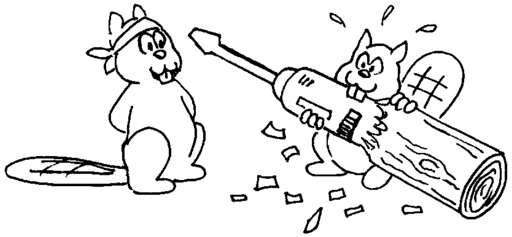
Entropy, Counting, and Programmable Interconnect

André DeHon
<andre@ai.mit.edu>

Reinventing Computing



Acknowledgments: This research is supported by the Advanced Research Projects Agency of the Department of Defense under Rome Labs contract number F30602-94-C-0252.

MIT AI Lab

Outline

- What is this talk about?
- Why might you care?
- Interconnect bits required?
  - Examples
  - Bounds
- Interconnect bits in use
- Implications
  - Area
  - Reconfiguration Time

Talk Focus

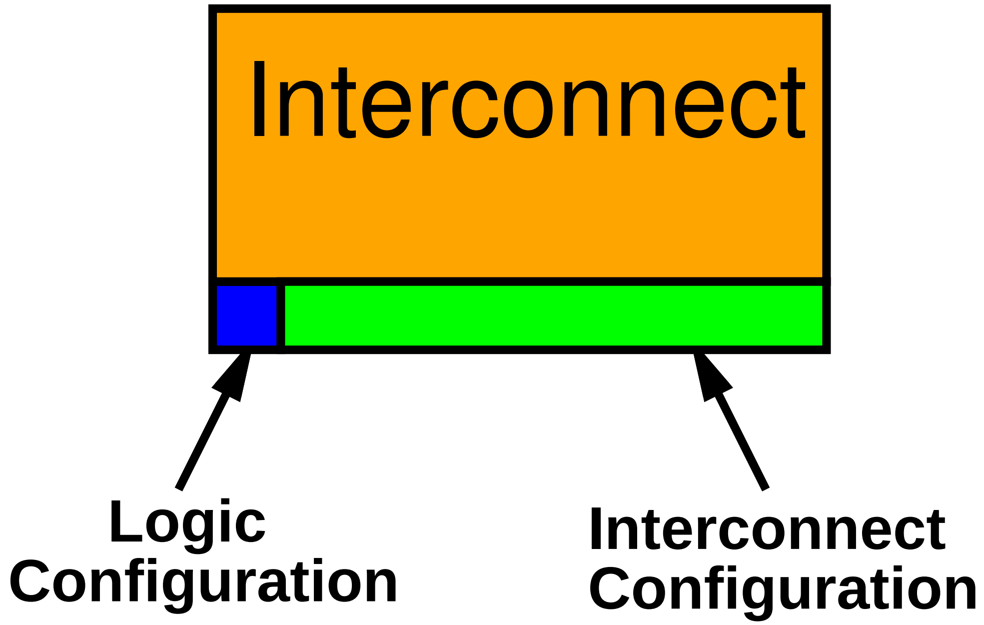
Interconnect Description Size –

How many bits does it take to configure the programmable interconnect?

...but don't let that fool you.

Most of the area in reconfigurable devices is in the interconnect!

