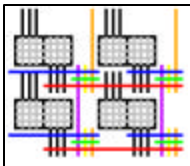


Interconnect

André DeHon
<andre@cs.caltech.edu>

Thursday, June 20, 2002

Physical Entities

- **Idea:** Computations take up space
  - Bigger/smaller computations
  - Size→resources→cost
  - Size→distance→delay

Impact

- Consequence is:
  - Properties of the physical world ultimately affect our computations
    - Delay = Distance / Speed
    - Scattering, mean-free-path
    - Thermodynamics (reversibility, kT,…)

Interconnect

- Perhaps nowhere is this more present than in **interconnect**
  - Speed of light delay
  - Finite size of devices
    - Ultimate limits (Feynman's "Bottom")
    - What we can pattern and control today
    - How well we can localize phenomena (tunneling)
  - Area and geometry of wires

Today

- Interconnect
- Wires and VLSI
- Dominance of Interconnect
- Implications for physical computing systems

Physical Interconnect

- Anything that allows one physical component of the computer to communicate with another
  - **Wires** that connect transistors or gates
  - **Traces** on printed circuit boards that connect components
  - **Cables** and backplanes that connect boards
  - Ethernet and video **cables** that connect workstations, switches, and IO
  - **Fibers** that connect our building routers

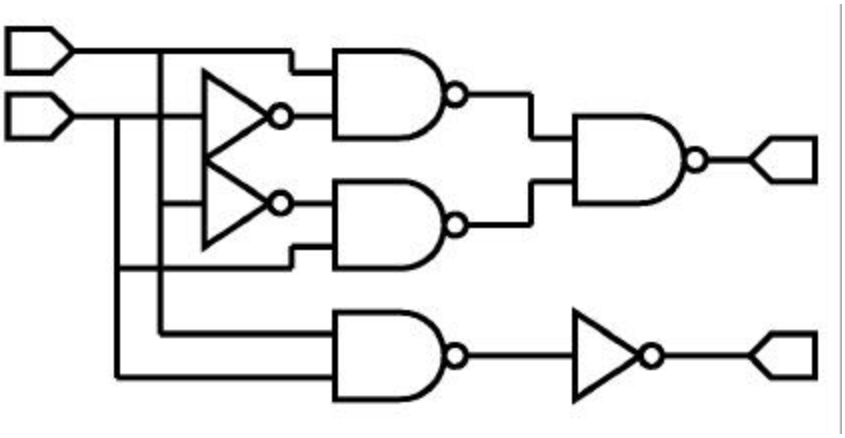
Interconnect

- Today, let's concentrate on
  - **gates** and **wires**
- Modern component contains millions of gates (*e.g.* 2-input **nor** gate)
- Each gate takes up finite space
- To work together, these gates need to communicate with each other
  - Need wires for interconnect

