Balancing Interconnect and Computation in a Reconfigurable Array

Why you don't really want 100% LUT utilization

Dr. André DeHon
BRASS Project
University of California at Berkeley

Question

How much interconnect do I need for my computing/programmable array?

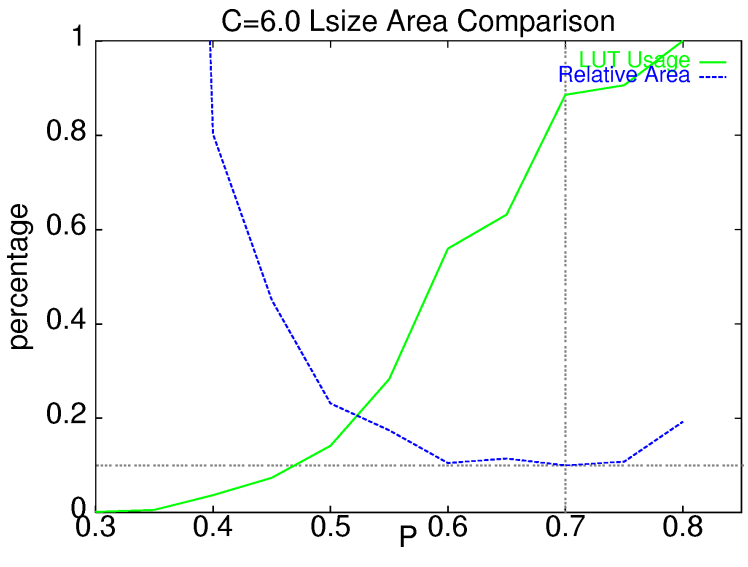
Problem(?): too little interconnect
→won't be able to use all the gates/LUTs

Typical subgoal: how much interconnect to use (almost) all LUTs?

Wrong Subgoal

- **Observation:**
  - interconnect is dominant area on FPGAs
  - more important to use interconnect efficiently than to use LUTs efficiently
- **Different question/subgoal:**
  - What level of interconnect gives the least implementation area for applications?

LUT Utilization predict Area?

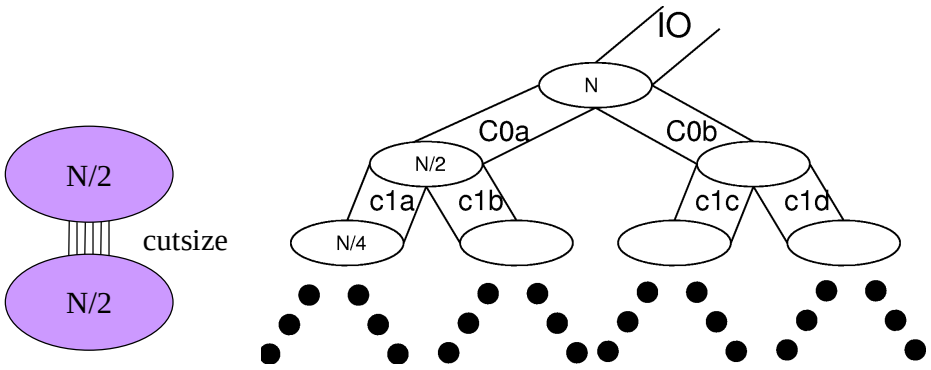


Outline

- Question: how much interconnect?
- Teaser: less than 100% LUT utilization
- Model
- Application characteristics
- Compose
- Conclusions

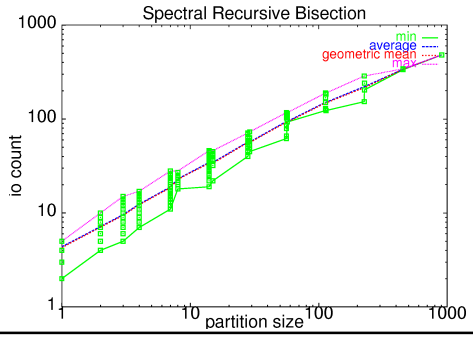
Model Interconnect Requirements and Richness

- Recursively partition (bisect) design
  - Look at I/O from each partition (subtree)