Relative Area



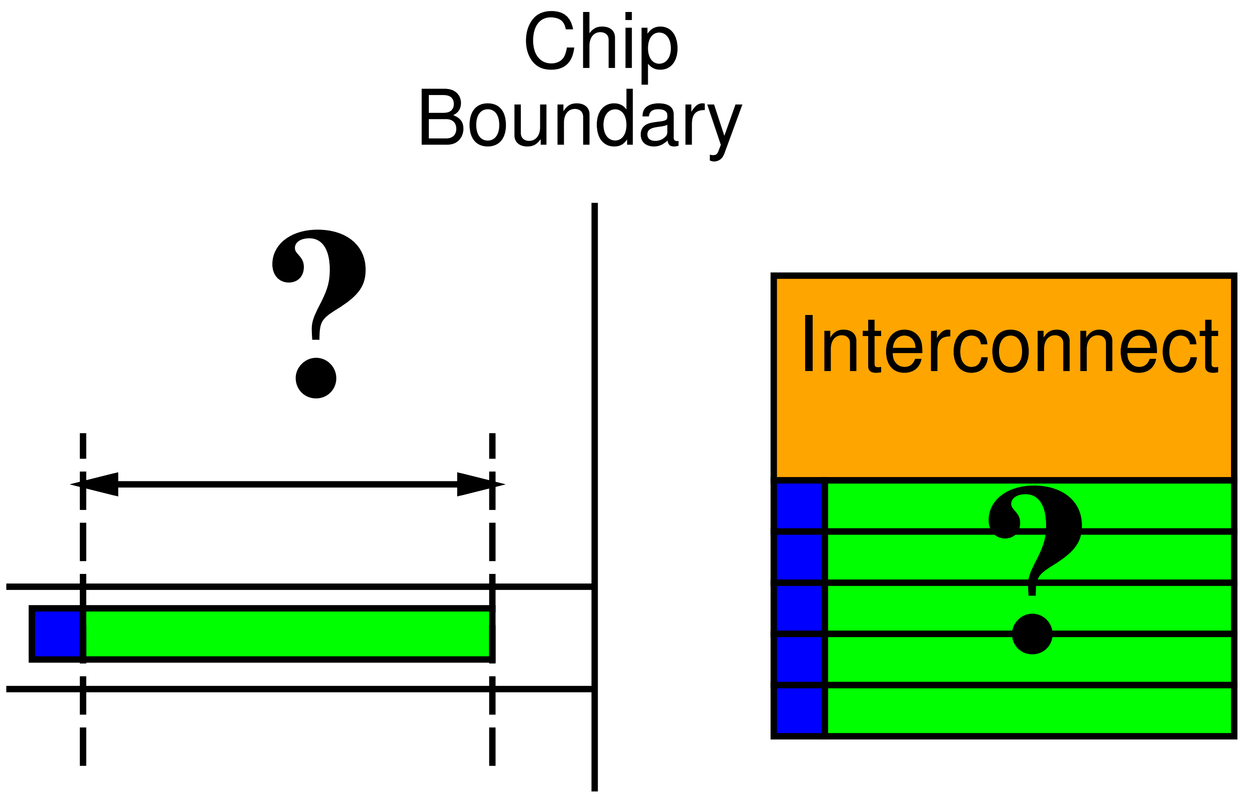
Conventional FPGAs:

Typical 80% to programmable interconnect
20% to configuration
90% of this describes interconnect

Why might you care?

- Increase **speed** of reload
- Increase **capacity** through reuse
- Decrease **cost** by reducing chip **size**

Talk Addresses

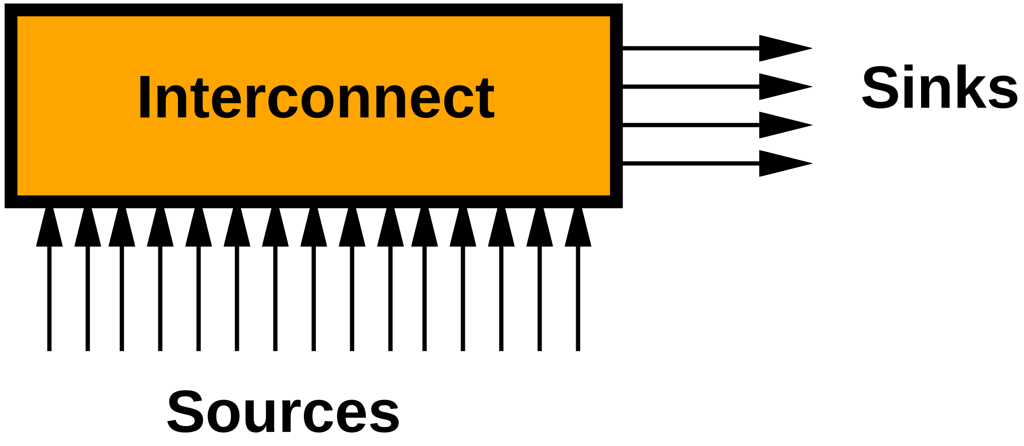


- How much data has to cross the chip boundary?
- How much data has to be stored in on-chip configurations?

