in critical path
 - doesn't say delay in LUTs or in wires
 - does assume uniform interconnect delay
- **Area:** number of LUTs
 - Assumes adequate interconnect to use LUTs

6

Penn ESE535 Spring 2009 -- DeHon

LUT Mapping

- NP-Hard in general
- Fanout-free -- can solve optimally *given* decomposition
 - (but which one?)
- Delay optimal mapping achievable in Polynomial time
- Area w/ fanout NP-complete

Penn ESE535 Spring 2009 -- DeHon

7

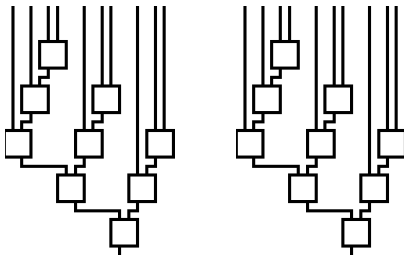
Preliminaries

- What matters/makes this interesting?
 - Area / Delay target
 - Decomposition
 - Fanout
 - replication
 - reconvergent

Penn ESE535 Spring 2009 -- DeHon

8

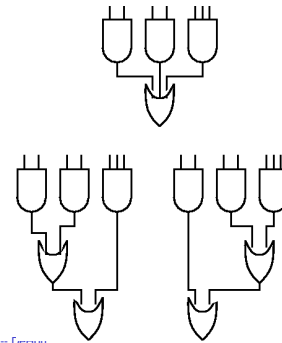
Area vs. Delay



Penn ESE535 Spring 2009 -- DeHon

9

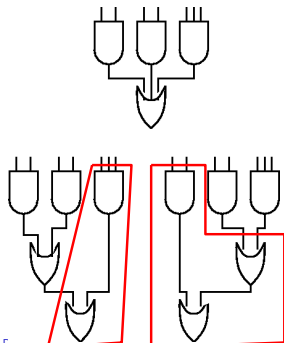
Decomposition



Penn ESE535 Spring 2009 -- DeHon

10

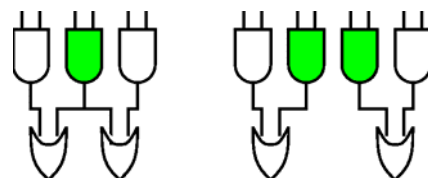
Decomposition



Penn ESE535 Spring 2009 -- DeHon

11

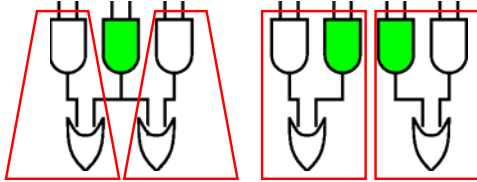
Fanout: Replication



Penn ESE535 Spring 2009 -- DeHon

12

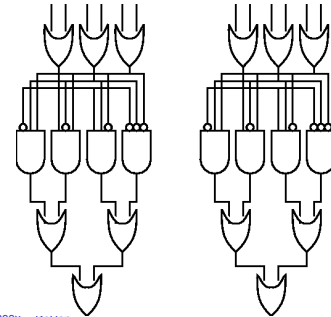
Fanout: Replication



Penn ESE535 Spring 2009 -- DeHon

13

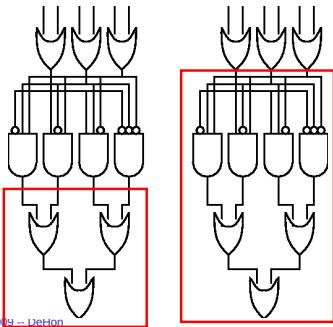
Fanout: Reconvergence



Penn ESE535 Spring 2009 -- DeHon

14

Fanout: Reconvergence



Penn ESE535 Spring 2009 -- DeHon

15

Monotone Property

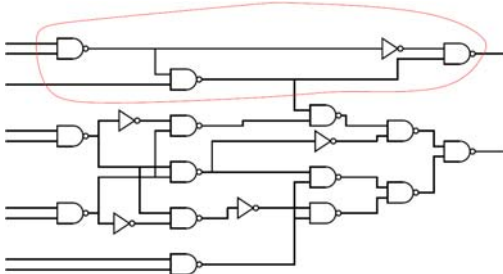
- Does cost function increase monotonically as more of the graph is included? (do all subsets have property)
 - gate count?
 - I/O?
- Important?
 - How far back do we need to search?

Penn ESE535 Spring 2009 -- DeHon

16

Definition

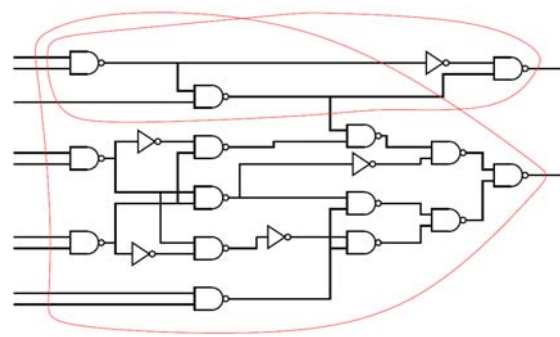
- **Cone:** set of nodes in the recursive fanin of a node



Penn ESE535 Spring 2009 -- DeHon

17

Example Cones



Penn ESE535 Spring 2009 -- DeHon

18

Delay

Penn ESE535 Spring 2009 -- DeHon

19

Dynamic Programming

- Optimal covering of a logic cone is:
 - Minimum cost (all possible coverings)
- Evaluate costs of each node based on:
 - cover node
 - cones covering each fanin to node cover
- Evaluate node costs in topological order
- **Key:** are calculating optimal solutions to subproblems
 - only have to evaluate covering options at each node

Penn ESE535 Spring 2009 -- DeHon

20

Flowmap

- **Key Idea:**
 - LUT holds anything with K inputs
 - Use network flow to find cuts
 - $\equiv$ logic can pack into LUT including reconvergence
 - ...allows replication
 - Optimal depth arise from optimal depth solution to subproblems

Penn ESE535 Spring 2009 -- DeHon

21

Day 10

MaxFlow

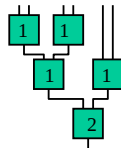
- Set all edge flows to zero
 - $F[u,v]=0$
- While there is a path from s,t
 - (breadth-first-search)
 - for each edge in path $f[u,v]=f[u,v]+1$
 - $f[v,u]=f[u,v]$
 - When $c[v,u]=f[v,u]$ remove edge from search
- $O(|E| \cdot \text{cutsizes})$

Penn ESE535 Spring 2009 -- DeHon

22

Flowmap

- Delay objective:
 - minimum height, K-feasible cut
 - *i.e.* cut no more than K edges
 - start by bounding fanin $\leq K$
- Height of node will be:
 - height of predecessors *or*
 - one greater than height of predecessors
- Check shorter first



Penn ESE535 Spring 2009 -- DeHon

23

Flowmap

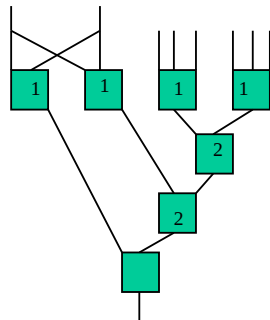
- Construct flow problem
 - sink $\leftarrow$ target node being mapped
 - source $\leftarrow$ start set (primary inputs)
 - flow infinite into start set
 - flow of one on each link
 - to see if height same as predecessors
 - collapse all predecessors of maximum height into sink (single node, cut must be above)
 - height +1 case is trivially true

