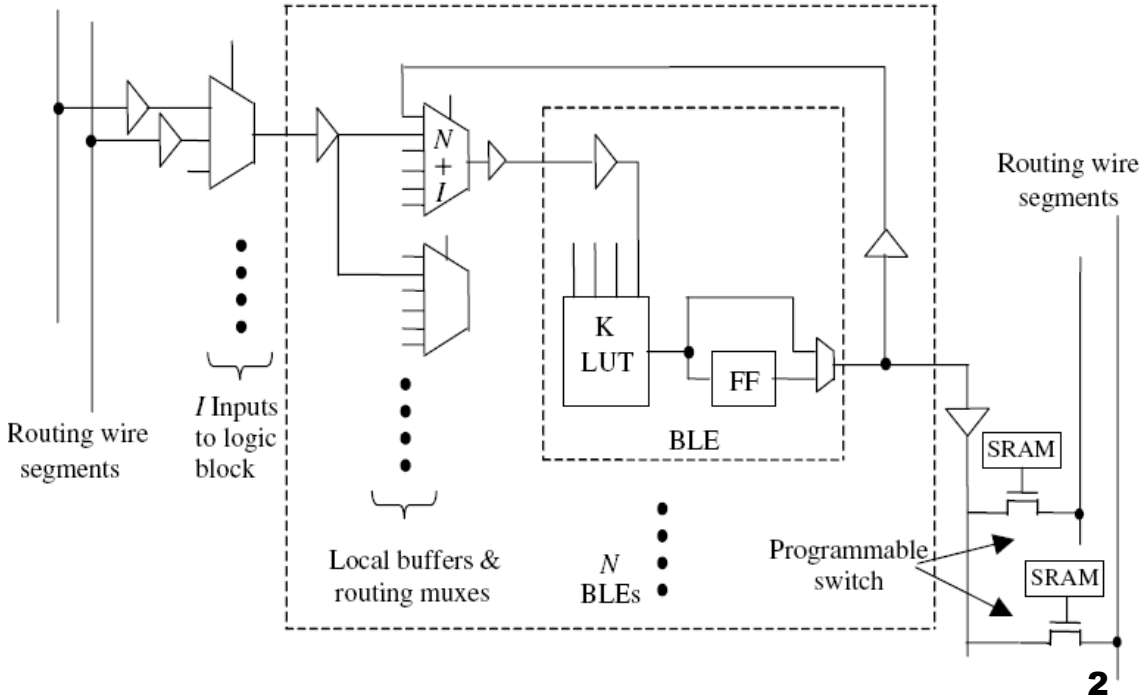
Circuit Design and Architecture Exploration of FPGAs

