CS184a: Computer Architecture (Structures and Organization)

Day11: October 30, 2000

Interconnect Requirements

Last Time

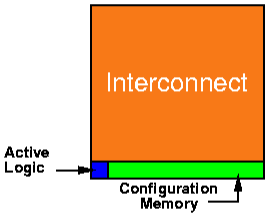
- Saw various compute blocks
- Role of automated mapping in exploring design space
- To exploit structure in typical designs we need programmable interconnect
- All reasonable, scalable structures:
  - small to moderate sized logic blocks
  - connected via programmable interconnect
- been saying delay across programmable interconnect is a big factor

Today

- Interconnect Design Space
- Dominance of Interconnect
- Simple things
  - and why they don't work
- Interconnect Implications
- Characterizing Interconnect Requirements

Dominant Area

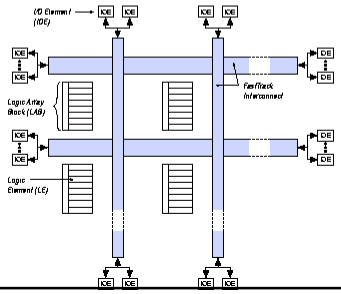
$$A_{bit\_elm} = A_{fixed} + \underbrace{N_{SW}(N_p, w, p) \cdot A_{SW}}_{\text{interconnect}} + \underbrace{\left(\frac{c}{w}\right) \cdot n_{ibits} \cdot A_{mem\_cell}}_{\text{instruction memory}} + \underbrace{d \cdot A_{mem\_cell}}_{\text{retiming memory}}$$



Function	Area (λ^2)
LUT MUX + ff	20K (generous, closer to 10K)
Programming Memory	80K (240K typical unencoded)
Interconnect	700K (for $N_p = 2048$)

Dominant Time

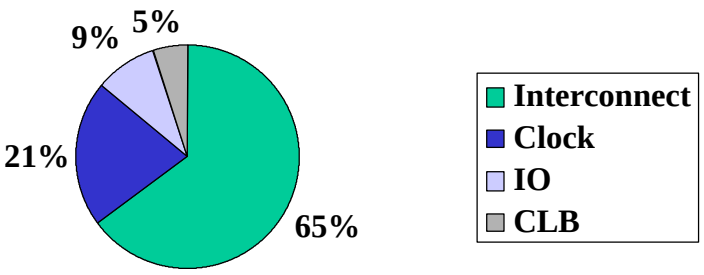
Design	Path	Total Delay	LUT Delay	Inter. %
Altera 10K130V-2	LUT-local-LUT	2.5 ns	2.1 ns	16%
	LUT-row-local-LUT	6.6 ns	2.1 ns	68%
	LUT-column-local-LUT	11.1 ns	2.1 ns	81%
	LUT-row-column-local-LUT	15.6 ns	2.1 ns	87%
	LUT-row-fanout-local-LUT (fanout)	28 ns	2.1 ns	90%



Dominant Time

Design	Path	Total Delay	LUT Delay	Inter. %
DPGA	LUT-LUT (in subarray)	3.5 ns	1.5 ns	60%
	LUT-xbar-LUT	7 ns	1.5 ns	80%
HSRA	LUT-LUT	8 ns	<2 ns	25%
	LUT-cascade	4 ns/4	<2 ns	0%
	LUT-4tree-LUT	8 ns	<2 ns	80%
	LUT-8tree-LUT	12 ns	<2 ns	83%
	LUT-16tree-LUT	16 ns	<2 ns	88%
	LUT-64tree-LUT	20 ns	<2 ns	90%

Dominant Power



XC4003A data from Eric Kusse (UCB MS 1997)

For Spatial Architectures

- Interconnect dominant
  - area
  - power
  - time
- …so need to understand in order to optimize architectures