Penn ESE535 Spring 2009 -- DeHon

24

Example Subgraph

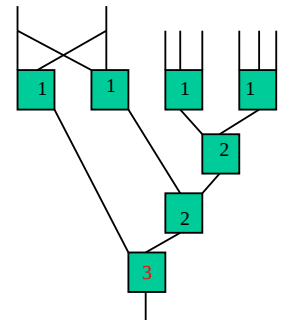
Target: $K=4$



Penn ESE535 Spring 2009 -- DeHon

25

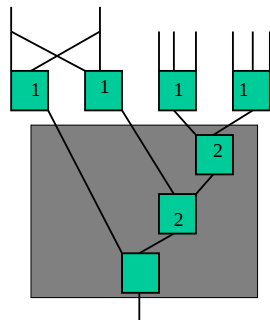
Trivial: Height +1



Penn ESE535 Spring 2009 -- DeHon

26

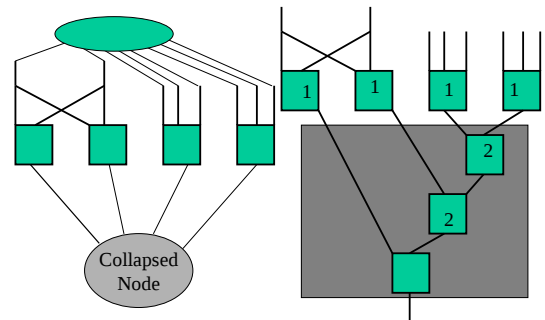
Collapse at max height



Penn ESE535 Spring 2009 -- DeHon

27

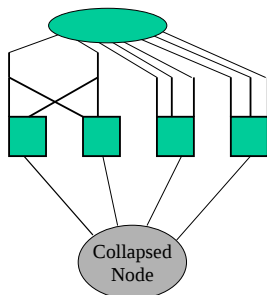
Collapse at max height



Penn ESE535 Spring 2009 -- DeHon

28

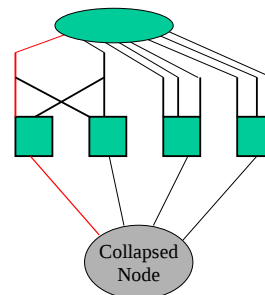
Augmenting Flows



Penn ESE535 Spring 2009 -- DeHon

29

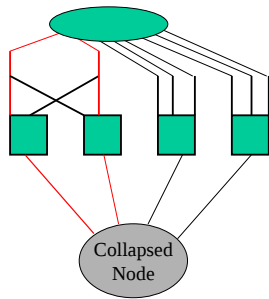
Augmenting Flows



Penn ESE535 Spring 2009 -- DeHon

30

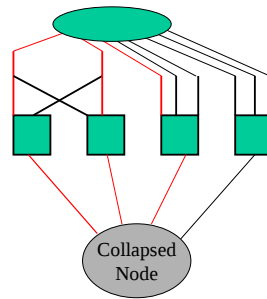
Augmenting Flows



Penn ESE535 Spring 2009 -- DeHon

31

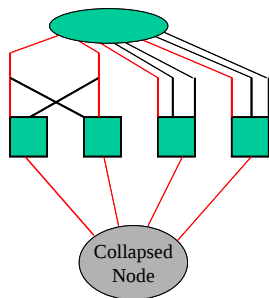
Augmenting Flows



Penn ESE535 Spring 2009 -- DeHon

32

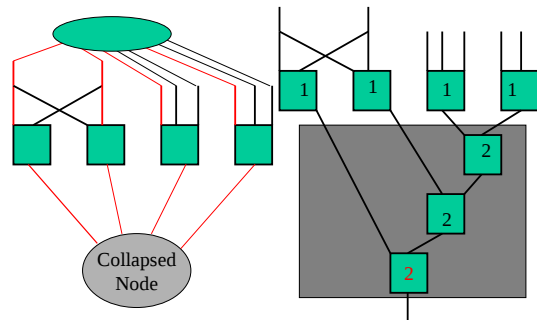
Augmenting Flows



Penn ESE535 Spring 2009 -- DeHon

33

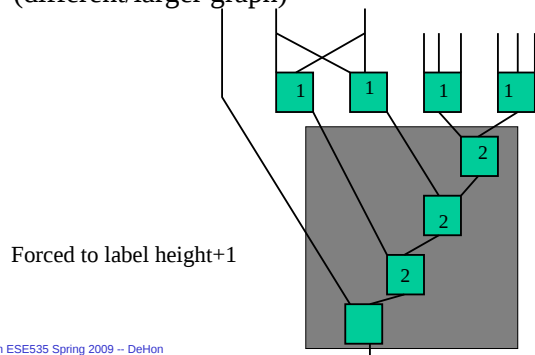
Collapse at max height



Penn ESE535 Spring 2009 -- DeHon

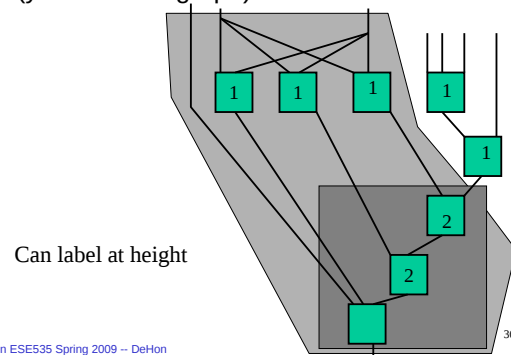
34

Collapse not work (different/larger graph)



Penn ESE535 Spring 2009 -- DeHon

Reconvergent fanout (yes different graph)



Penn ESE535 Spring 2009 -- DeHon

36

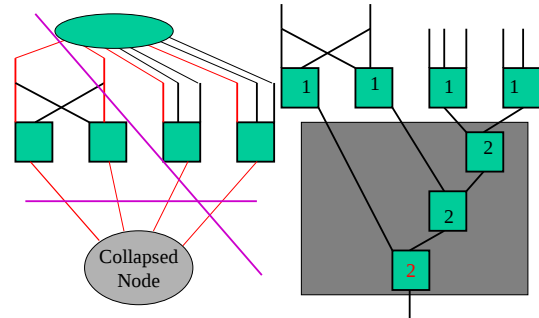
Flowmap

- Max-flow Min-cut algorithm to find cut
- Use augmenting paths until discover max flow $> K$
- $O(K|e|)$ time to discover K-feasible cut
 - (or that does not exist)
- Depth identification: $O(KN|e|)$

Penn ESE535 Spring 2009 -- DeHon

37

Mincut may not be unique

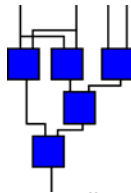


Penn ESE535 Spring 2009 -- DeHon

38

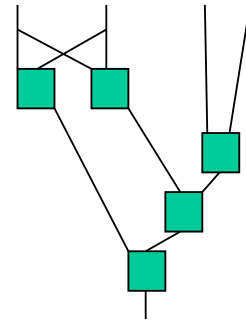
Flowmap

- Min-cut may not be unique
- To minimize area achieving delay optimum
 - find max volume min-cut
 - Compute max flow $\Rightarrow$ find min cut
 - remove edges consumed by max flow
 - DFS from source
 - Complement set is max volume set



Penn ESE535 Spring 2009 -- DeHon

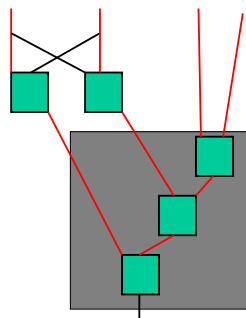
Graph



Penn ESE535 Spring 2009 -- DeHon

40

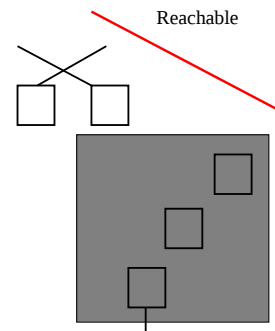
Graph: maxflow (K=4)



Penn ESE535 Spring 2009 -- DeHon

41

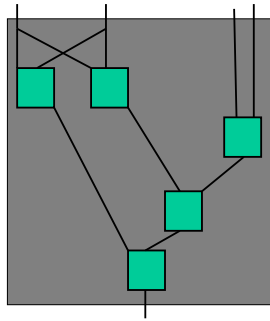
Graph: BFS source



Penn ESE535 Spring 2009 -- DeHon

42

Max Volume min-cut



Penn ESE535 Spring 2009 -- DeHon

43

Flowmap

- Covering from labeling is straightforward
 - process in reverse topological order
 - allocate identified K-feasible cut to LUT
 - remove node
 - postprocess to minimize LUT count
- Notes:
 - replication implicit (covered multiple places)
 - nodes purely internal to one or more covers may not get their own LUTs