Regularizing Growth

- How do bisection bandwidths shrink (grow) at different levels of bisection hierarchy?
- **Basic assumption:** Geometric
  - 1
  - $1/\alpha$
  - $1/\alpha^2$



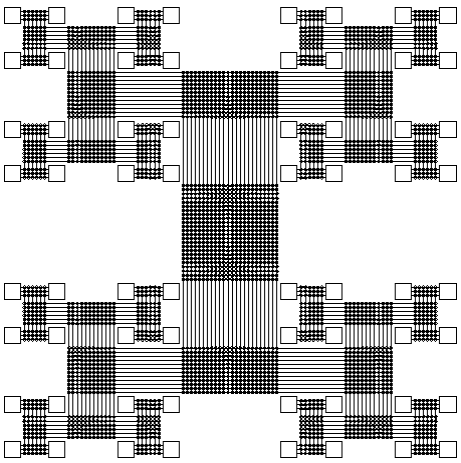
Rent's Rule

- Long standing **empirical** relationship
  - $IO = C \times N^P$
  - $0 \leq P \leq 1.0$
- Embodies geometric assumption (C,P)
  - Two parameters
    - C base of growth
    - P capture growth ($\alpha = 2^P$)
- Captures notion of locality

Step 1: Build Architecture Model

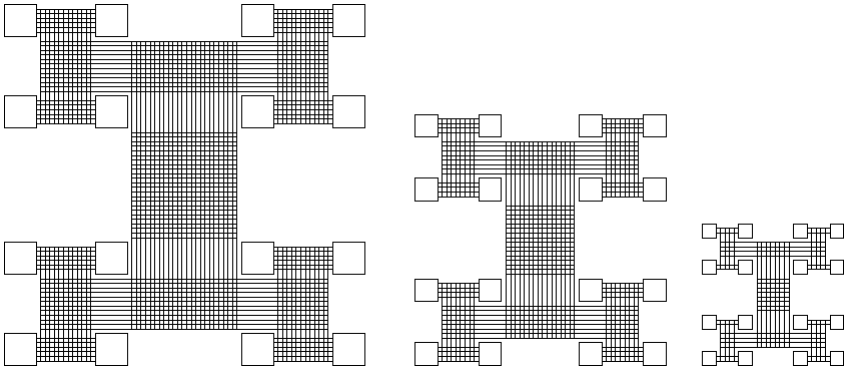
- Assume geometric growth
- Build architecture can tune
  - F, C
    α, p

Tree of Meshes

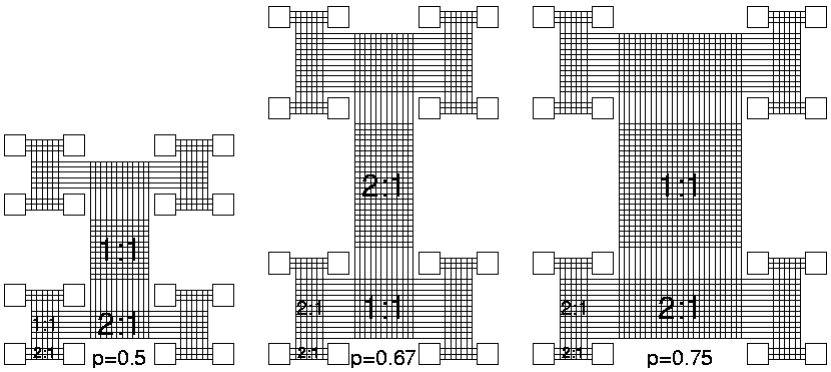


- Tree
- Restricted internal bandwidth
- Can match to model

Parameterize C



Parameterize Growth



(2 1)* => $\alpha=\sqrt{2}$

(2 2 2 1)* => $\alpha=2^{(3/4)}$

(2 2 1)* => $\alpha=(2*2)^{(1/3)}=2^{(2/3)}$

Step 2: Area Model

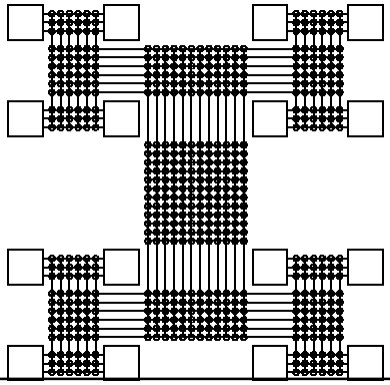
- Need to know effect of architecture parameters on area (costs)
  - focus on dominant components
    - wires (saw on Thursday)
    - switches
    - logic blocks(?)