Interconnect

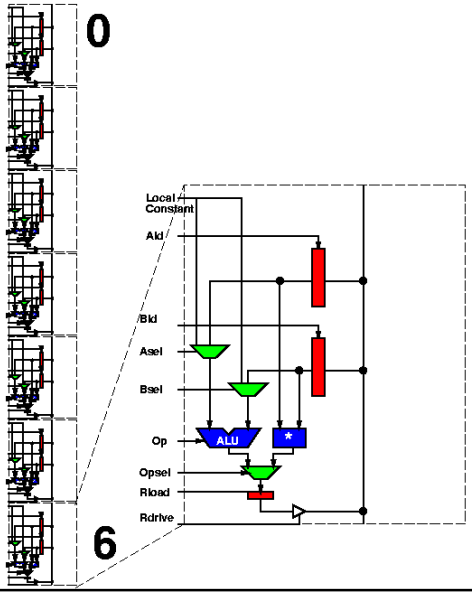
- **Problem**
  - Thousands of independent (bit) operators producing results
    - true of FPGAs today
    - …true for *LIW, multi-uP, etc. in future
  - Each taking as inputs the results of other (bit) processing elements
  - Interconnect is **late bound**
    - don't know until after fabrication

Design Issues

- **Flexibility** -- route "anything"
  - (w/in reason?)
- **Area** -- wires, switches
- **Delay** -- switches in path, stubs, wire length
- **Power** -- switch, wire capacitance
- **Routability** -- computational difficulty finding routes

(1) Shared Bus

- Familiar case
- Use single interconnect resource
- Reuse in Time
- Consequence?



Shared Bus

- Consider operation: $y=Ax^2+Bx+C$
  - 3 mpys
  - 2 adds
  - ~5 values need to be routed from producer to consumer
- Performance lower bound if have design w/:
  - m multipliers
  - u madd units
  - a adders
  - i simultaneous interconnection busses

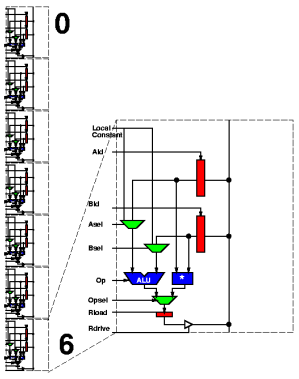
Viewpoint

- Interconnect is a resource
- Bottleneck for design can be in availability of any resource
- Lower Bound on Delay:

  Logical Resource / Physical Resources
- May be worse
  - dependencies
  - ability to use resource

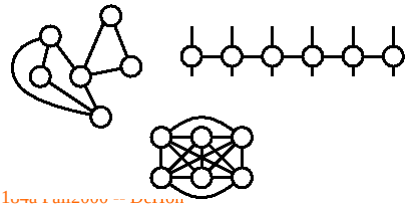
Shared Bus

- Flexibility (+)
  - routes everything (given enough time)
  - can be trick to schedule use optimally
- Delay (Power) (--)
  - wire length O(kn)
  - parasitic stubs: kn+n
  - series switch: 1
  - O(kn)
  - sequentialize I/B
- Area (++)
  - kn switches
  - O(n)



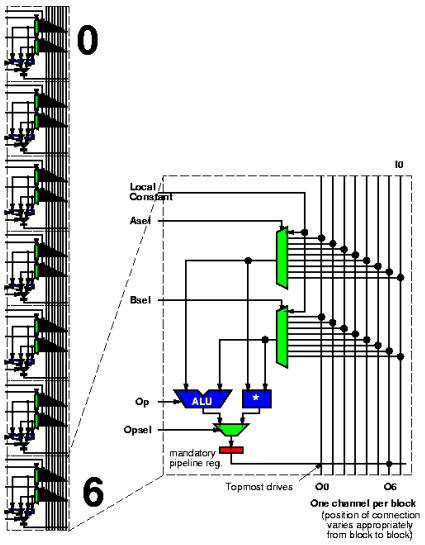
Term: Bisection Bandwidth

- Partition design into two equal size halves
- Minimize wires (nets) with ends in both halves
- Number of wires crossing is **bisection bandwidth**