Penn ESE535 Spring 2009 -- DeHon

44

Flowmap Roundup

- Label
 - Work from inputs to outputs
 - Find max label of predecessors
 - Collapse new node with all predecessors at this label
 - Can find flow cut $\leq K$?
 - Yes: mark with label (find max-volume cut extent)
 - No: mark with label+1
- Cover
 - Work from outputs to inputs
 - Allocate LUT for identified cluster/cover
 - Recurse covering selection on inputs to identified LUT

Penn ESE535 Spring 2009 -- DeHon

45

Area

Penn ESE535 Spring 2009 -- DeHon

46

DF-Map

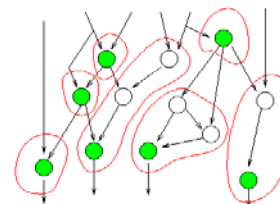
- Duplication Free Mapping
 - can find optimal area under this constraint
 - (but optimal area may not be duplication free)

[Cong+Ding, IEEE TR VLSI Sys. V2n2p137]

Penn ESE535 Spring 2009 -- DeHon

47

Maximum Fanout Free Cones



MFFC: bit more general than trees

Penn ESE535 Spring 2009 -- DeHon

48

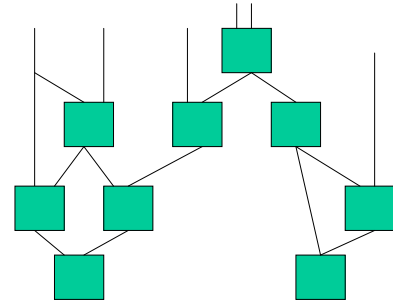
MFFC

- Follow cone backward
- end at node that fans out (has output) outside the code

Penn ESE535 Spring 2009 -- DeHon

49

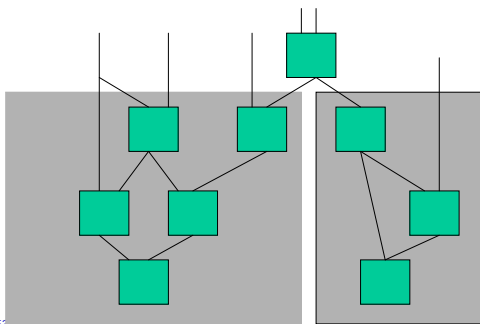
MFFC example



Penn ESE535 Spring 2009 -- DeHon

50

MFFC example



Penn ESE535 Spring 2009 -- DeHon

DF-Map

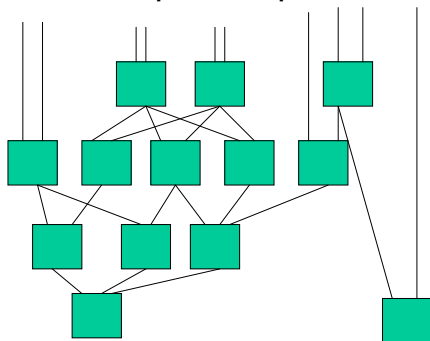
- Partition into graph into MFFCs
- Optimally map each MFFC
- In dynamic programming
 - for each node
 - examine **each** K-feasible cut
 - **note: this is very different than flowmap where only had to examine a single cut**
 - pick cut to minimize cost
 - $1 + \sum \text{MFFCs for fanins}$

Penn ESE535 Spring 2009 -- DeHon

52

DF-Map Example

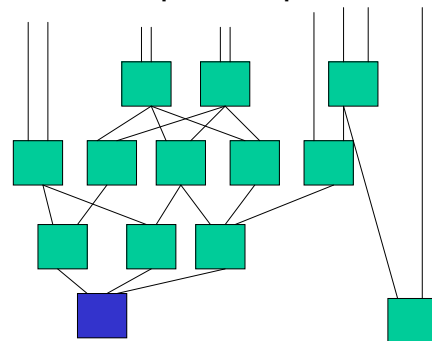
Cones?



Penn ESE535 Spring 2009 -- DeHon

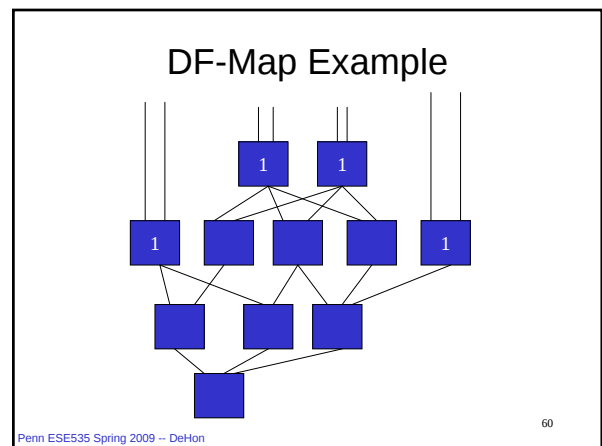
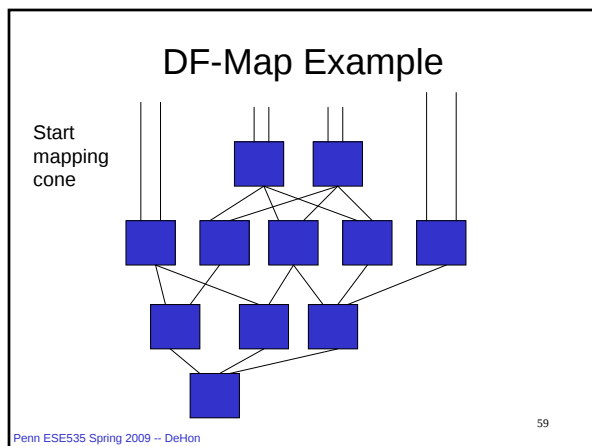
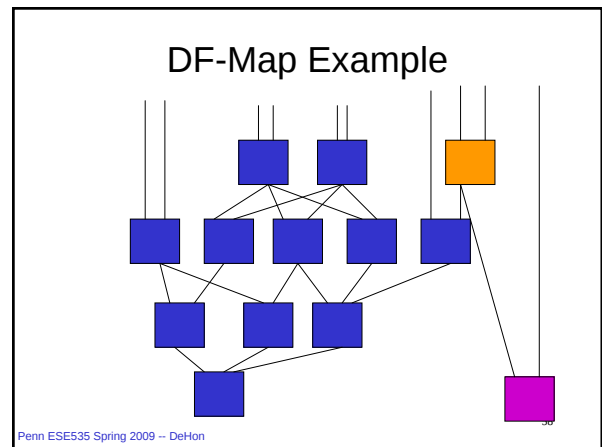
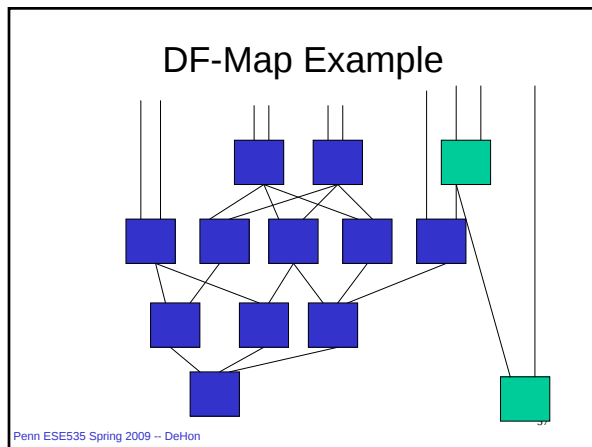
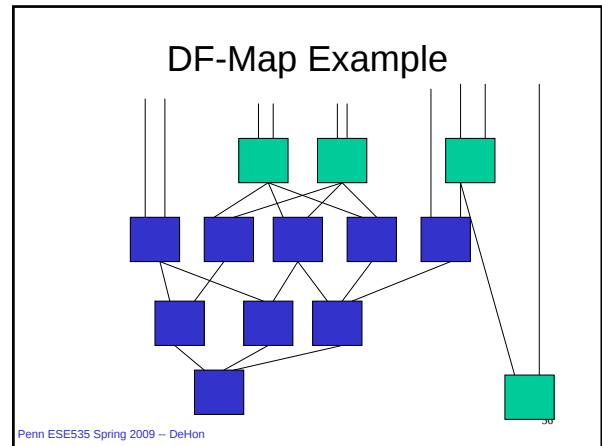
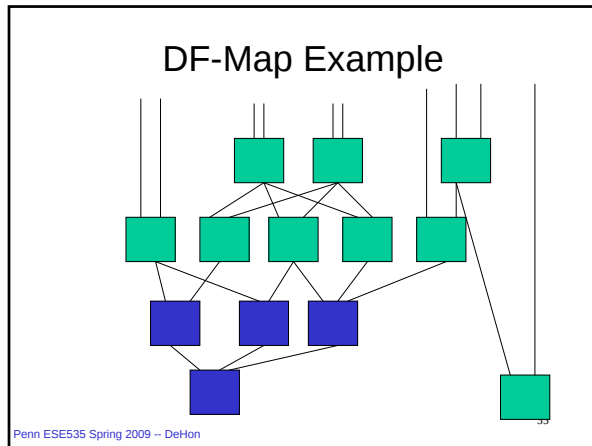
53