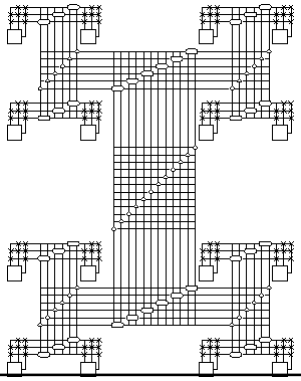
Area Parameters

- $A_{logic} = 40K\lambda^2$
- $A_{sw} = 2.5K\lambda^2$
- Wire Pitch = 8λ

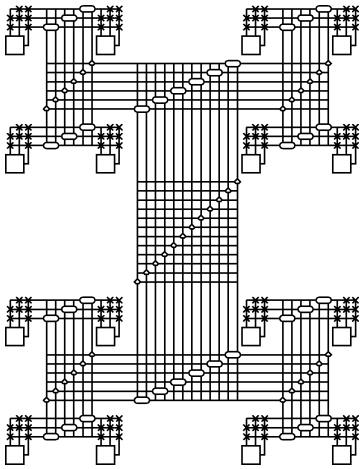
Switchbox Population

- Full population is excessive (see next week)
- Hypothesis: linear population adequate
  - still to be (dis)proven

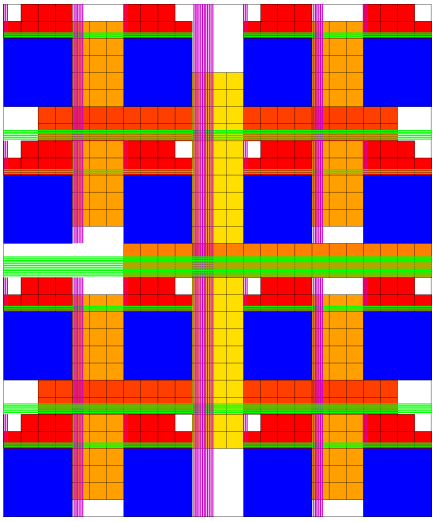




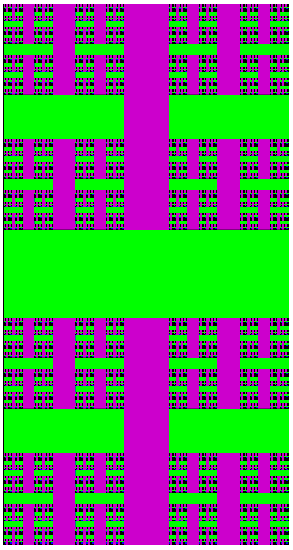
“Cartoon” VLSI Area Model



(Example artificially small for clarity)



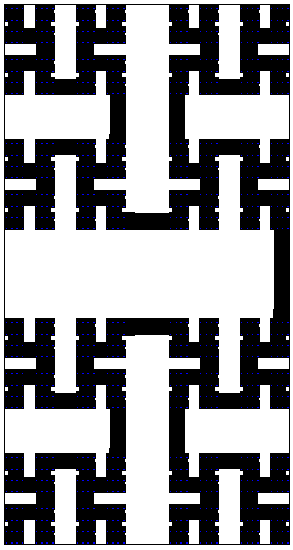
Larger “Cartoon”