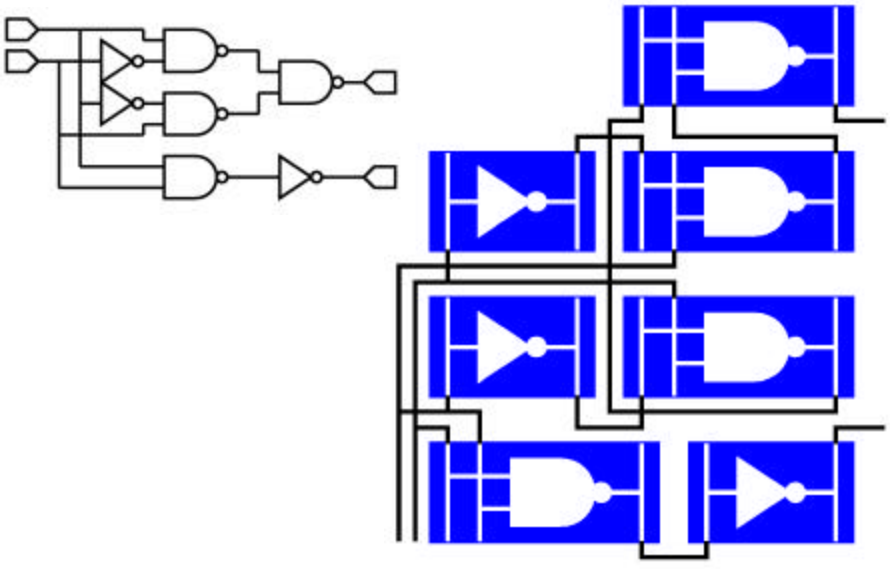
Last Time

- We saw that
  - Modest size programmable gates
  - Connected by programmable interconnect
- Are more efficient than
  - Tiny programmable gates
  - Large LUTs
- Even though the interconnect may take up most of the area!

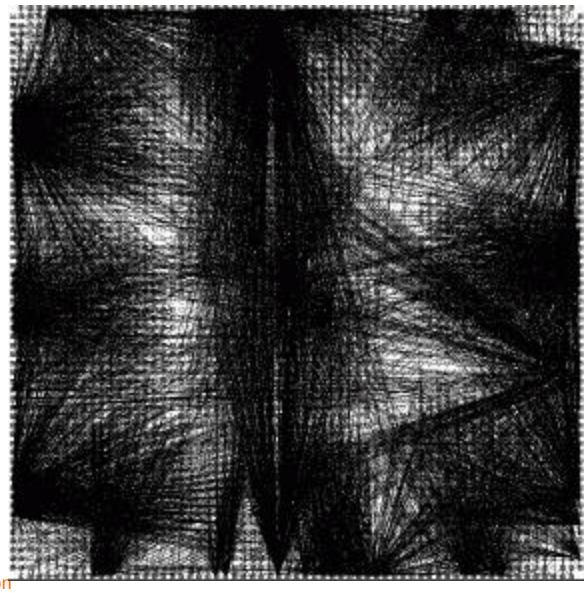
Small Example



Physical Layout



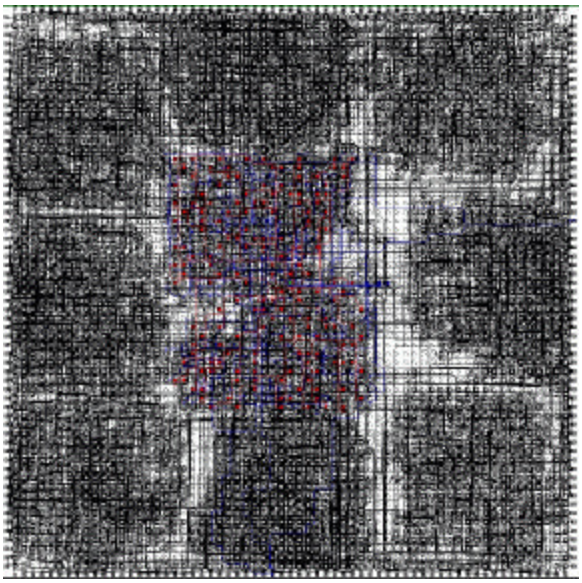
Larger Example



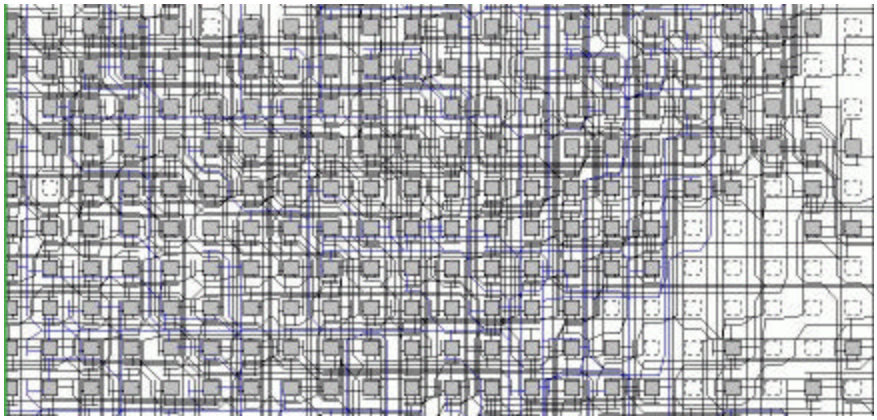
DES Circuit

Larger Example (DES) Routed

Must find place for all those wires.