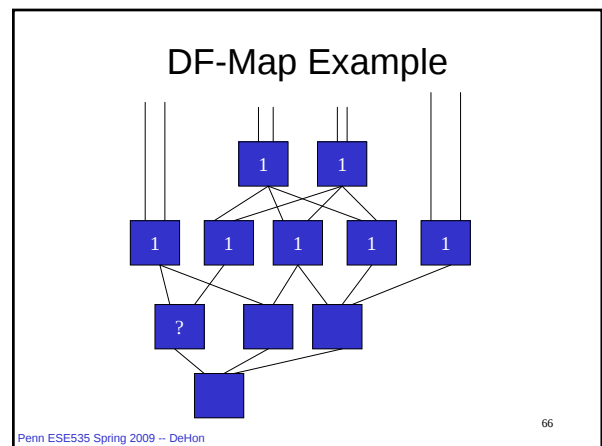
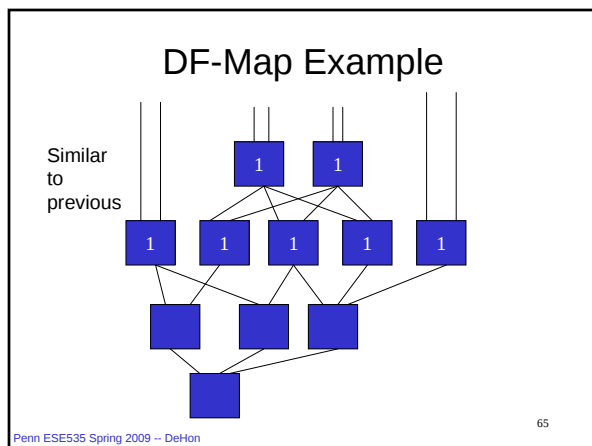
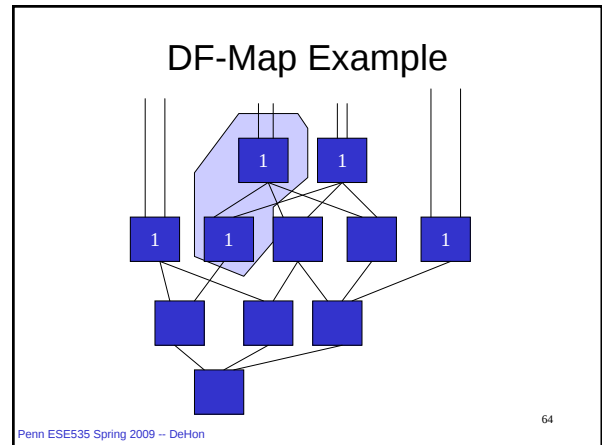
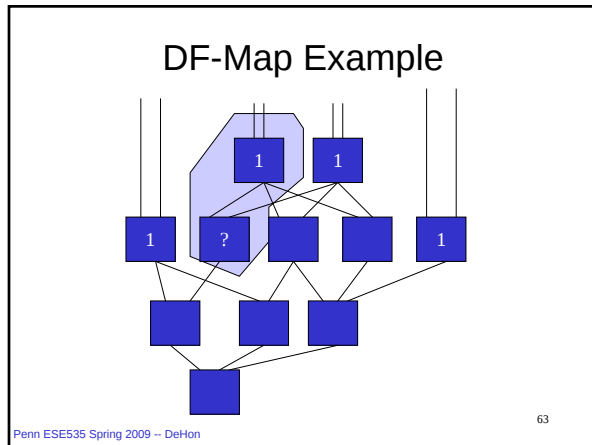
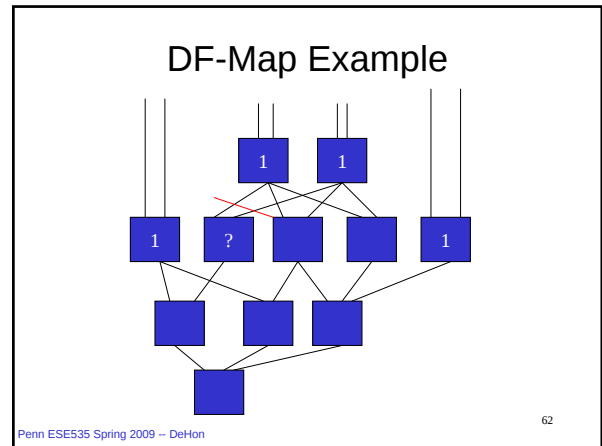
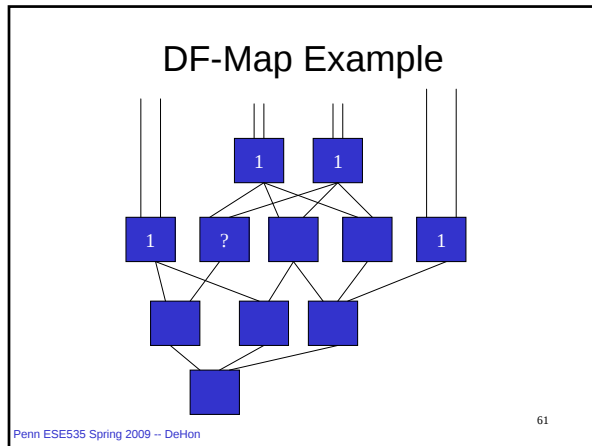
DF-Map Example