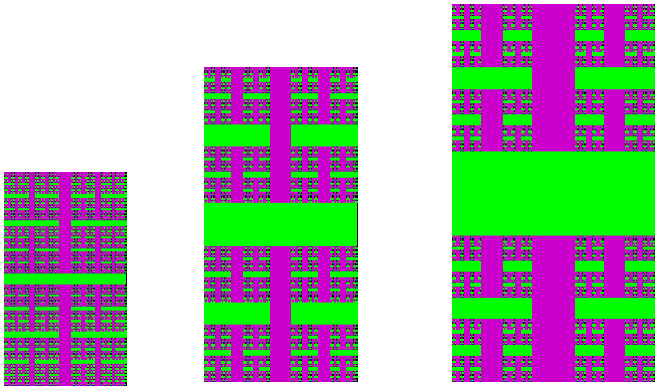
1024 LUT Network

P=0.67

LUT Area 3%



Effects of P on Area



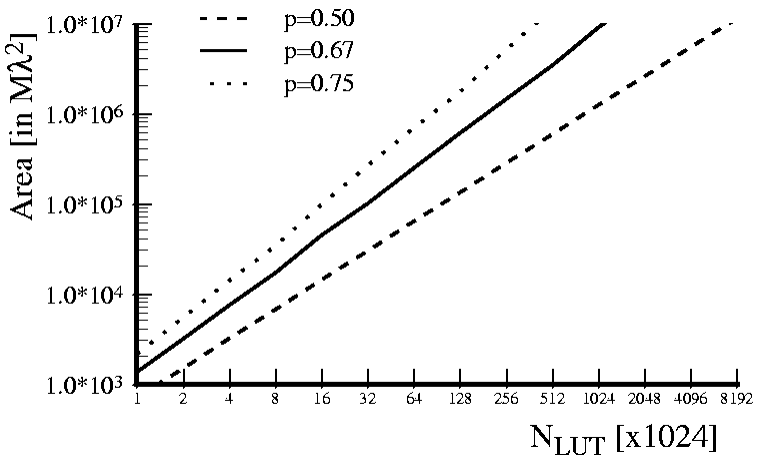
0.25
P=0.5

0.37
P=0.67

1.00
P=0.75

1024 LUT Area Comparison

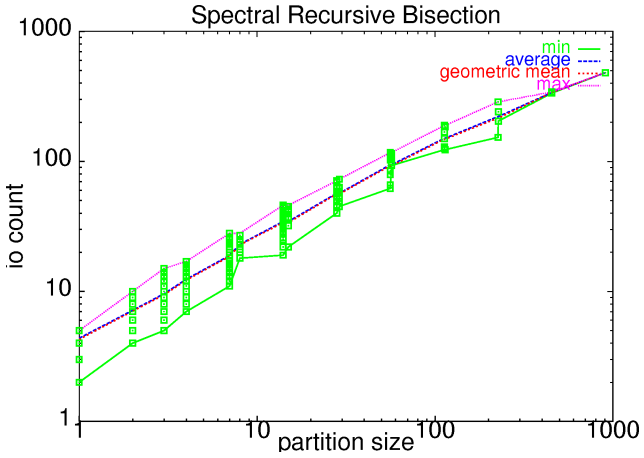
Effects of P on Capacity



Step 3: Characterize Application Requirements

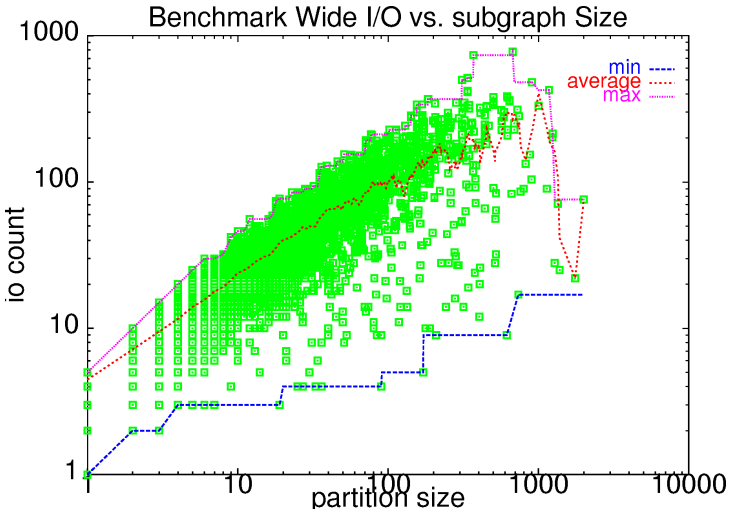
- Identify representative applications.
  - Today: IWLS93 logic benchmarks
- How much structure there?
- How much variation among applications?