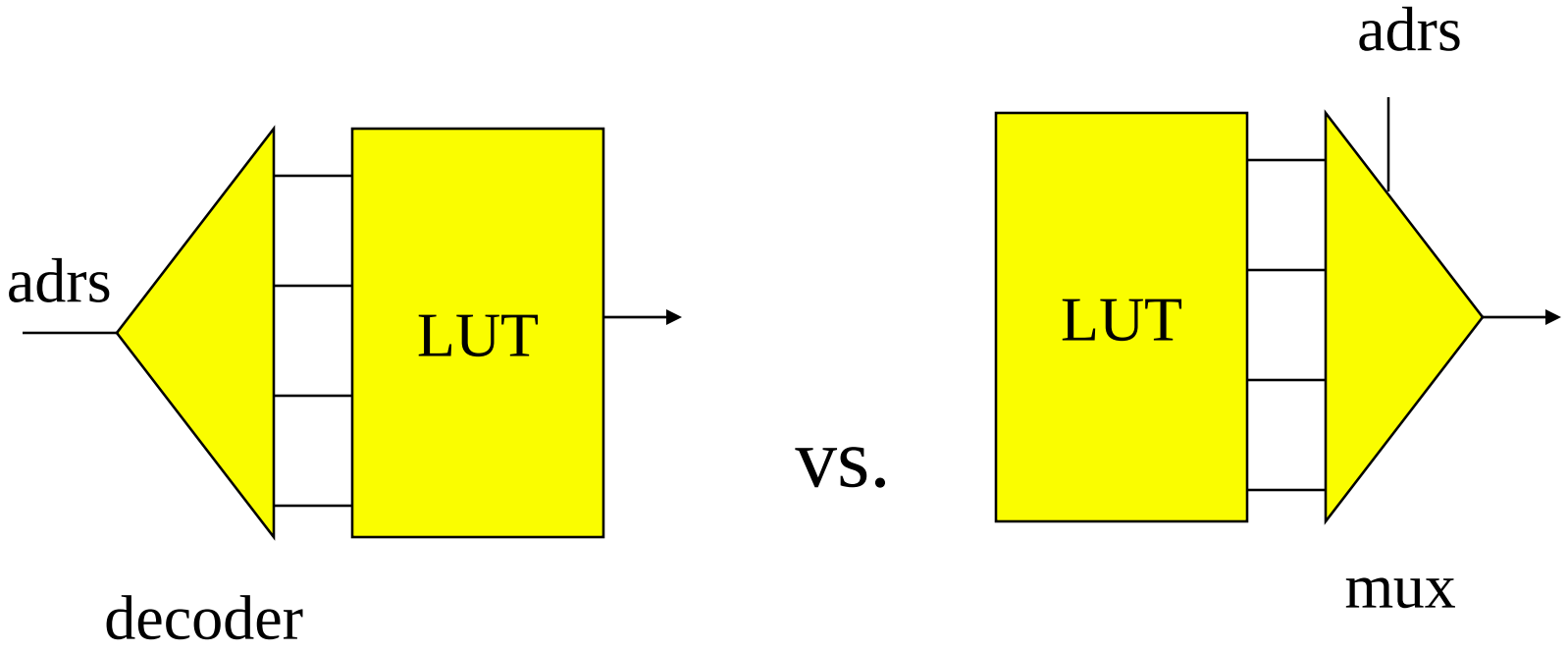
Topics

- Circuit Design for FPGAs
- Logic Block Architecture Study
- Routing Architecture Study