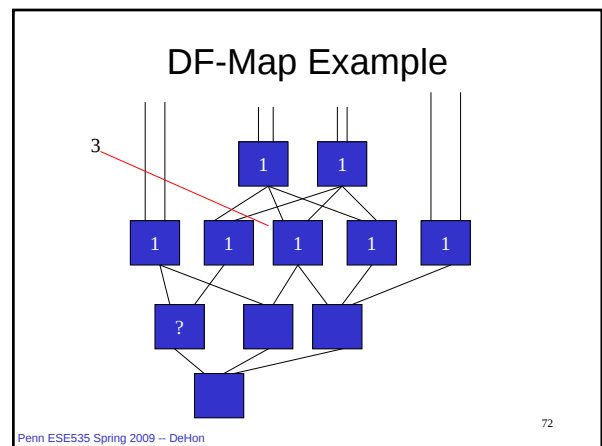
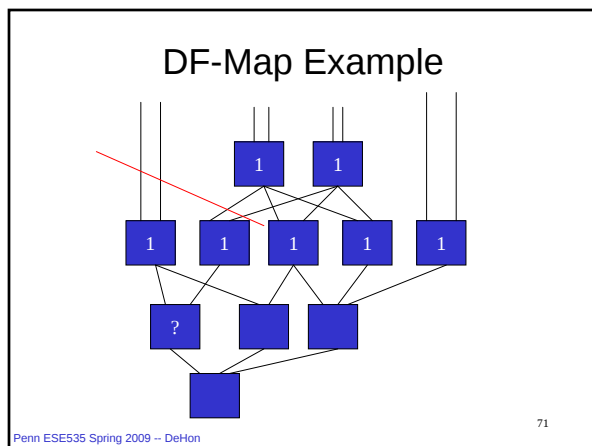
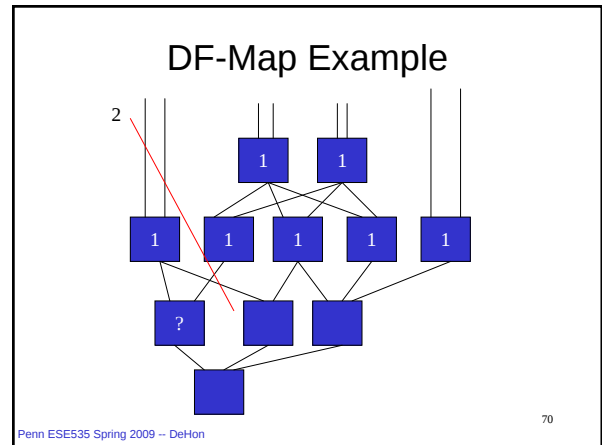
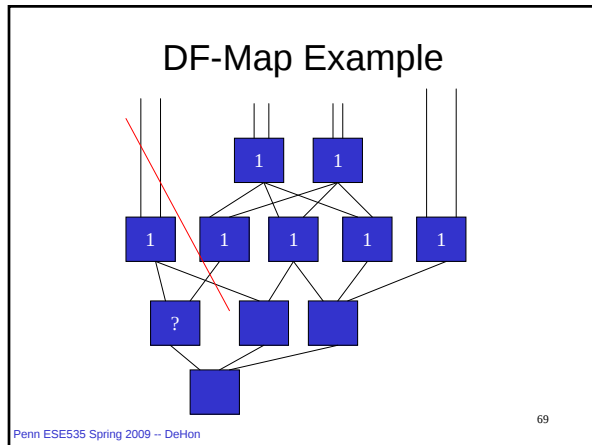
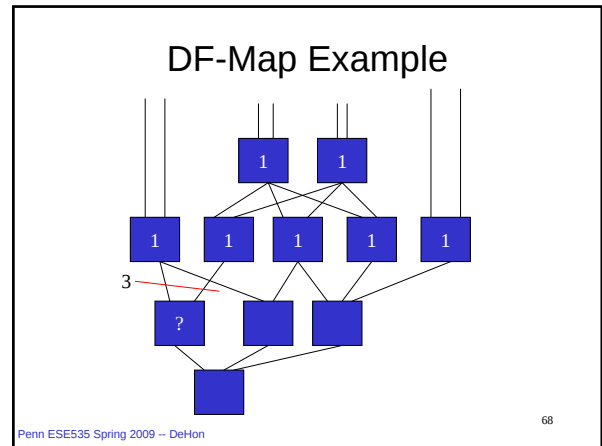
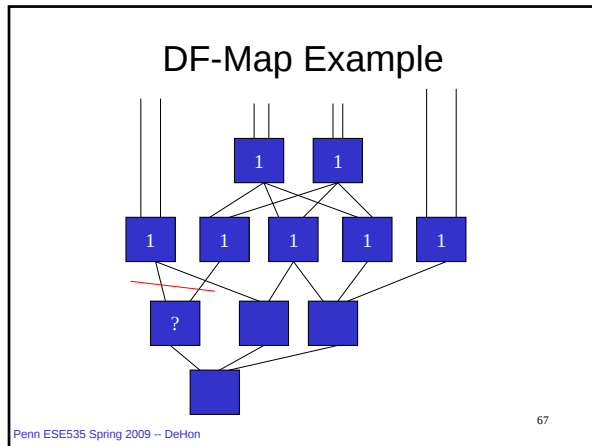


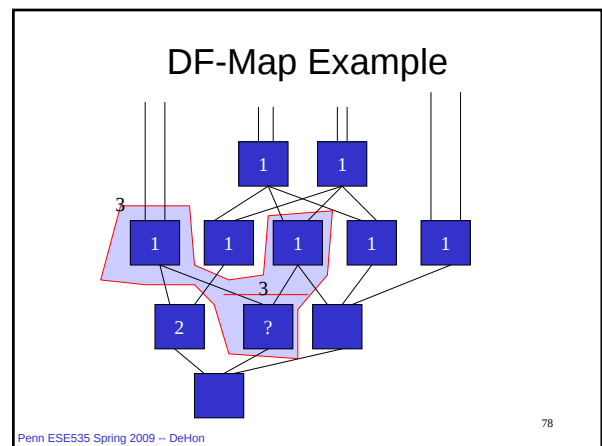
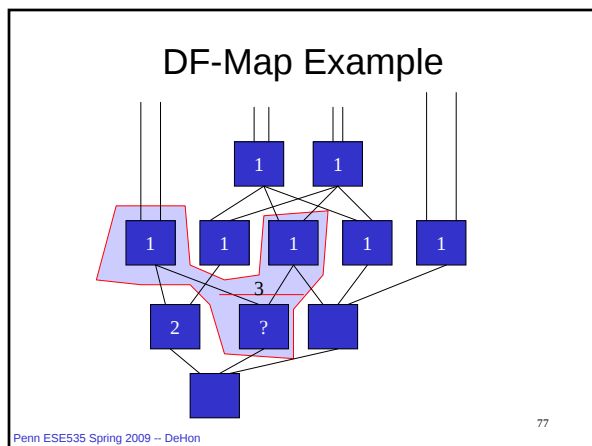
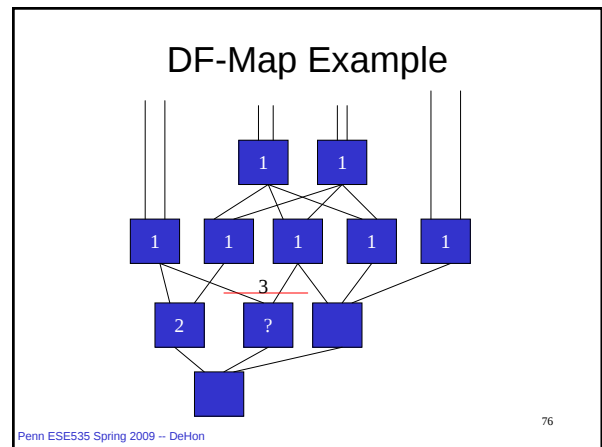
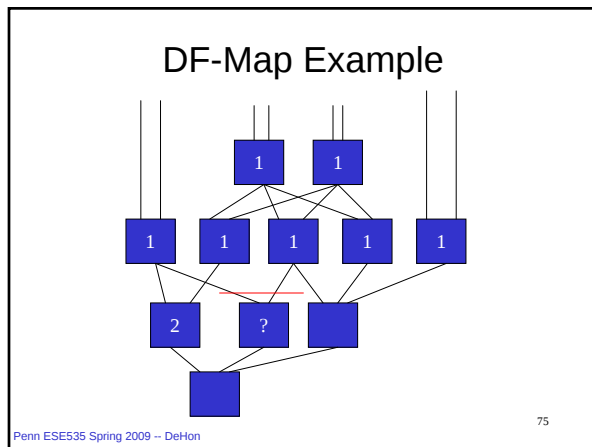
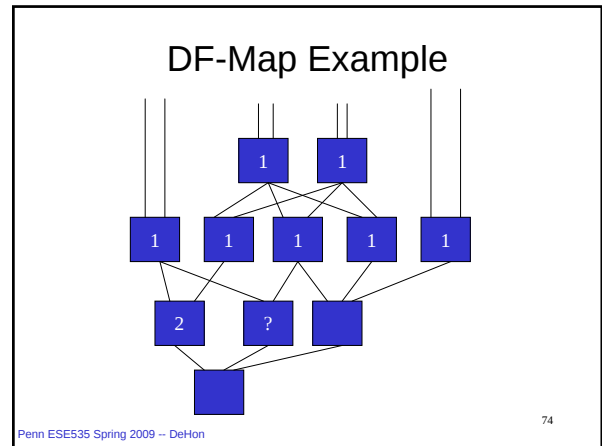
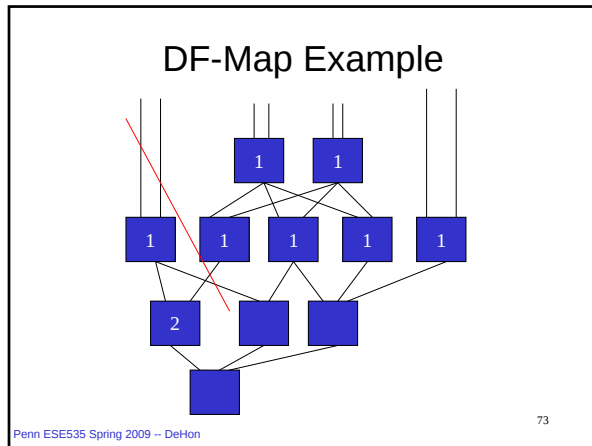
Penn ESE535 Spring 2009 -- DeHon