Application Requirements

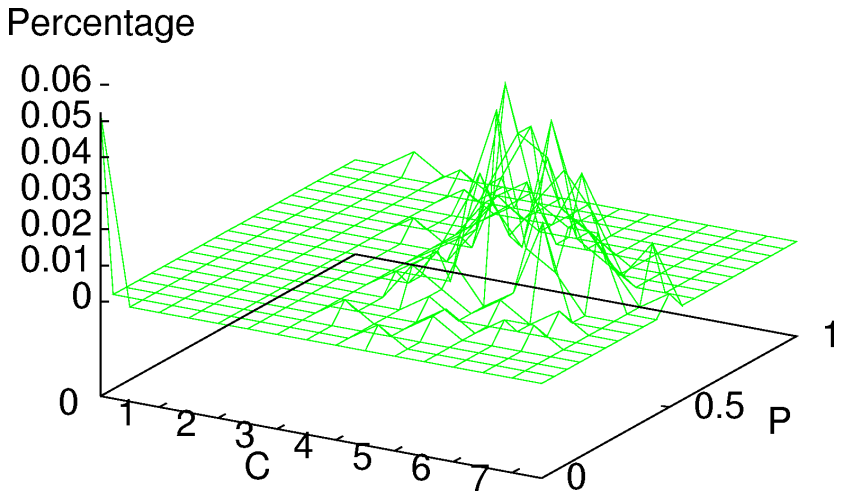


Max: C=7, P=0.68 Avg: C=5, P=0.72

Application Requirements: Benchmark Wide (MCNC)



Benchmark Parameters



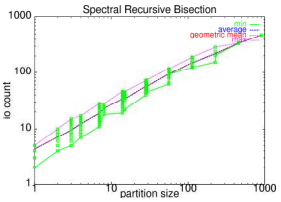
Interconnect requirements vary across applications.

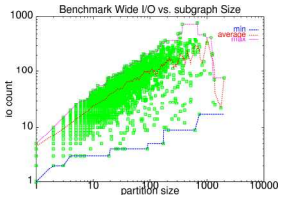
Complication

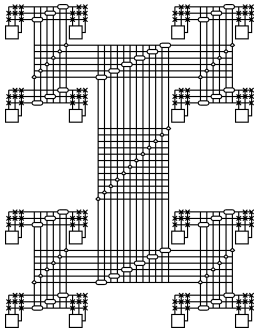
- Interconnect requirements vary among applications
- Interconnect richness has large effect on area
- What is effect of architecture/application mismatch?
  - Interconnect too rich?
  - Interconnect too poor?

Network Fixed Schedule

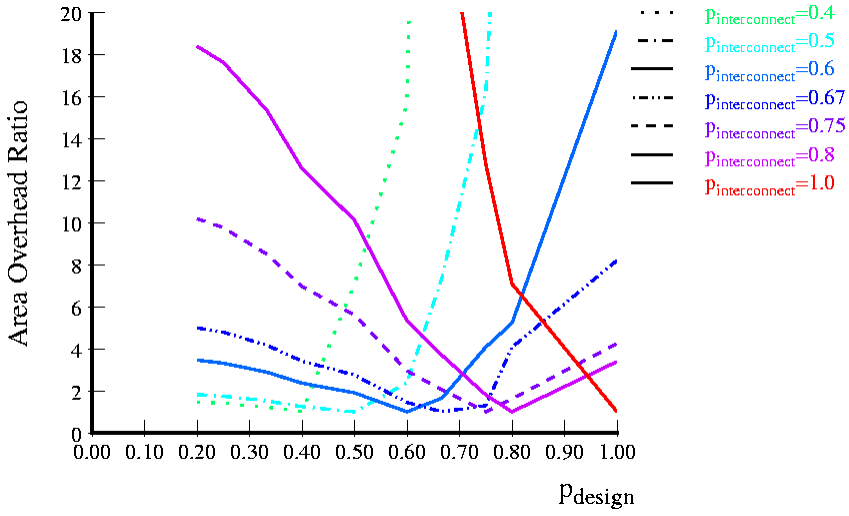
- Network will have a fixed wiring schedule
- Applications have varying requirements
- To assess impact of mismatch
  - map to network schedules
  - look at area required