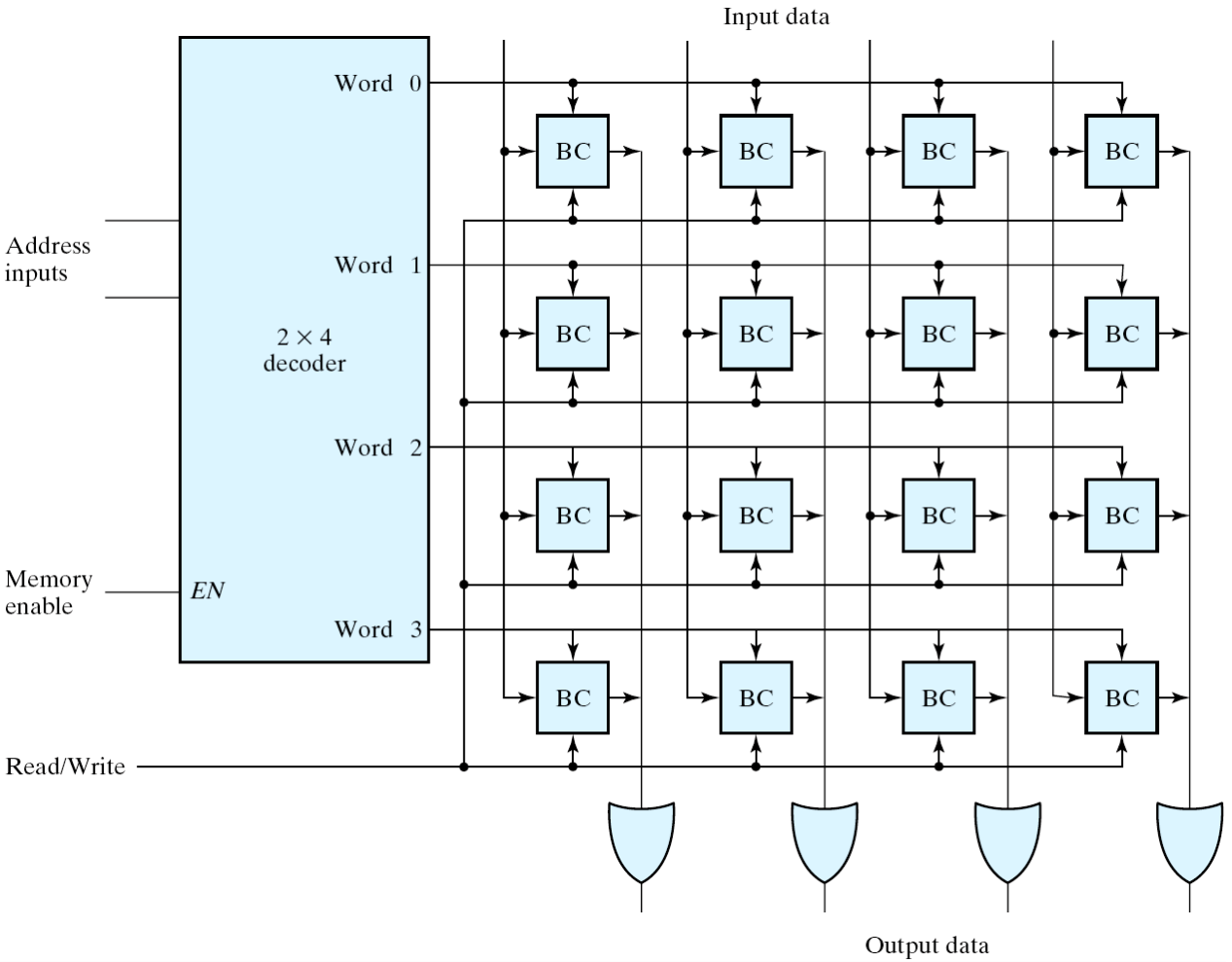
Lookup Table Circuitry

- 2 options for bit-selection
  - decoder or multiplexer?



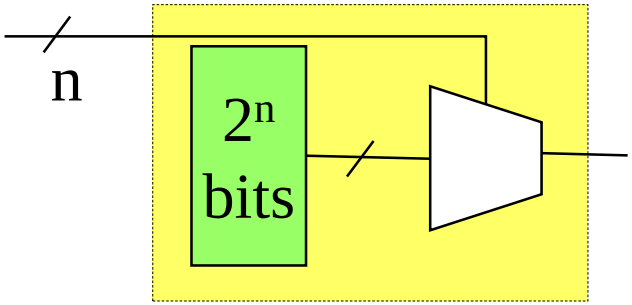
Bit-Selection in Conventional RAM/ROM

- RAM/ROM typically uses decoder for bit-selection



Bit-Selection for LUT

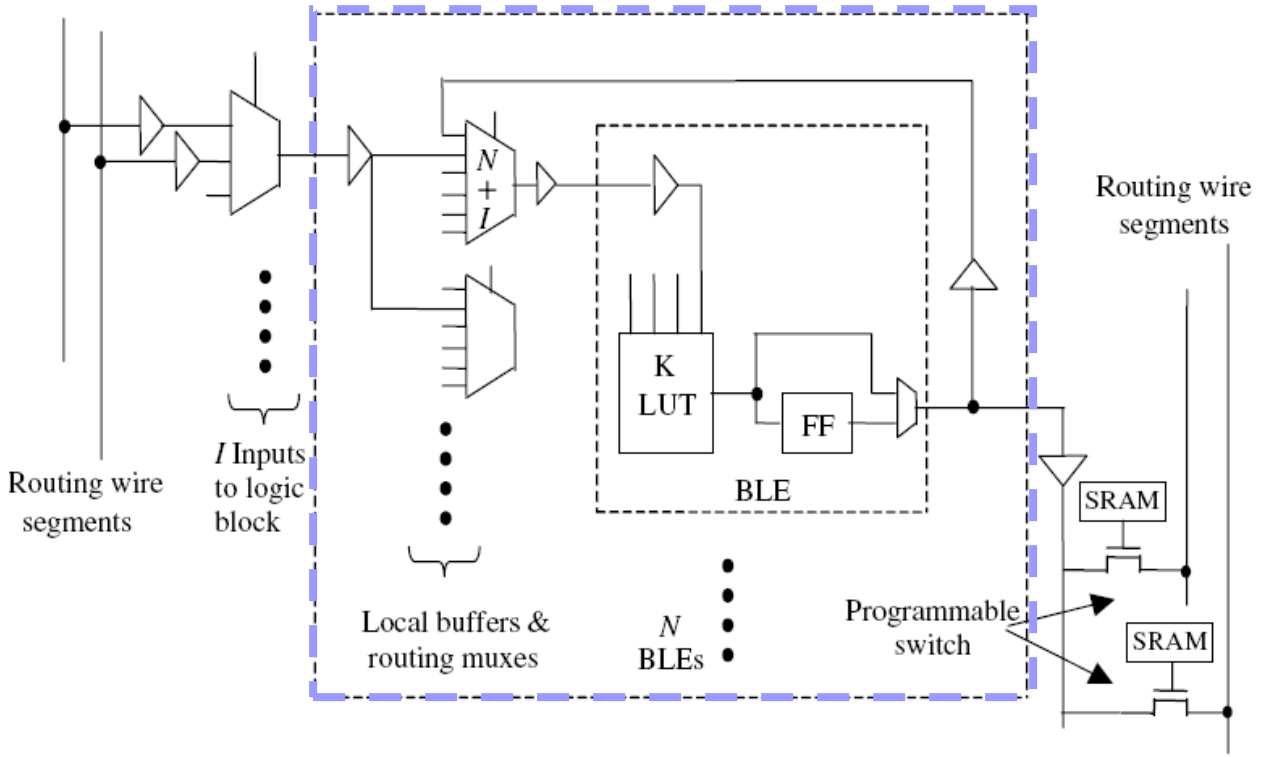
- FPGA's LUT uses a multiplexer for bit-selection.