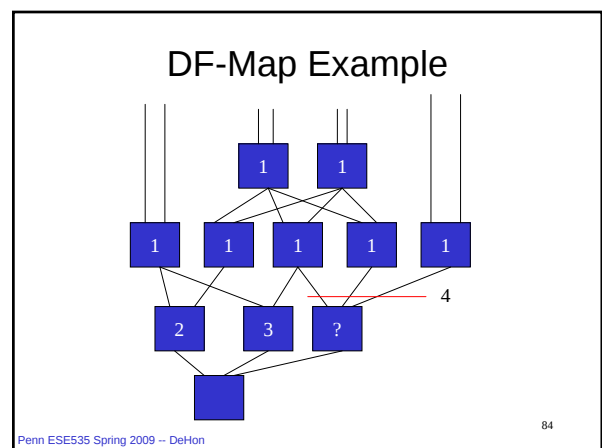
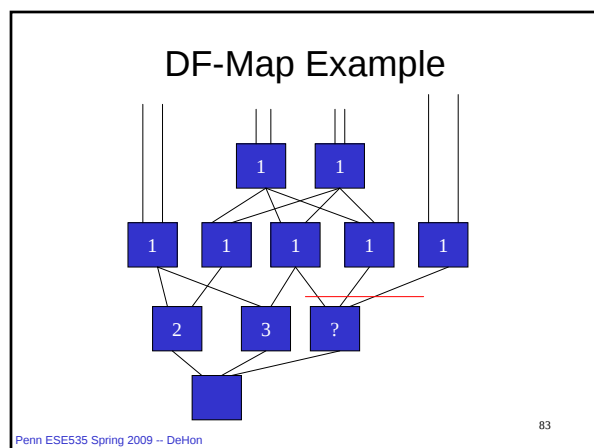
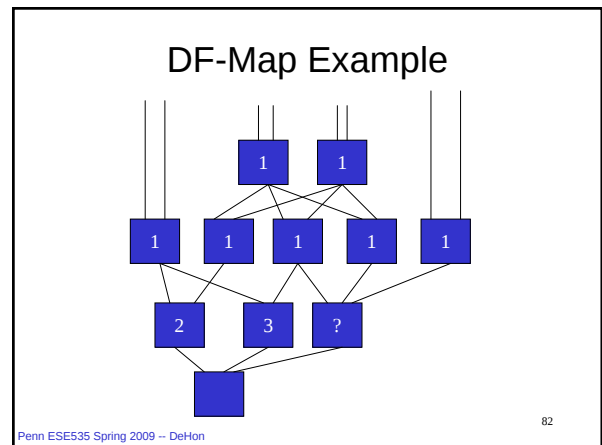
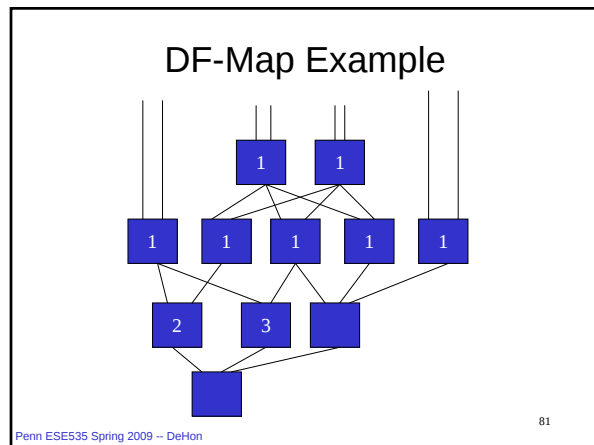
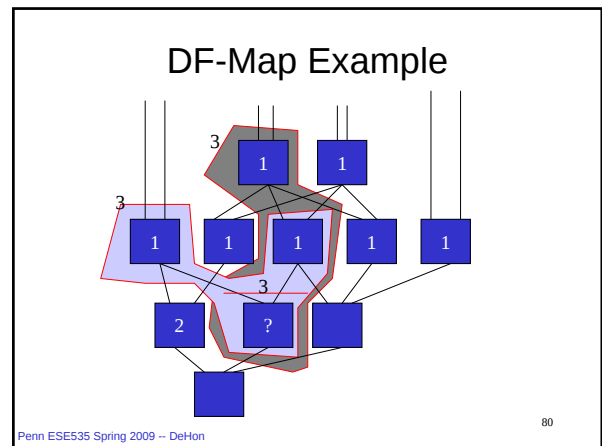
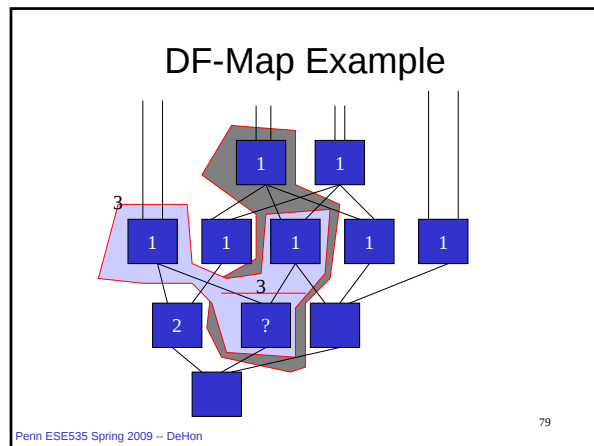
54