Interconnect Mismatch in Theory

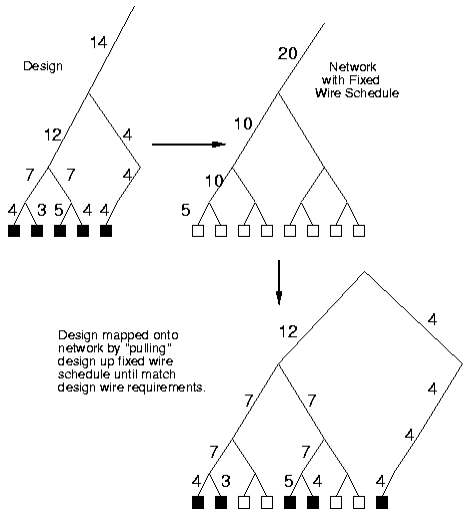


Step 4: Assess Resource Impact

- Map designs to parameterized architecture
- Identify architectural resource required

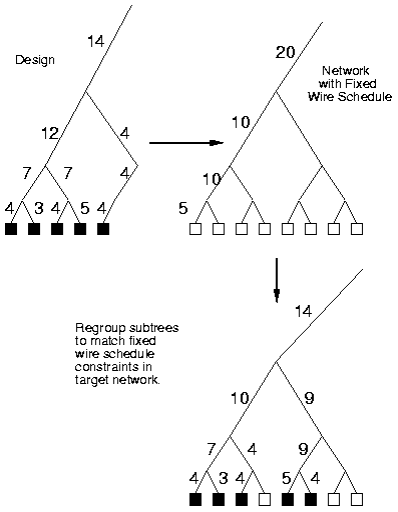
Mapping to Fixed Wire Schedule

- Easy if need less wires than Net
- If need more wires than net, must depopulate to meet interconnect limitations.



Mapping to Fixed-WS

- Better results if "reassociate" rather than keeping original subtrees.



Observation

- Don't really want a "bisection" of LUTs
  - subtree filled to capacity by either of
    - LUTs
    - root bandwidth
  - May be profitable to cut at some place other than midpoint
    - not require "balance" condition
  - "Bisection" should account for both LUT and wiring limitations

Challenge

- Not know where to cut design into
  - not knowing when wires will limit subtree capacity

Brute Force Solution

- Explore all cuts
  - start with all LUTs in group
  - consider “all” balances
  - try cut
  - recurse

Brute Force

- Too expensive
- Exponential work

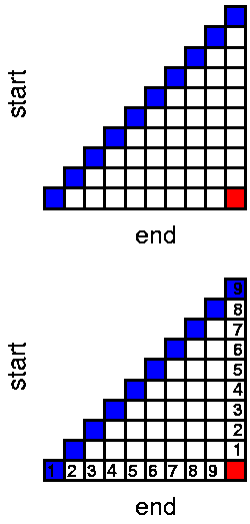
- …viable if solving same subproblems

Simplification

- Single linear ordering
- Partitions = pick split point on ordering
- Reduce to finding cost of [start,end] ranges (subtrees) within linear ordering
- Only n^2 such subproblems
- Can solve with dynamic programming

Dynamic Programming

- Start with base set of size 1
- Compute all splits of size n, from solutions to all problems of size n-1 or smaller
- Done when compute where to split 0,N-1