- Multiplexer presents smaller load to memory cells.
  - Allows smaller memory cells

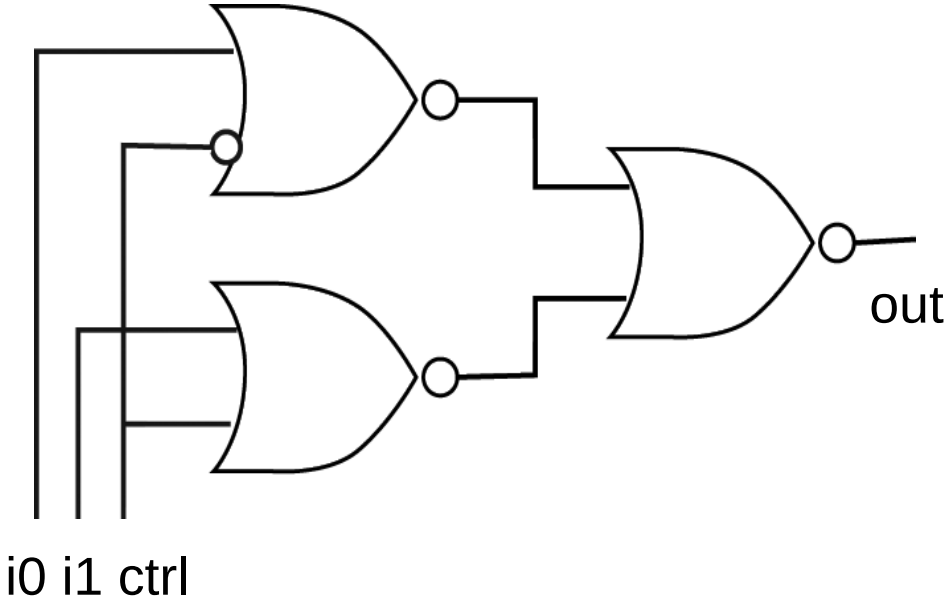
Multiplexers in FPGA

- Muxes used inside LUTs and for routing (intra-block and inter-block).