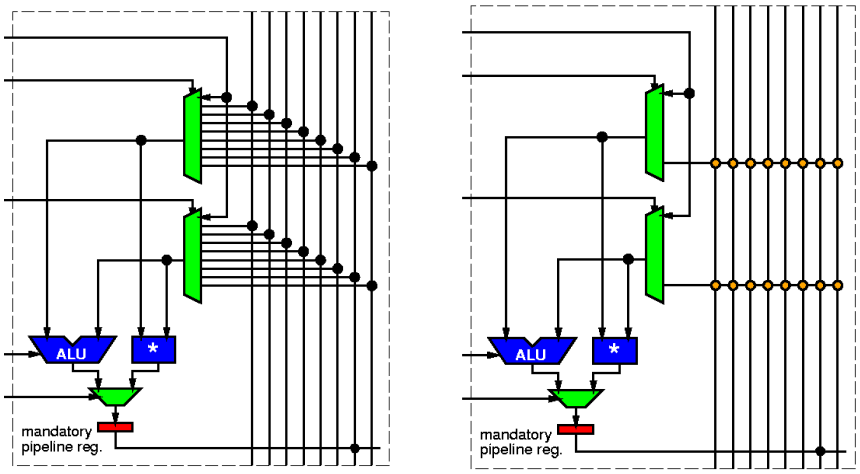


(2) Crossbar

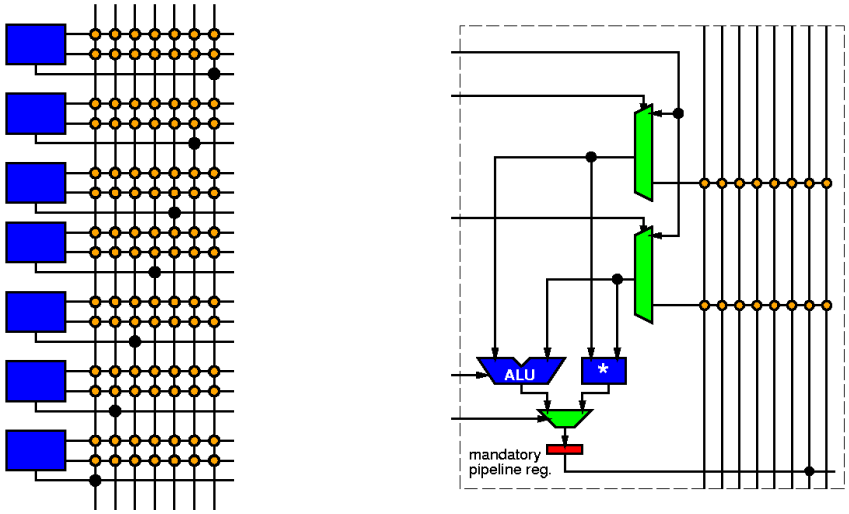
- Avoid bottleneck
- Every output gets its own interconnect channel



Crossbar

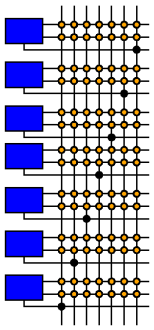


Crossbar



Crossbar

- Flexibility (++)
  - routes everything (guaranteed)
- Delay (Power) (-)
  - wire length O(kn)
  - parasitic stubs: kn+n
  - series switch: 1
  - O(kn)
- Area (-)
  - Bisection bandwidth n
  - kn^2 switches
  - $O(n^2)$



Crossbar

- Too expensive
  - Switch Area = $k*n^2*2.5K\lambda^2$
  - Switch Area/LUT = $k*n*2.5K\lambda^2$
  - n=1024, k=4 => 10M λ^2
- What can we do?