Closeup (DES Routed)



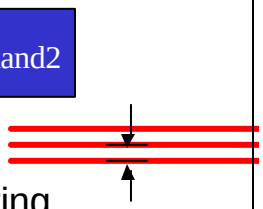
Wires can take up significant space.

Claim

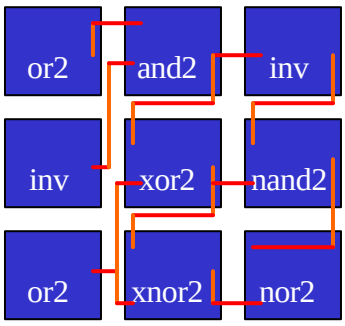
- For
  - Sufficiently large computations
  - "arbitrary" design (and many particular)
  - with finite size wires
- Area associated with **interconnect** will **dominate** that required for **gates.**
  - Natural consequence of physical geometry in two-dimensional space
    - (any finite dimensions)

Wires and VLSI

- Simple VLSI model
  - Gates have fixed size (A_{gate})
  - Wires have finite spacing (W_{wire})
  - Have a small, finite number of wiring layers
    - *E.g.*
      - one for horizontal wiring
      - one for vertical wiring
  - Assume wires can run over gates



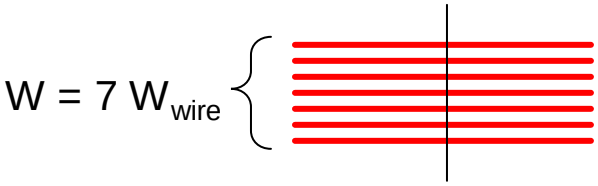
Visually: Wires and VLSI



Important Consequence

- A set of wires
- crossing a line
- take up space:

$$W = (N \times W_{wire}) / N_{layers}$$



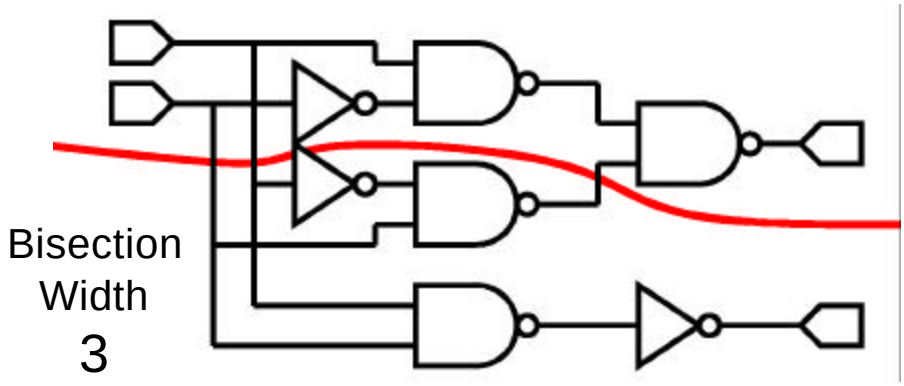
Thompson's Argument

- The minimum area of a VLSI component is bounded by the larger of:
  - The area to hold all the gates
    - $A_{chip} \geq N \times A_{gate}$
  - The area required by the wiring
    - $A_{chip} \geq N_{horizontal}\ W_{wire} \times N_{vertical}\ W_{wire}$

How many wires?

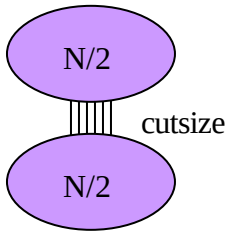
- We can get a **lower bound** on the total number of horizontal (vertical) wires by considering the **bisection** of the computational graph:
  - Cut the graph of gates in half
  - Minimize connections between halves
  - Count number of connections in cut
  - Gives a lower bound on number of wires

Bisection



Next Question

- In general, if we:
  - Cut design in half
  - Minimizing cut wires
- How many wires will be in the bisection?