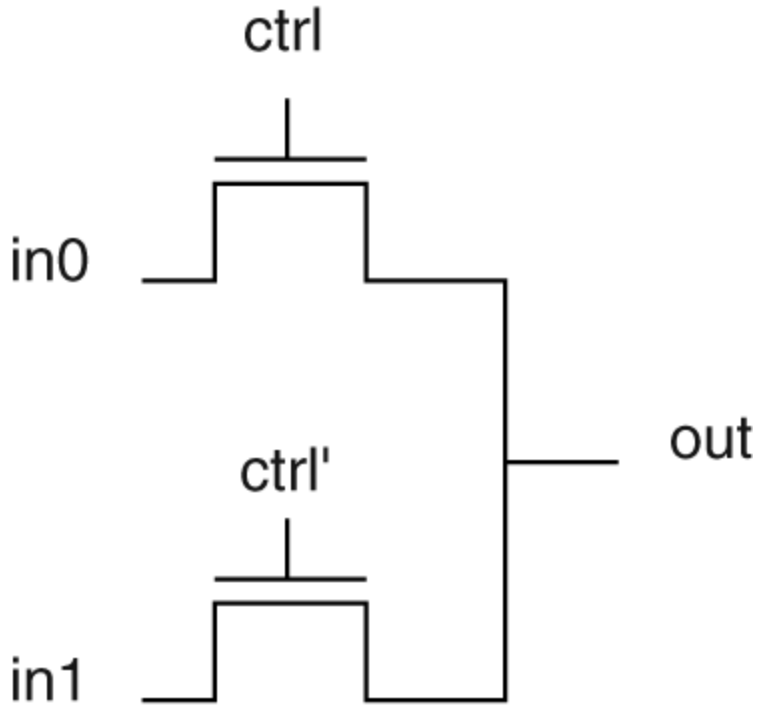


A logic block and its periphery

Multiplexer Design



By static gates



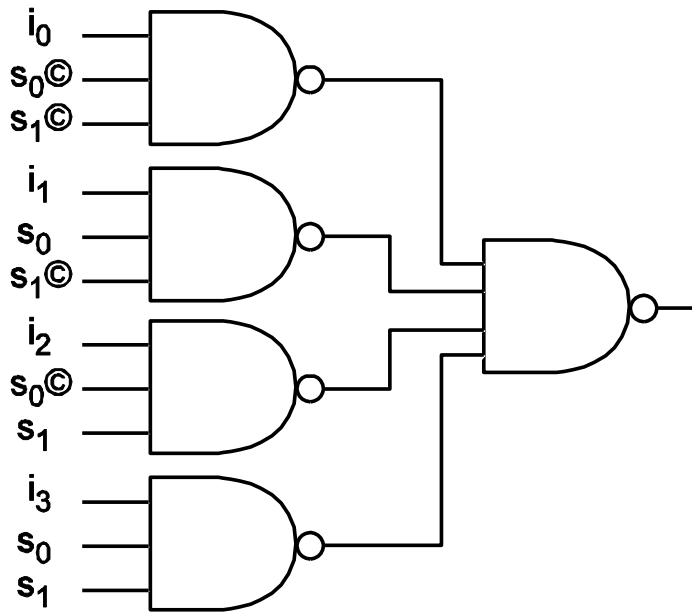
By pass transistors

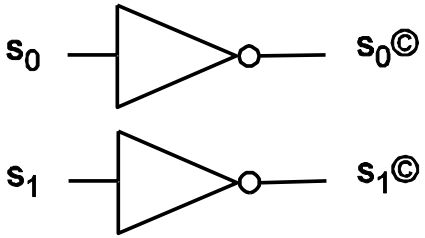
Multiplexer Design

- How about using complementary switches instead of pass transistors?
- Pass transistor uses 1 transistor while complementary switch uses 2.
- Pass transistor is faster than complementary switch:
  - Equal-strength p-type is 2.5X n-type width.
  - Total resistance is 0.5X, total capacitance is 3.5X.
  - RC delay is 0.5 x 3.5 = 1.75 times n-type switch.

Performance of Static Gate MUX

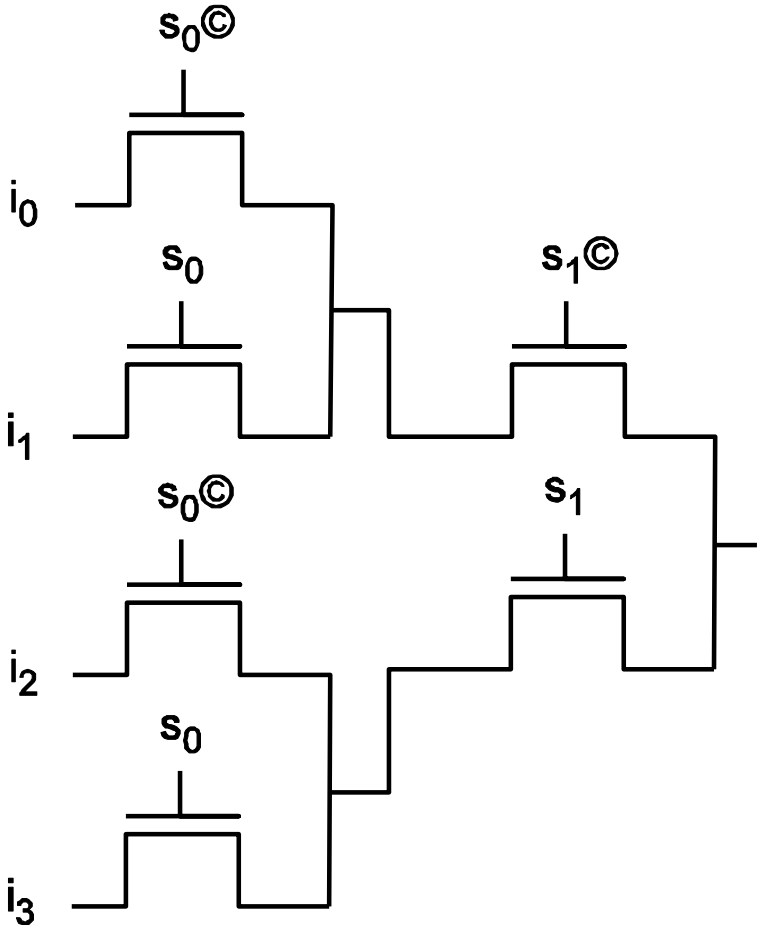
- Delay through *n*-input NAND is (*n*+2)/3 using logical effort computation.
- For *b*-to-1 MUX
  - Lg *b* + 1 inputs at first level, so delay is (lg *b* + 3)/3.
  - Delay at second level is (*b*+2)/3.
- Delay grows as *b*.