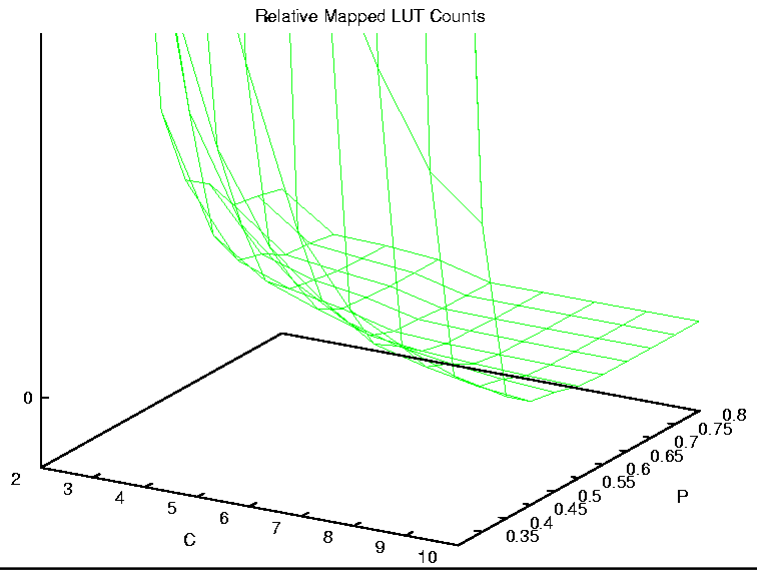
Dynamic Programming

- Just one possible “heuristic” solution to this problem
  - not optimal
  - dependent on ordering
  - sacrifices ability to reorder on splits to avoid exponential problem size
- Opportunity to find a better solution here...

Ordering LUTs

- Another problem
  - lay out gates in 1D line
  - minimize sum of squared wire length
    - tend to cluster connected gates together
  - Is solvable mathematically for optimal
    - Eigenvector of connectivity matrix
- Use this 1D ordering for our linear ordering