Avoiding Crossbar Costs

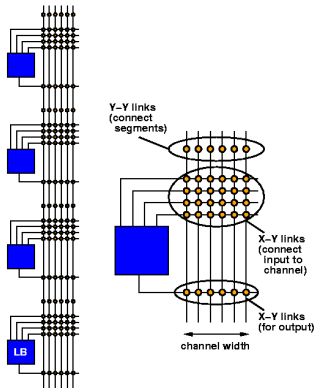
- Typical architecture trick:
  - exploit expected problem structure

Avoiding Crossbar Costs

- Typical architecture trick:
  - exploit expected problem structure
- We have freedom in operator placement
- Designs have spatial locality
- =>place connected components “close” together
  - don’t need full interconnect?

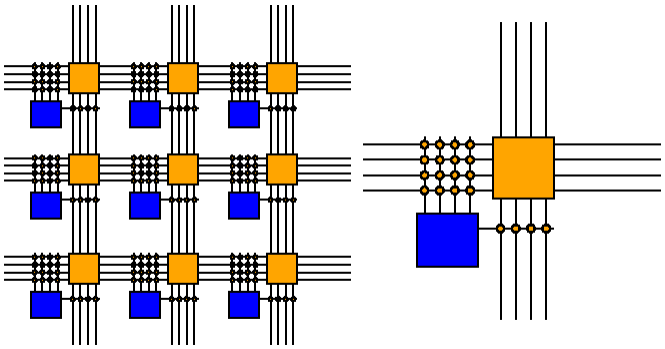
Exploit Locality

- Wires expensive
- Local interconnect cheap
- 1D versions?
- (explore on hmwrk)



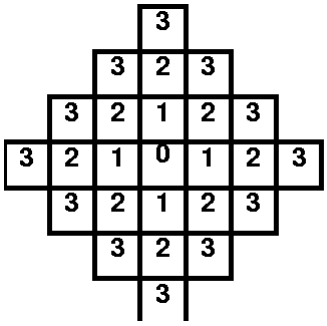
Exploit Locality

- Wires expensive
- Local interconnect cheap
- Use 2D to make more things closer
- Mesh?



Mesh Analysis

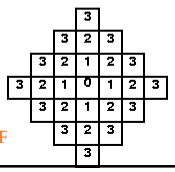
- Can we place everything close?

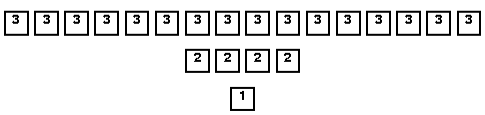


Mesh "Closeness"

- Try placing "everything" close

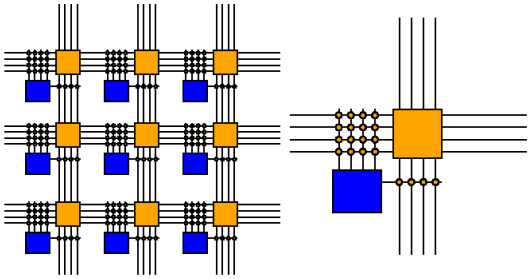
Manhattan Distance	Places	Transitive Fanin
1	4	4
2	8	16
3	12	64
⋮	⋮	⋮
n	$4n$	4^n





Mesh Analysis

- Flexibility - ?
  - Ok w/ large w
- Delay (Power)
  - Series switches
    - 1--$\sqrt{n}$
  - Wire length
    - w--$\sqrt{n}$
  - Stubs
    - O(w)--O(w$\sqrt{n}$)
- Area
  - Bisection BW -- w$\sqrt{n}$
  - Switches -- O(nw)
  - O(w^2n) [linear pop]
  - larger on homework



Mesh

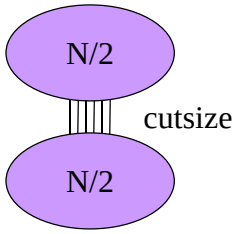
- Plausible
- …but What's w
- …and how does it grow?

Characterize Locality

- Want to exploit locality
- How much locality do we have?
- Impact on resources required?