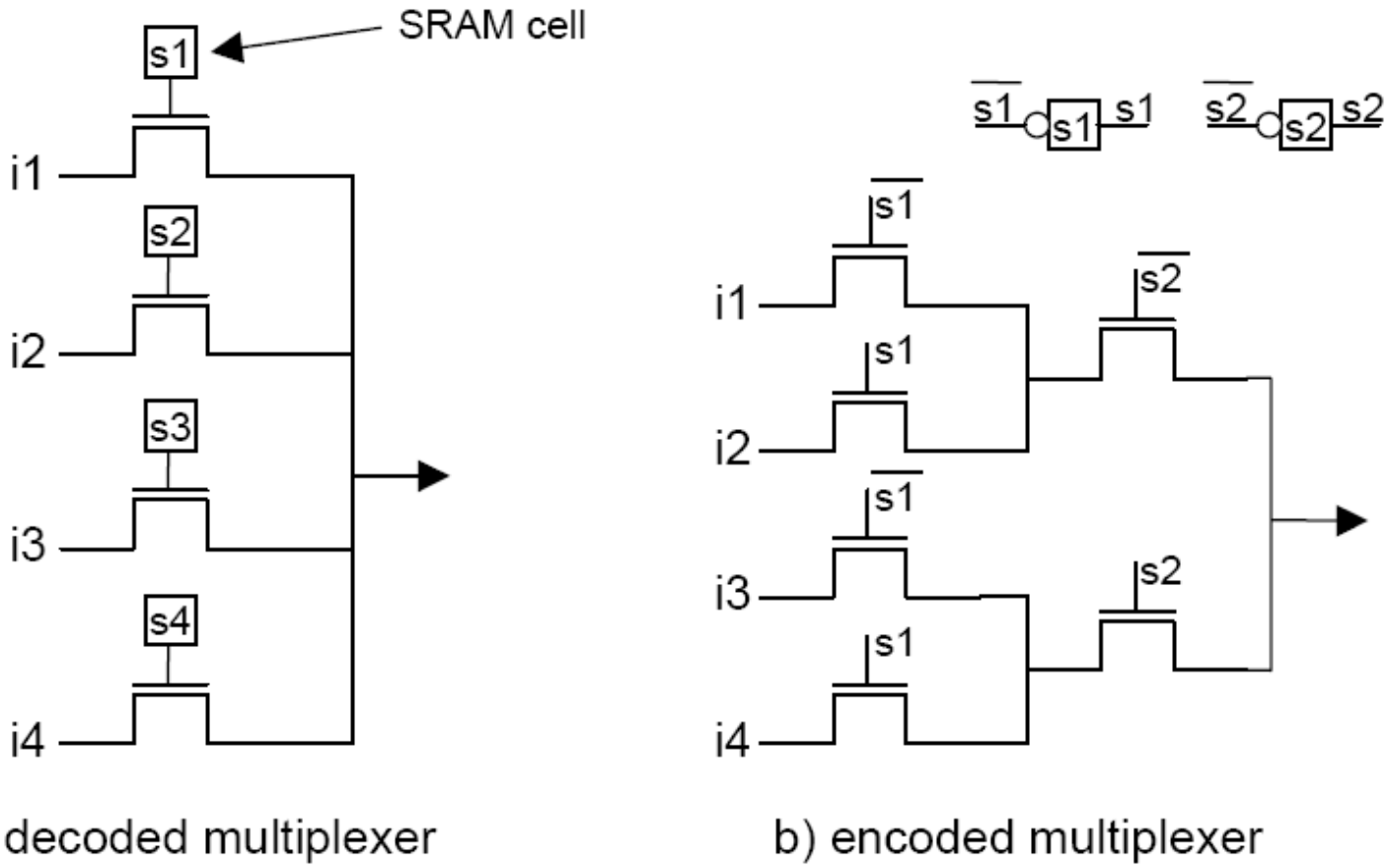
Performance of Tree-based Pass Transistor MUX

- Delay proportional to square of path length.
- Delay grows as $(\lg b)^2$.