MIT AI Lab

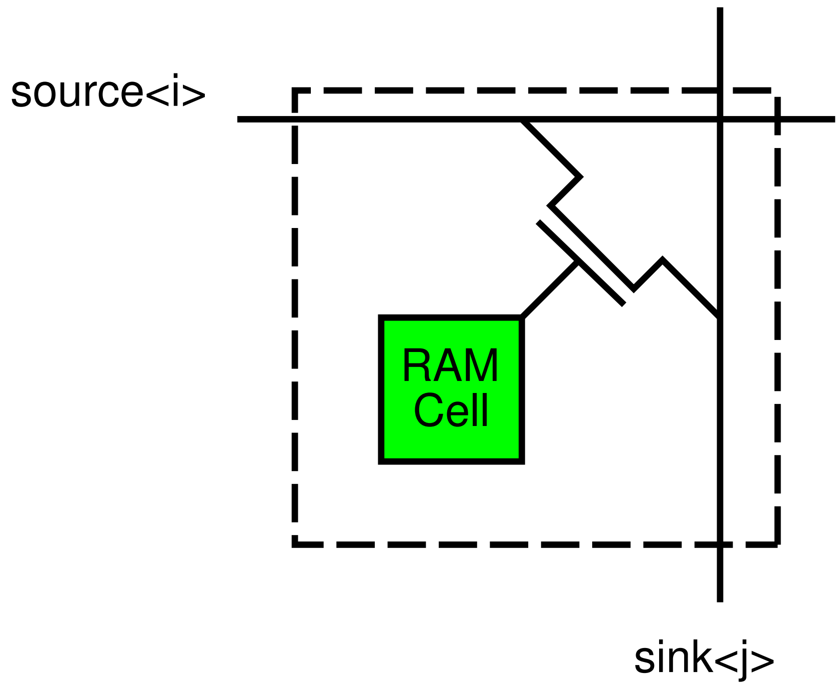
Example

Sample Interconnect



- n **sources (*e.g.* 15)**
- m **sinks (*e.g.* 4)**

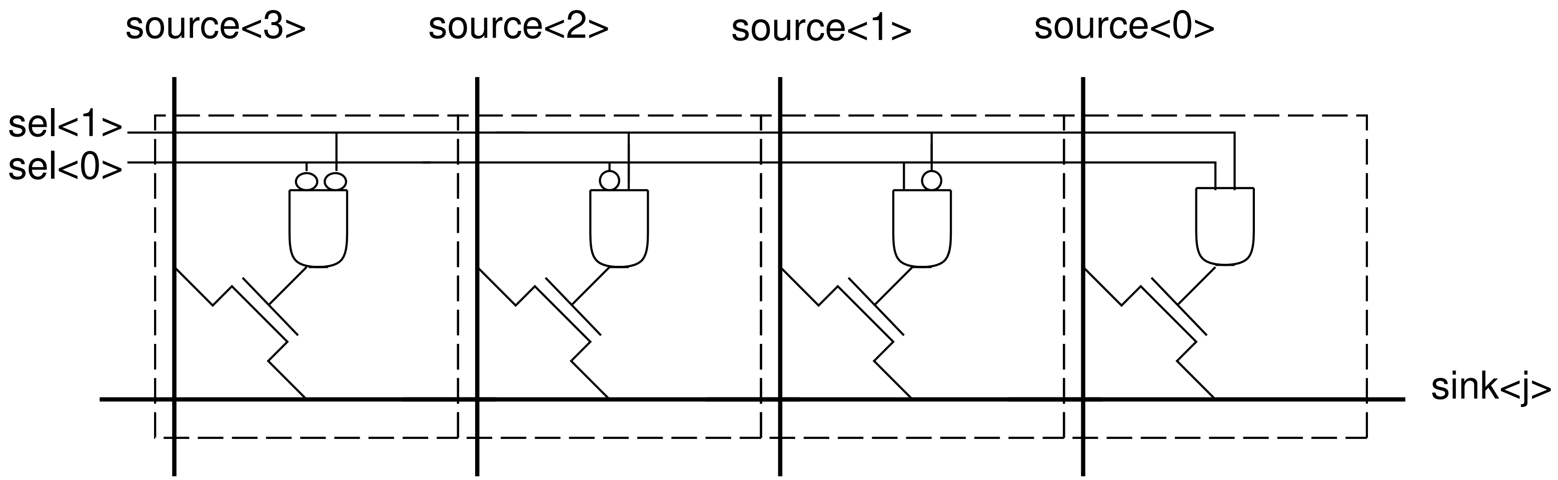
Full Crossbar



Full Crossbar

- $m \times n$ **crosspoints (*e.g.* 60)**
- $m \times n$ **memory cells (?) (*e.g.* 60)**

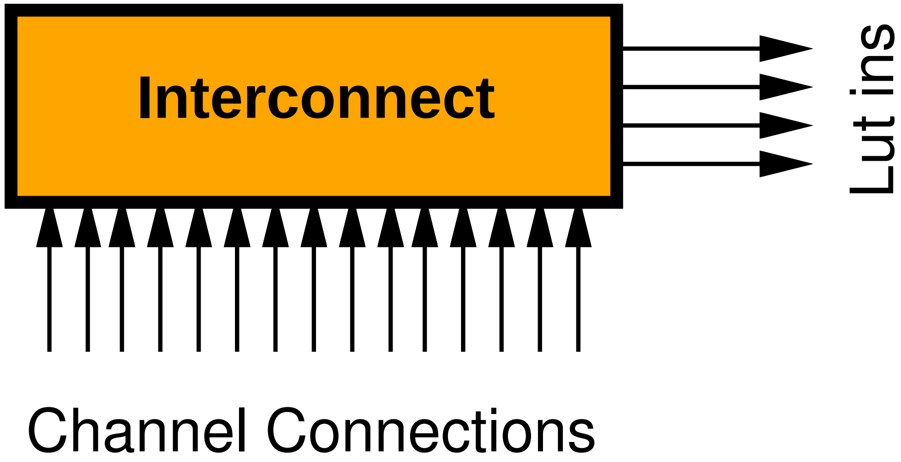
Encoded Crossbar



Encoded Crossbar

- $m \times n$ **crosspoints (*e.g.* 60)**
- $m \log_2(n)$ **memory cells (*e.g.* 15)**

Distinguishable Sinks?



If m sinks are all equivalent (*e.g.* k inputs to a k-LUT):

$$\left\lceil \log_2 \binom{n}{m} \right\rceil$$

$$\left\lceil \log_2 \binom{15}{4} \right\rceil = 11$$

MIT AI Lab

How Many Bits?

How Many Bits?

So, how many bits do we really need to describe any interconnect?

Use full crossbar to get an upper bound:

$$\textbf{Interconnect Bits/Sink} <= \log_2(Sources)$$

$$\textbf{Interconnect Bits/4-LUT} <= 4 \cdot \log_2(\textbf{LUTs} + \textbf{Inputs})$$

e.g. **1000 4-LUT device, 200 I/Os** $\approx$ **44 bits/4-LUT**
e.g. **9000 4-LUT device, 600 I/Os** $\approx$ **56 bits/4-LUT**

How Many Bits? – Sink Commonality

Exploit k-LUT input commonality:

$$\textbf{Interconnect Bits}/k\textbf{-LUT} <= \log_2 \binom{\textbf{LUTs} + \textbf{Inputs}}{k}$$

e.g. **1000 4-LUT device, 200 I/Os $\approx$ 37 bits/4-LUT**
e.g. **9000 4-LUT device, 600 I/Os $\approx$ 49 bits/4-LUT**

(compare with 16 bits to specify LUT function)