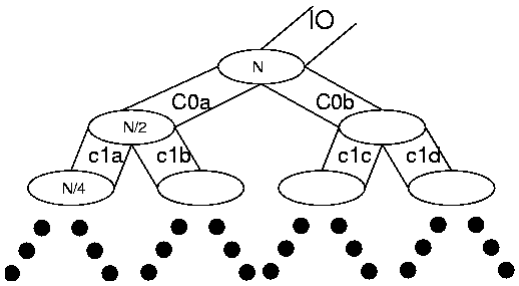
Bisection Bandwidth

- Bisect design
- Bisection bandwidth of design
  - => lower bound on network bisection bandwidth
- Design with more locality
  - => lower bisection bandwidth
- Enough?



Characterizing Locality

- Single cut not capture locality within halves
- Cut again
  - => recursive bisection

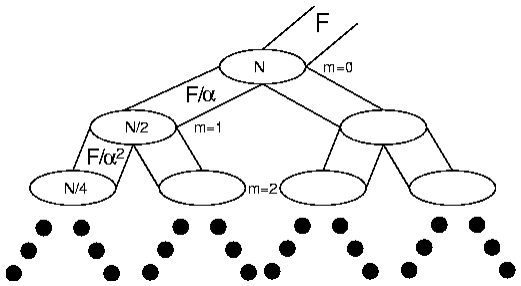


Regularizing Growth

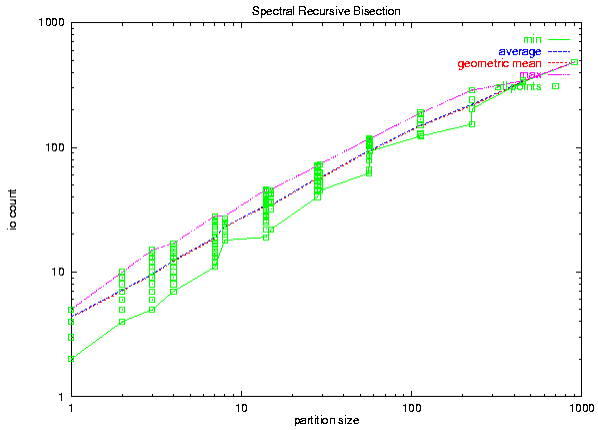
- How do bisection bandwidths shrink (grow) at different levels of bisection hierarchy?
- Basic assumption: Geometric
  - 1
  - $1/\alpha$
  - $1/\alpha^2$

Geometric Growth

- (F,α)-bifurcator
  - F bandwidth at root
  - geometric regression α at each level



Good Model?



Log-log plot ==> straight lines represent geometric growth

Rent's Rule

- Long standing **empirical** relationship
  - $IO = C*N^P$
  - $0 \leq P \leq 1.0$
  - compare (F,α)-bifurcator
    $\alpha = 2^P$
- Captures notion of locality
  - some signals generated and consumed locally
  - reconvergent fanout

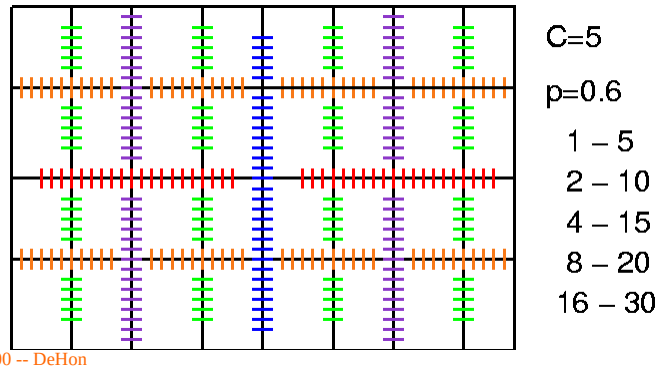
Monday class stopped here

Rent's Rule

- Typically consider
  - $0.5 \leq P \leq 0.75$
- "High-Speed" Logic P=0.67
- Memory (P~0.1-0.2)
- Example (i10)
  - max C=7, P=0.68
  - avg C=5, P=0.72

What tell us about design?