Arbitrary Graph

- Graph with N nodes
- Cut in half
  - N/2 gates on each side
- **Worst-case:**
  - Every gate output on each side
  - Is used somewhere on other side
  - Cut contains N wires

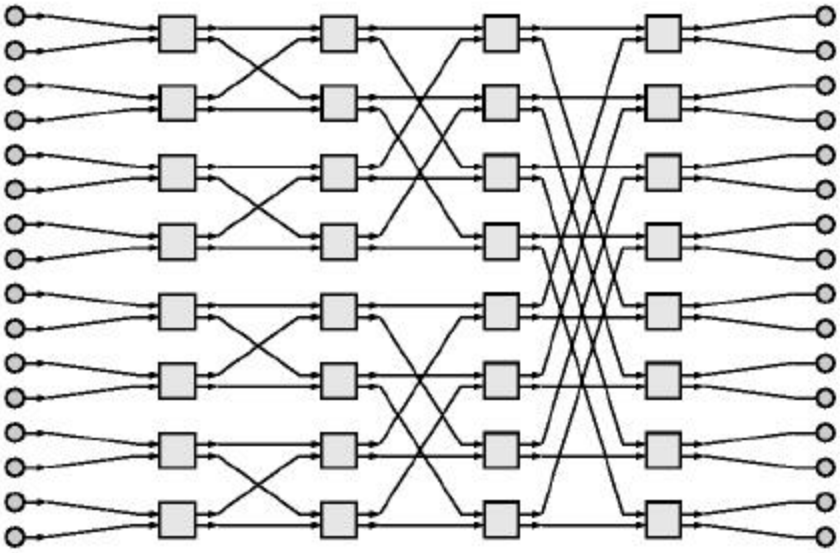
Arbitrary Graph

- For a random graph
  - Something proportional to this is likely
- That is:
  - Given a random graph with N nodes
  - The number of wires in the bisection is likely to be: $c \times N$

Particular Computational Graphs

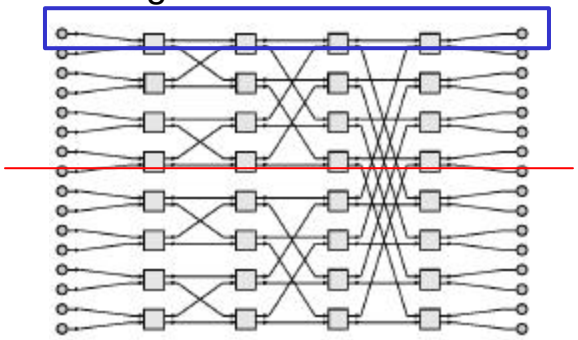
- Some important computations have exactly this property
  - FFT (Fast Fourier Transform)
  - Sorting

FFT



FFT

- Can implement with N/2 nodes
  - Group row together
- Any bisection will cut N/2 wire bundles
  - True for any reordering



Assembling what we know

- $A_{chip} \geq N \times A_{gate}$
- $A_{chip} \geq N_{horizontal}\ W_{wire} \times N_{vertical}\ W_{wire}$
- $N_{horizontal} = c \times N$
- $N_{vertical} = c \times N$
  - [bound true recursively in graph]
- $A_{chip} \geq cN\ W_{wire} \times c\ N\ W_{wire}$

Assembling …

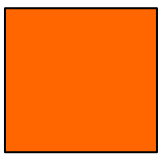
- $A_{chip} \geq N \times A_{gate}$
- $A_{chip} \geq cN\ W_{wire} \times cN\ W_{wire}$
- $A_{chip} \geq (cN\ W_{wire})^2$
- $A_{chip} \geq N^2 \times c'$

Result

- $A_{chip} \geq N \times A_{gate}$
- $A_{chip} \geq N^2 \times c'$
- Wire area grows faster than gate area
- Wire area grows with the square of gate area
- For sufficiently large N,
  - Wire area dominates gate area

Intuitive Version

- Consider a region of a chip
- Gate capacity in the region goes as area (s^2)
- Wiring capacity into region goes as perimeter (4s)
- Perimeter grows more slowly than area
  - Wire capacity saturates before gate



Result

- $A_{chip} \geq N^2 \times c'$
- Wire area grows with the square of gate area
- Troubling:
  - To **double** the size of our computation
  - Must **quadruple** the size of our chip!