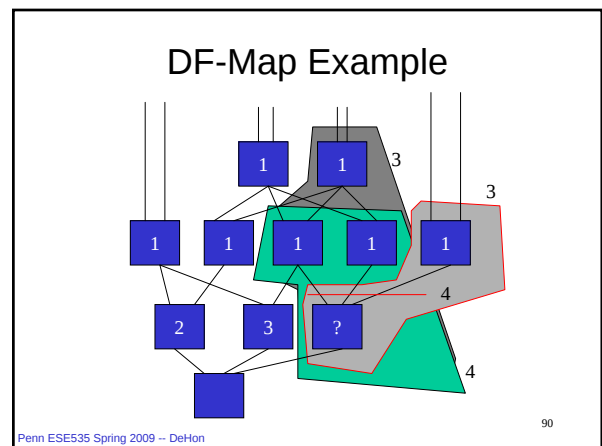
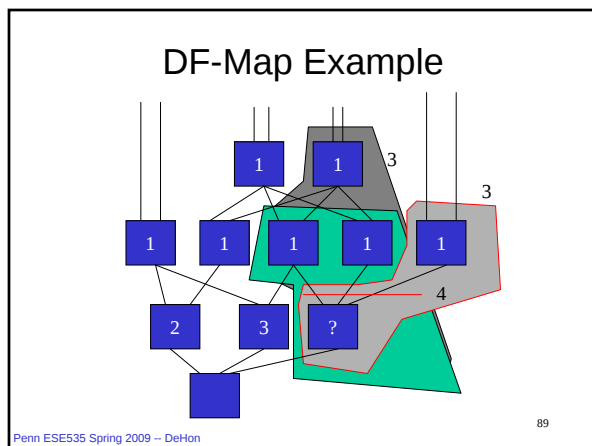
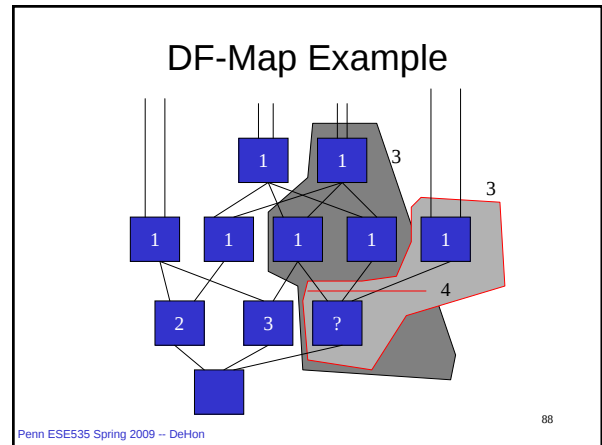
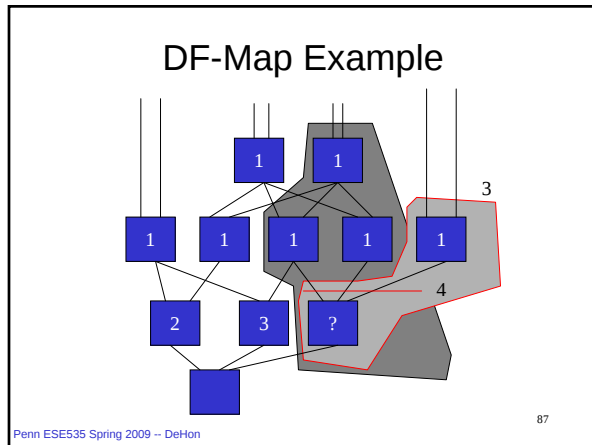
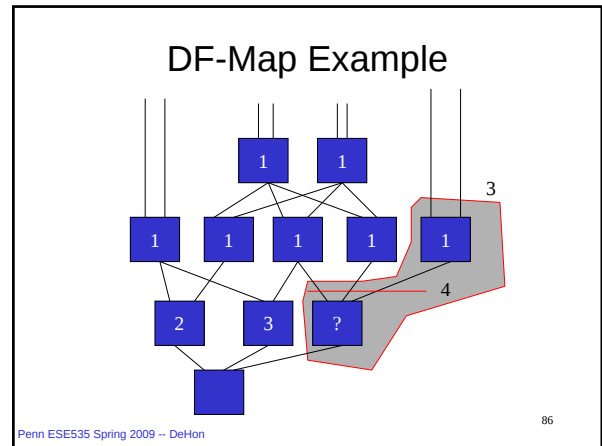
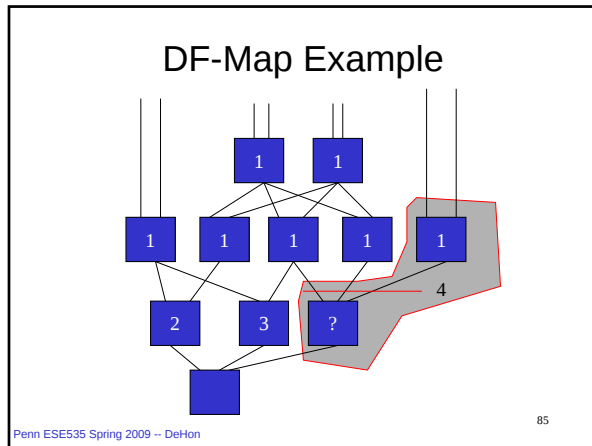


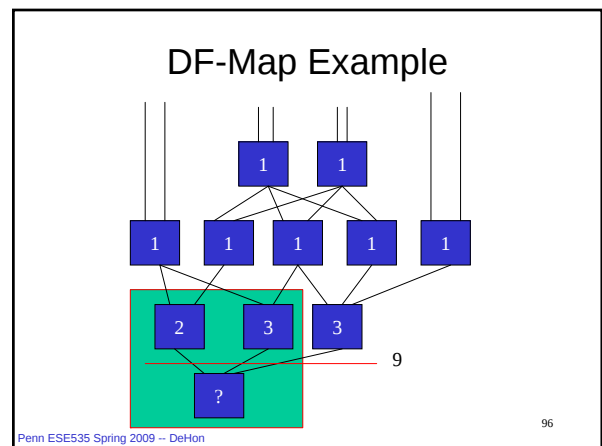
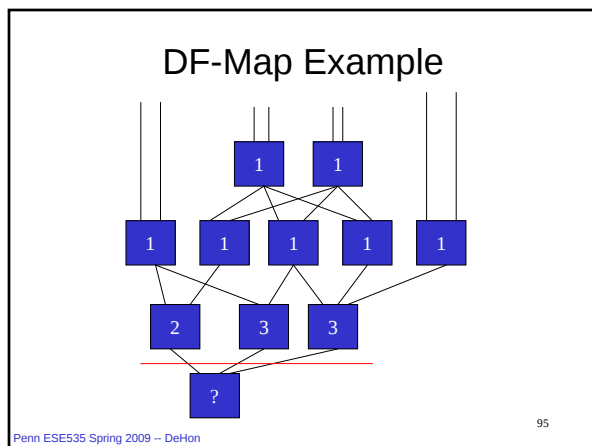
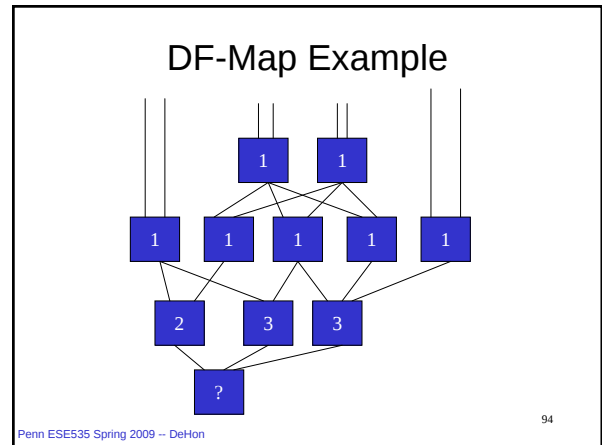
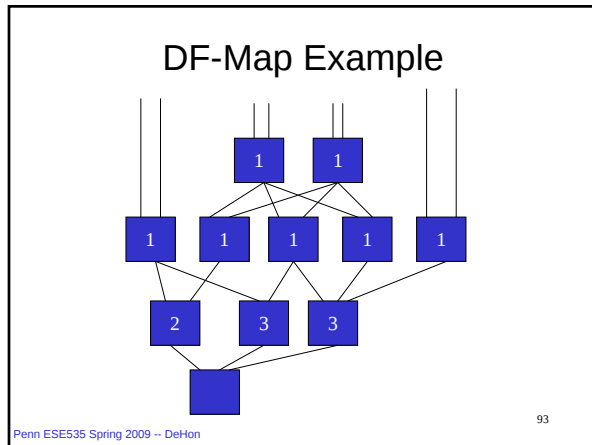
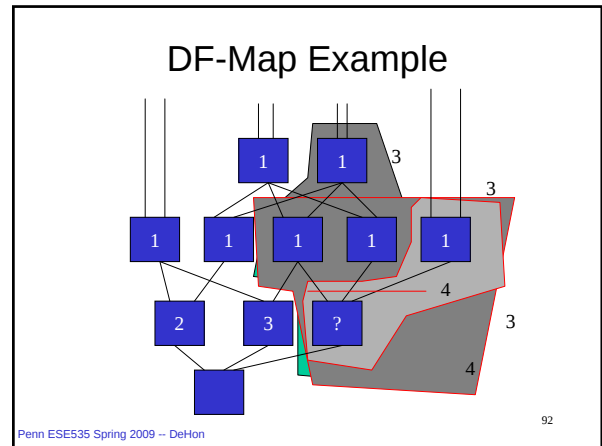
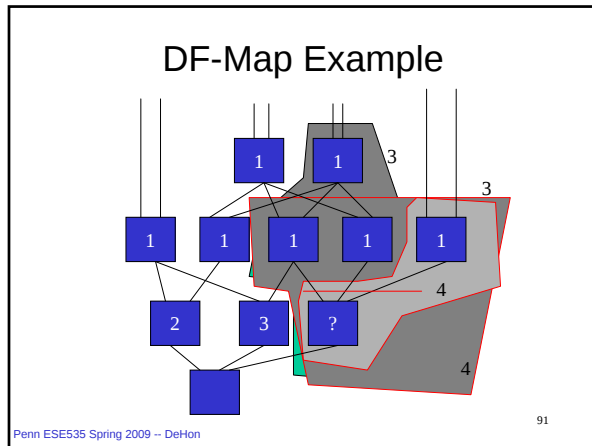