- Recursive bandwidth requirements in network

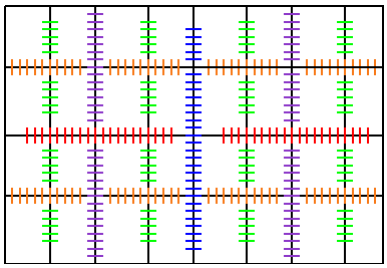


What tell us about design?

- Recursive bandwidth requirements in network
  - lower bound on resource requirements
- N.B. **necessary** but not **sufficient** condition on network design
  - *I.e.* design must also be able to *use* the wires

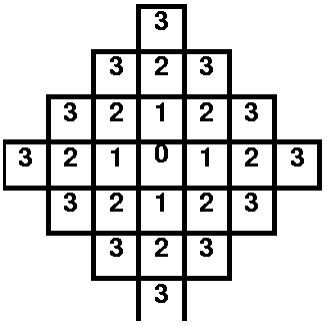
What tell us about design?

- Interconnect lengths
  - Intuition
    - if p>0.5, everything cannot be nearest neighbor
    - as p grows, so wire distances



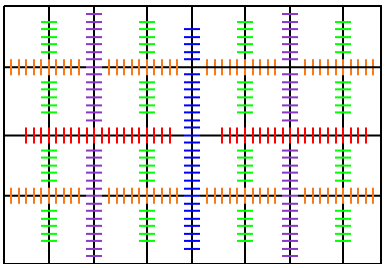
What tell us about design?

- Interconnect lengths
  - IO=$(n^2)^P$ cross distance n
  - dIO/dn end at exactly distance n
  - E(l)=Integral 0 to n=$\sqrt{N}$
    - of $n*(dIO/dn)/n^2$
    - assume iid sources
  - $E(l)=O(N^{(p-0.5)})$
    - p>0.5



What Tell us about design?

- IO∝N^P
- Bisection BW∝N^P
- side length ∝N^P
  - N if p<0.5
- Area ∝N^{2p}
  - p>0.5



N.B. 2D VLSI world has
"natural" Rent of P=0.5
vs. perimeter)

- Modern “systems” on a chip -- likely to contain subcomponents of varying Rent complexity
- Less I/O at certain “natural” boundaries
- System close
  - (Rent’s Rule apply to workstation, PC, PDA?)

- Bad news
  - Area ~ O(N
    - faster than N
  - Avg $^{(p-0.5)}$)
    - grows with N
- once appreciate its effects?
- *I.e.* style/criteria so we mitigate effects?

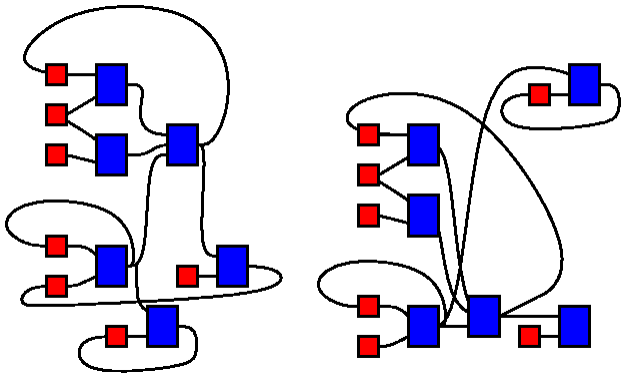
What Rent didn't tell us

- Bisection bandwidth purely geometrical

  No constraint for delay

  – *I.e*

Critical Path and Bisection



Minimum cut may cross critical path multiple times.
Minimizing long wires in critical path => increase cut size.

Rent Weakness

- Not account for path topology
- ? Can we define a "Temporal" Rent which takes into consideration?
  - Promising research topic

Finishing Up...

Interconnect Dominant

– power, delay, area

-
- Can’t afford full crossbar

  Need to exploit locality
-

Big Ideas [MSB Ideas]

2^p)

- p>0.5 => interconnect growing faster than compute elements
  – expect interconnect to dominate other resources