Interlude

Miles of Wire

- Consider FPGA
  - Programmable Gate Arrays
  - Today providing ~1 Million gate capacity devices
- "What we really sell is miles of wiring."
  - Clive McCarthy (Altera) circa 1998
- 15mm die ×15mm/0.5μm wire spacing
- (450m/layer) × 5 layers > 2 km

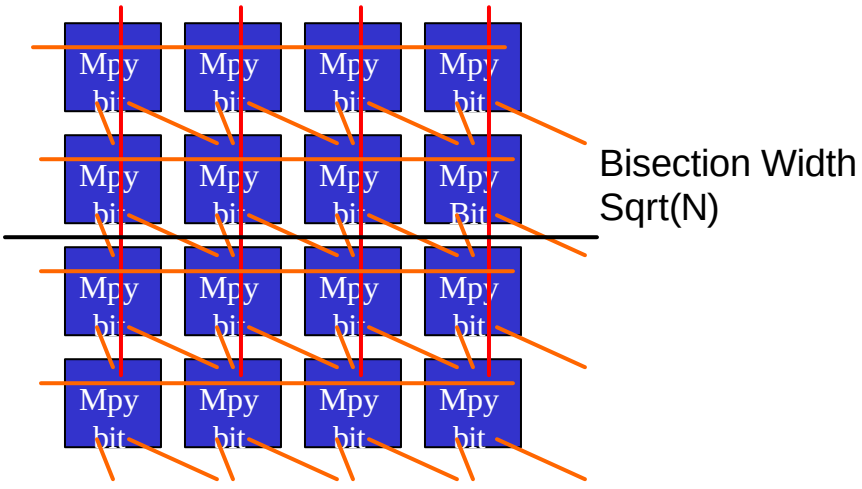
So what?

What do we do with this observation?

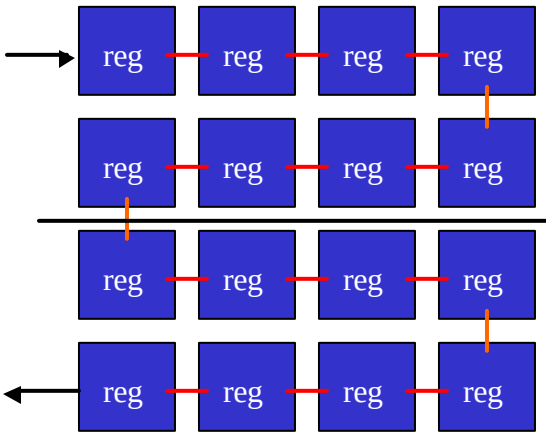
First Observation

- Not all designs have this large of a bisection
- Architecture is about understanding structure
- What is typical?

Array Multiplier



Shift Register



Bisection Width

- Trying to assess wiring or total area requirements on gates alone is short sighted.
  - But most people try to do this…
- **Bisection width** is an important, first order property of a design.

Rent's Rule

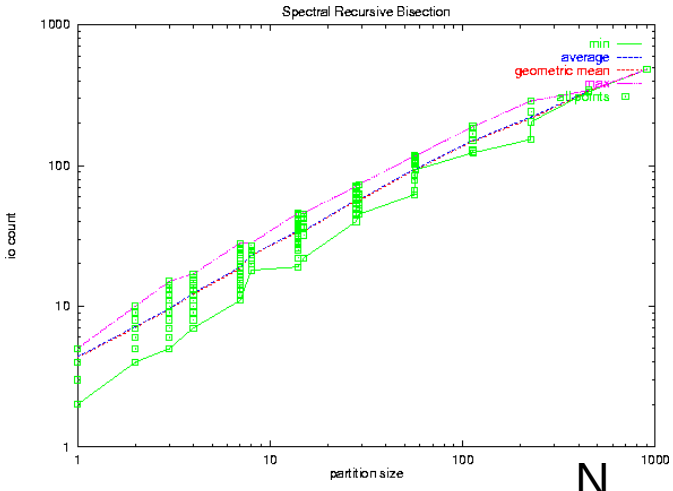
- In the world of circuit design, an empirical relationship to capture:

$$IO = c\,N^{p}$$

- $0 \le p \le 1$
- p – characterizes interconnect richness
- Typical: $0.5 \le p \le 0.7$
- "High-Speed" Logic p=0.67