How Many Bits? – Conventional Devices

Part	n_{blocks}	n_{io}
Xilinx xc4013	578 CLBs	192 IOBs
Xilinx xc5210	324 CLBs	196 IOs
Altera EPF81188	1008 LEs	184 IOEs
Pedagogical Reference	1024 4-LUTs	200

(each roughly 1K 4-LUTs)

How Many Bits? – Conventional Devices

Part	Logic Bits	Net Bits	Actual Bits
Xilinx xc4013	40	156	420
Xilinx xc5210	64	192	510
Altera EPF81188	16	44	190
Pedagogical Reference	16	44	–

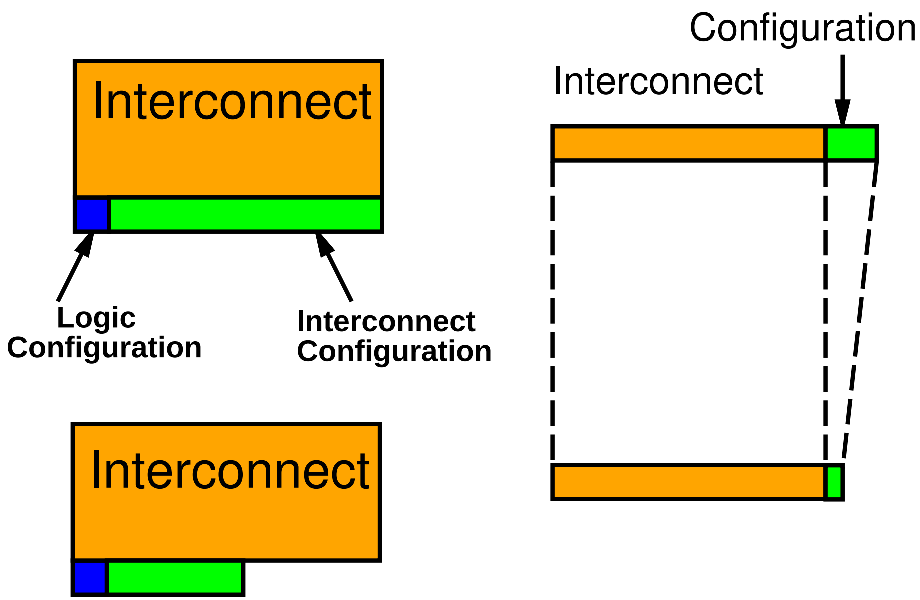
Net Bits/block $<=$

$$n_{block\_ins} \cdot \log_2\left(n_{blocks} \cdot n_{block\_outs} + n_{io} \cdot n_{io\_outs}\right)$$