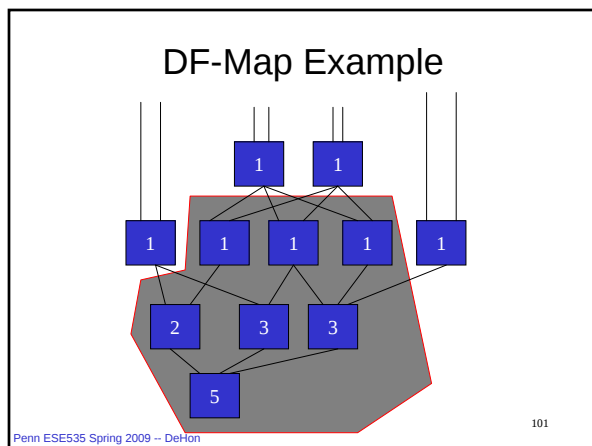
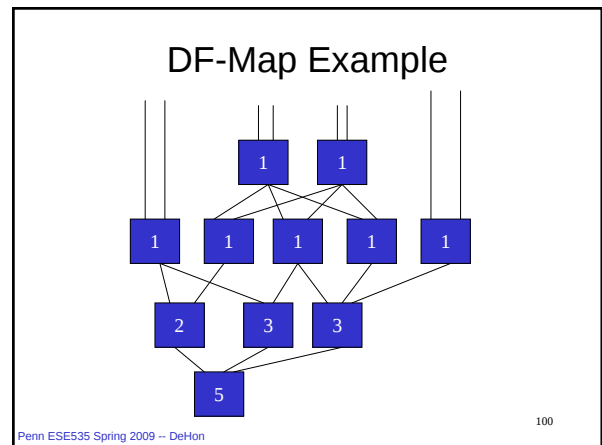
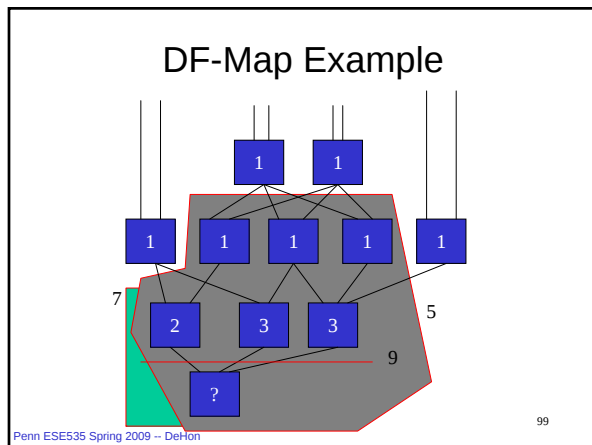
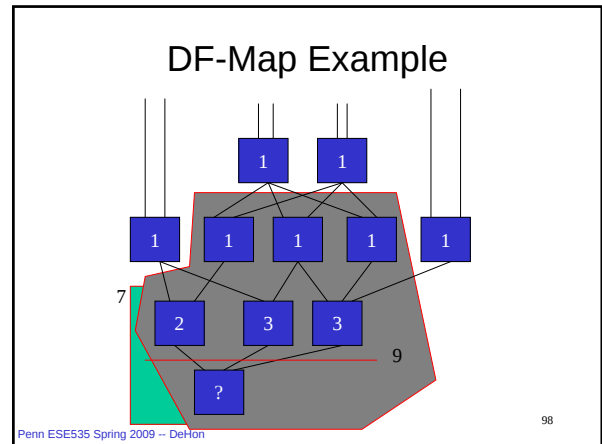
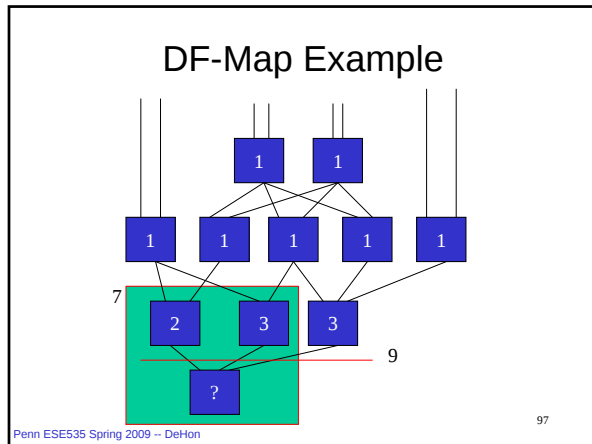












Composing

- Don't need minimum delay off the critical path
- Don't always want/need minimum delay
- Composite:
 - map with flowmap
 - Greedy decomposition of "most promising" non-critical nodes
 - DF-map these nodes

Penn ESE535 Spring 2009 -- DeHon

102

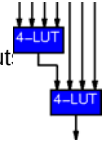
Variations on a Theme

Penn ESE535 Spring 2009 -- DeHon

103

Applicability to Non-LUTs?

- *E.g.* LUT Cascade
 - can handle some functions of K input
- How apply?



Penn ESE535 Spring 2009 -- DeHon

104

Adaptable to Non-LUTs

- Sketch:
 - Initial decomposition to nodes that will fit
 - Find max volume, min-height K-feasible cut
 - ask if logic block will cover
 - yes $\Rightarrow$ done
 - no $\Rightarrow$ exclude one (or more) nodes from block and repeat
 - exclude $\Rightarrow$ collapse into start set nodes
 - this makes heuristic

Penn ESE535 Spring 2009 -- DeHon

105

Partitioning?

- Effectively partitioning logic into clusters
 - LUT cluster
 - unlimited internal “gate” capacity
 - limited I/O (K)
 - simple delay cost model
 - 1 cross between clusters
 - 0 inside cluster

Penn ESE535 Spring 2009 -- DeHon

106

Partitioning

- Clustering
 - if strongly I/O limited, same basic idea works for partitioning to components
 - typically: partitioning onto multiple FPGAs
 - assumption: inter-FPGA delay $\gg$ intra-FPGA delay
 - w/ area constraints
 - similar to non-LUT case
 - make min-cut
 - will it fit?
 - Exclude some LUTs and repeat

Penn ESE535 Spring 2009 -- DeHon

107

Clustering for Delay

- W/ no IO constraint
- area is monotone property
- DP-label forward with delays
 - grab up largest labels (greatest delays) until fill cluster size
- Work backward from outputs creating clusters as needed

Penn ESE535 Spring 2009 -- DeHon

108

Area and IO?

- Real problem:
 - FPGA/chip partitioning
- Doing both optimally is NP-hard
- Heuristic around IO cut first should do well
 - (e.g. non-LUT slide)
 - [Yang and Wong, FPGA'94]

Penn ESE535 Spring 2009 -- DeHon

109

Partitioning

- To date:
 - primarily used for 2-level hierarchy
 - i.e. intra-FPGA, inter-FPGA
- Open/promising
 - adapt to multi-level for delay-optimized partitioning/placement on fixed-wire schedule
 - localize critical paths to smallest subtree possible?

Penn ESE535 Spring 2009 -- DeHon

110

Summary

- Optimal LUT mapping NP-hard in general
 - fanout, replication, ...
- K-LUTs makes delay optimal feasible
 - **single constraint:** IO capacity
 - **technique:** max-flow/min-cut
- Heuristic adaptations of basic idea to capacity constrained problem
 - promising area for interconnect delay optimization

Penn ESE535 Spring 2009 -- DeHon

111

Admin

- Reading for next time
 - Handout hardcopy
- Assignment 4
 - Hopefully done with problem 1 and 2
 - Problem 3 follows on today's lecture

Penn ESE535 Spring 2009 -- DeHon

112

Today's Big Ideas:

- IO may be a dominant cost
 - limiting capacity, delay
- Exploit structure: K-LUTs
- Mixing dominant modes
 - multiple objectives
- Define optimally solvable subproblem
 - duplication free mapping

Penn ESE535 Spring 2009 -- DeHon

113