Encoded MUX vs Decoded MUX

- Tradeoff between transistor count and delay

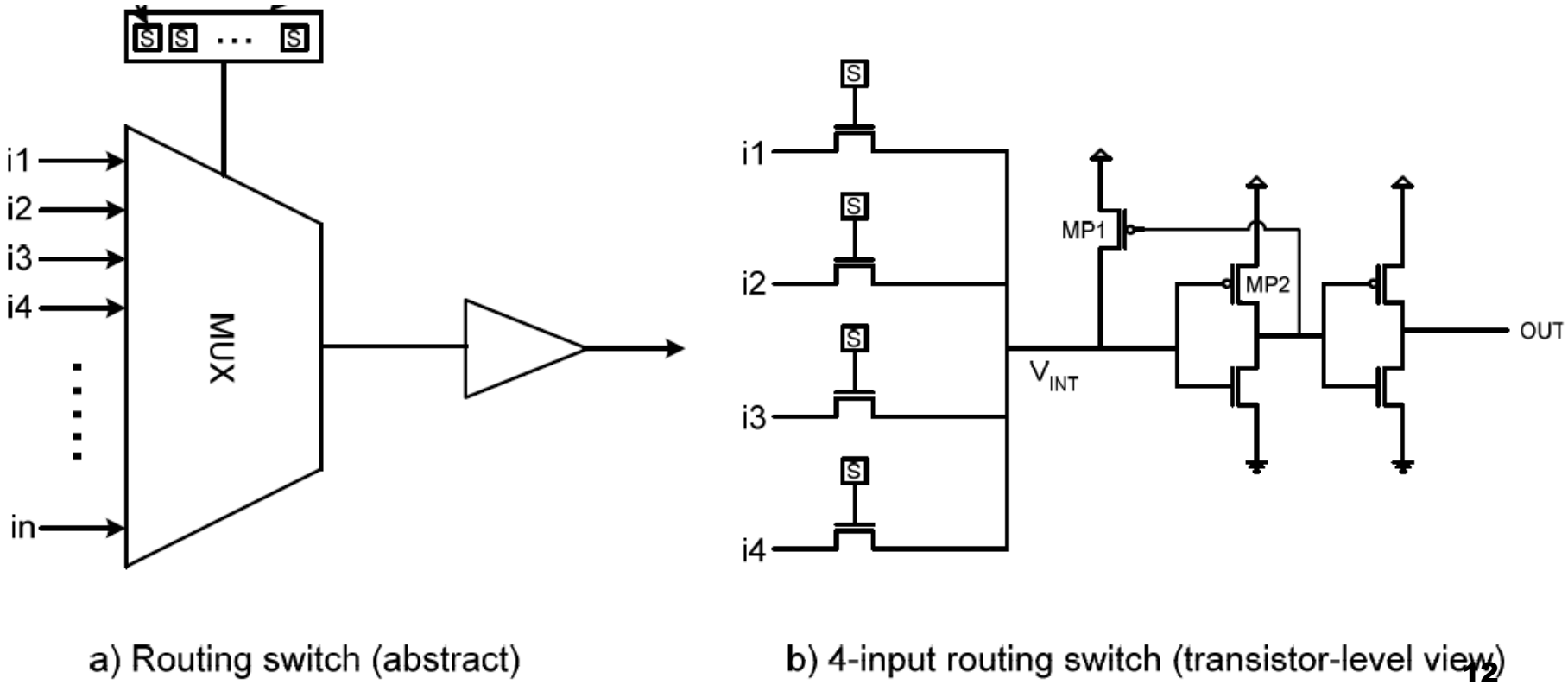


a) decoded multiplexer

b) encoded multiplexer

Leakage in MUX-based Routing Switch

- Level restoring buffer to avoid leakage at MP2 due to a weak V_{INT}