MIT
AI
Lab

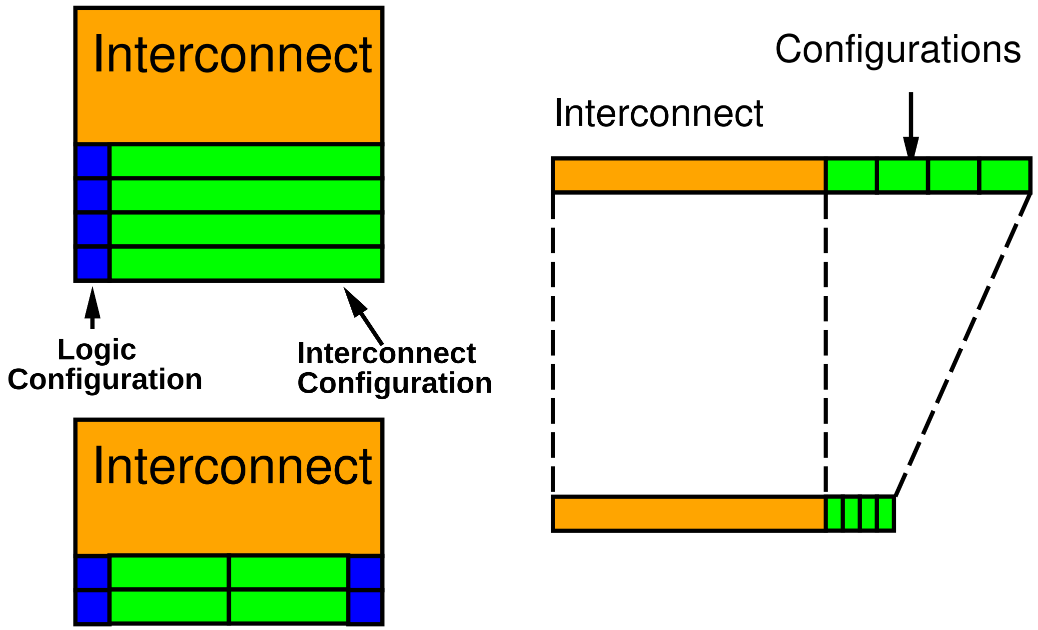
Implications

Implications – Area (Single Context)



Might save 5-10% of FPGA area.
Might save 50-75% of ROM/memory space.

Implications – Area (Multicontext)

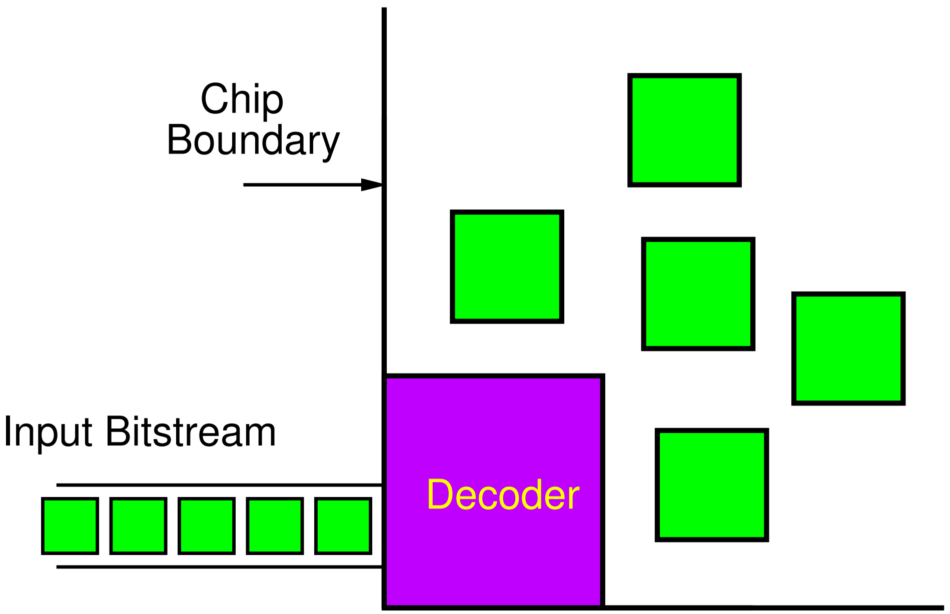


As configurations get large may save 50-75% of multi-context FPGA area.

Implications – Reconfiguration Time

Opportunity: cut reconfiguration time by **50-75%** without adding i/o pins or increasing clock rate

Applicable to conventional FPGA designs.



MIT AI Lab

Summary

- Conventional FPGAs
  - sparsely encoded interconnect
  - acceptable in conventional operating space
  - opportunity to reduce stored configuration size
- As rapid reconfiguration becomes important
  - denser encodings reduce reconfiguration bandwidth
  - multicontext devices reduce required bandwidth
  - denser encodings reduce the memory overhead for multicontext devices
- Bitstream compression reviewed here is design independent

Additional Information

- Full Technical Report in HTML
  - <http://www.ai.mit.edu/projects/transit/tn128/tn128.html>
- Full Technical Report in PostScript
  - <ftp://transit.ai.mit.edu/transit-notes/tn128.ps.Z>
- FPGA'96 paper in PostScript
  - <ftp://transit.ai.mit.edu/paper/entropy-fpga96.ps.Z>
- PostScript for these slides
  - <ftp://transit.ai.mit.edu/slides/entropy-fpga96.ps.Z>
- Reinventing Computing Home Page
  - <http://www.ai.mit.edu/projects/transit/rc_home_page.html>