Empirical Characterization of Bisection

IO



N

Log-log plot

C=7
P=0.68

Fit:
$IO=cN^{p}$

As a function of Bisection

- $A_{chip} \geq N \times A_{gate}$
- $A_{chip} \geq N_{horizontal}\ W_{wire} \times N_{vertical}\ W_{wire}$
- $N_{horizontal} = N_{vertical} = IO = cN^{p}$
- $A_{chip} \geq (cN)^{2p}$
- If p<0.5

$$A_{chip} \propto N$$

- If p>0.5

$$A_{chip} \propto N^{2p}$$

In terms of Rent's Rule

- If p<0.5, $A_{chip} \propto N$
- If p>0.5, $A_{chip} \propto N^{2p}$

- **Typical** designs have **p>0.5**

  $\rightarrow$ interconnect dominates

Programmable Machine Impact

Design of

Multiprocessors, FPGAs…

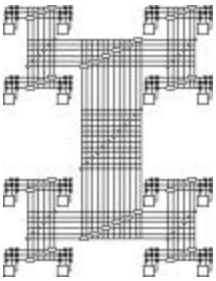
Impact on Programmables?

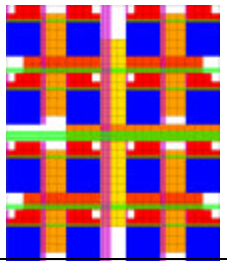
- What does this mean for our programmable devices?
  - Devices which may solve any problem?
  - *E.g.* multiprocessors, FPGAs
- Do we design for worst case?
  - Put N^2 area into interconnect
  - And guarantee can use all the gates?
- Or design to use the wires?
  - Wasting gates (processors) as necessary?

Interconnect: Experiment

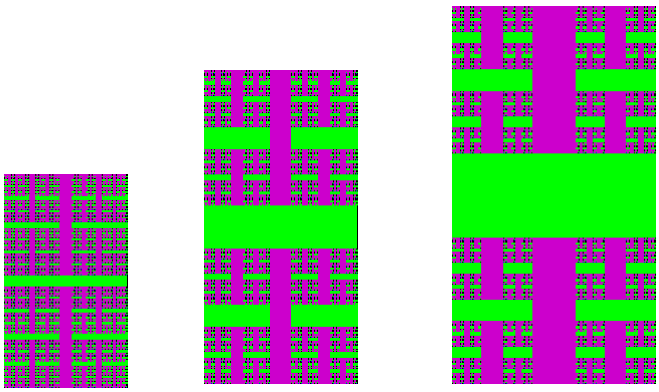
- Parameterizable network
  - tree of meshes/fat-
    bisection bw = Cn^P
  - bisection bw = Cn^P
- VLSI area model
- Mapping procedure
- Benchmark set
  - MCNC 4-LUT mapped