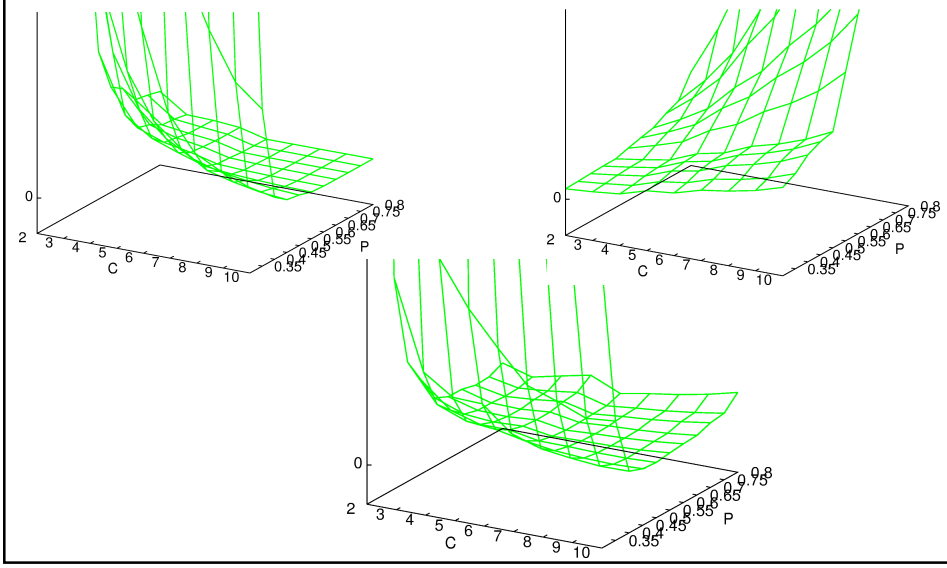
Mapping Results



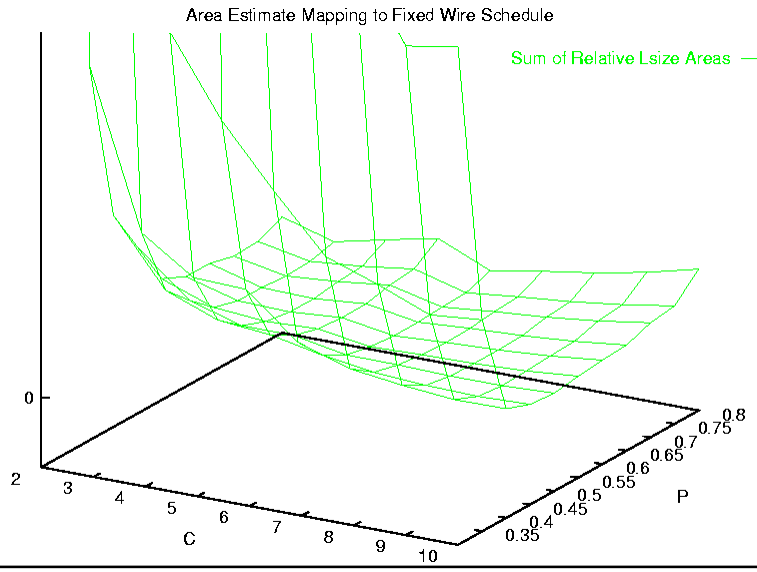
Step 5: Apply Area Model

- Assess impact of resource results

Resources × Area Model => Area



Net Area

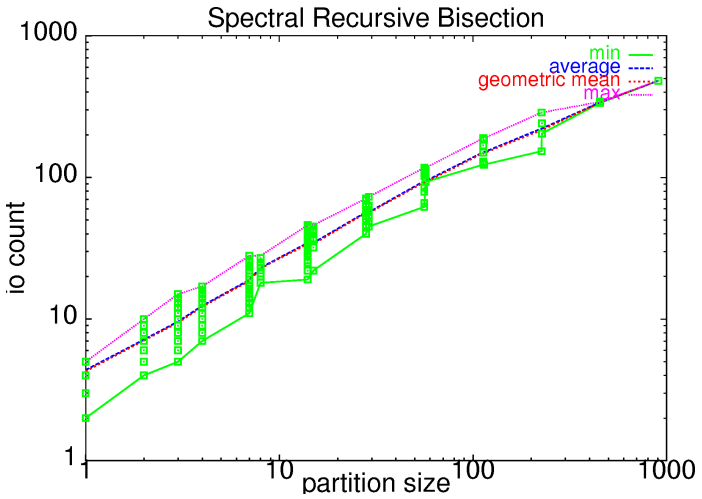


Picking Network Design Point

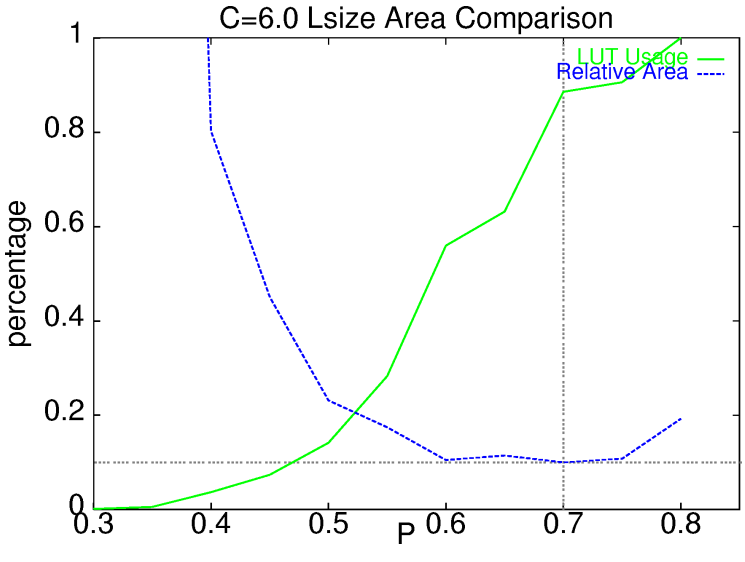
Linear $WP = 8\lambda$, $A_{sw} = 2500\lambda^2$

Minimize Objective	params C	P	Sigma rel area	Max rel area	LUT Util.
relative area	6	0.6	1.23	2.84	0.87
max relative area	6	0.65	1.24	2.38	0.89
area with full util	10	0.75	2.98	4.87	1.00

What about a single design?



LUT Utilization predict Area?



Summary

- Interconnect area dominates
  logic block area
- Interconnect requirements vary
  - among designs
  - within a single design
- To minimize area
  - focus on using dominant resource (interconnect)
  - may underuse non-dominant resources (LUTs)

Methodology

- Architecture model (parameterized)
- Cost model
- Important task characteristics
- Mapping Algorithm
  - Map to determine resources
- Apply cost model
- Digest results
  - find optimum (multiple?)
  - understand conflicts (avoidable?)