Details: FPGA'99





Effects of P on Area



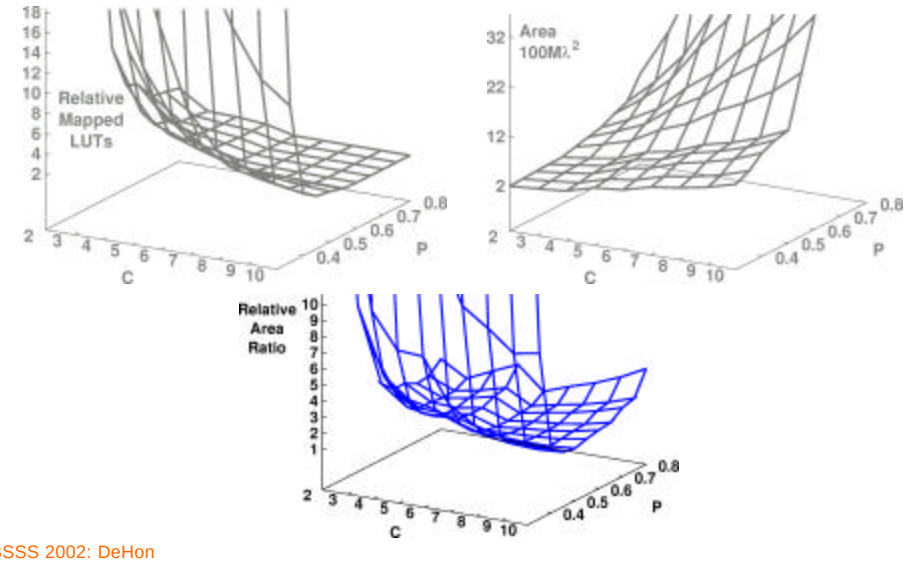
0.25
P=0.5

0.37
P=0.67

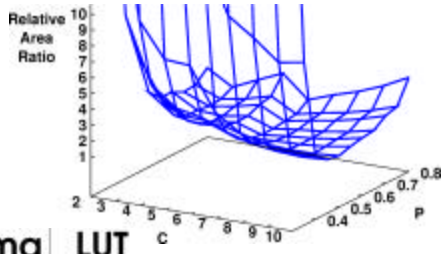
1.00
P=0.75

1024 LUT Area Comparison

Resources × Area Model ⇒ Area



Picking Network Design Point



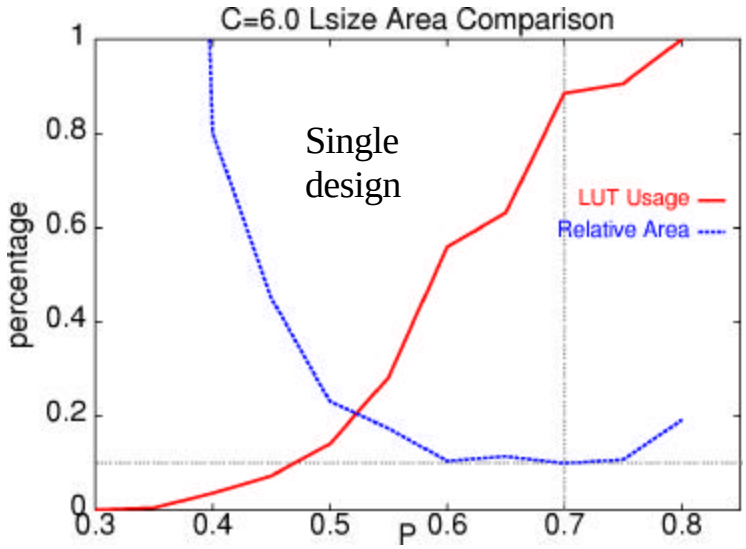
Minimize Objective	params C	P	Sigma rel area	LUT Util.
relative area	6	0.6	1.23	0.87
area with full util	10	0.75	2.98	1.00

Must provide reasonable level of interconnect;
…but don't guarantee 100% compute utilization.

Single Design

- Previous is for a set of designs
- What about a single design?
  - Do we minimize the area by providing enough wires to use all the gates for that single design?

Gate Utilization predict Area?



Consequences

- Even for a single design
  - We do not, necessarily, win by maximizing gate utilization
  - Are better off focusing on efficiently using the wires
    - Focus on using the **most expensive** resource!

Key Ideas

- Matter Computes
  - our computing machines are built out of physical phenomena
  - physical effects ultimately determine landscape for computations
- Interconnect requirements may dominate all other requirements
  - Compute, memory
  - Direct consequence of physical properties
- Efficient computations
  - May waste gates (compute) to use wires efficiently and minimize total area

Admin

- Project Discussion
  - 4:30pm here
  - Pitch projects, discuss ideas