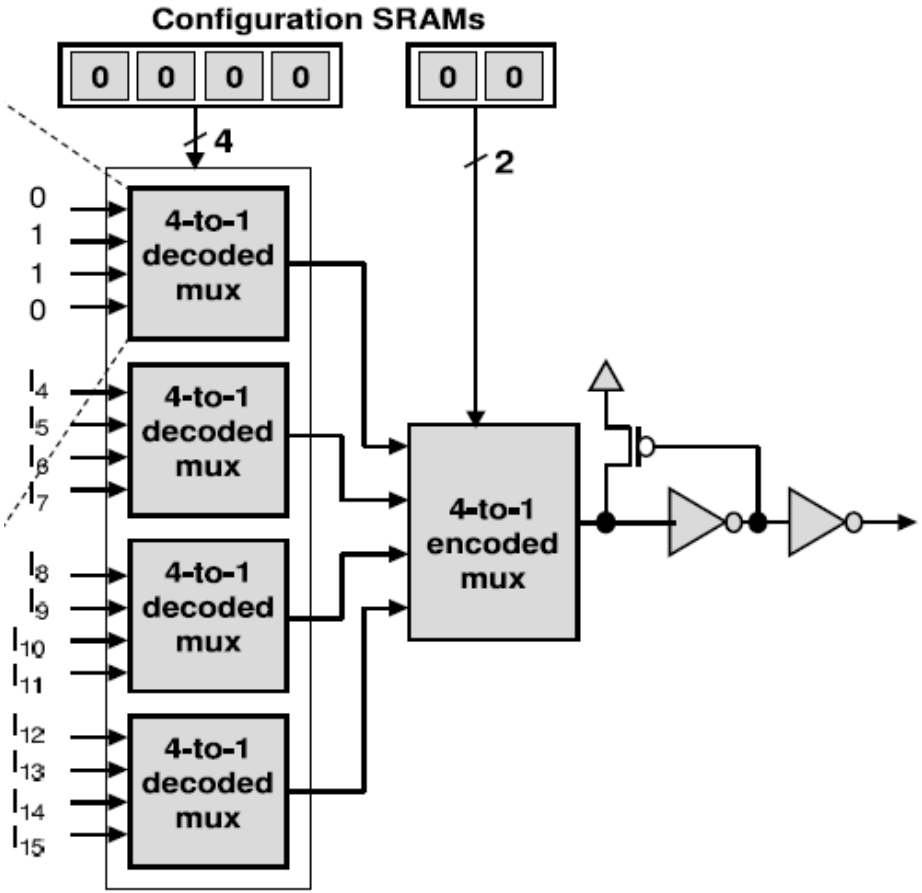


a) Routing switch (abstract)

b) 4-input routing switch (transistor-level view)

MUX-based Routing Switch Design

- Optimize: transistor count, delay, leakage

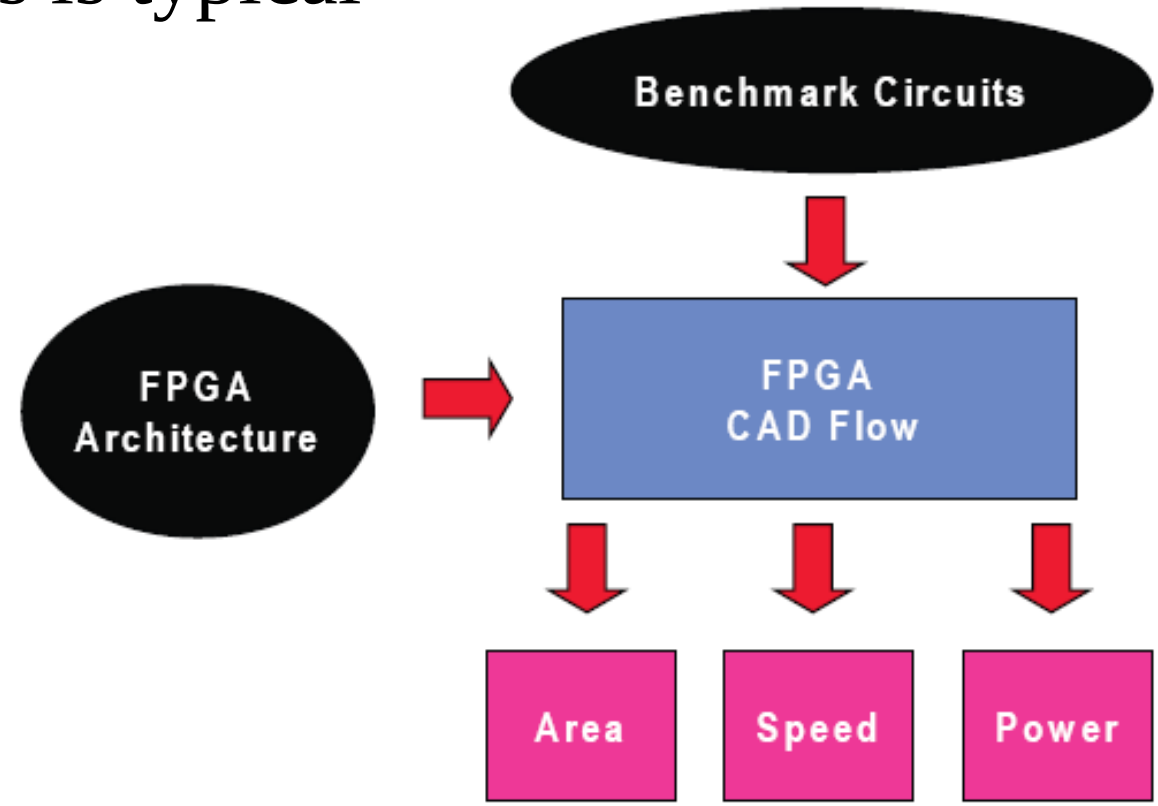


Architectural Issues

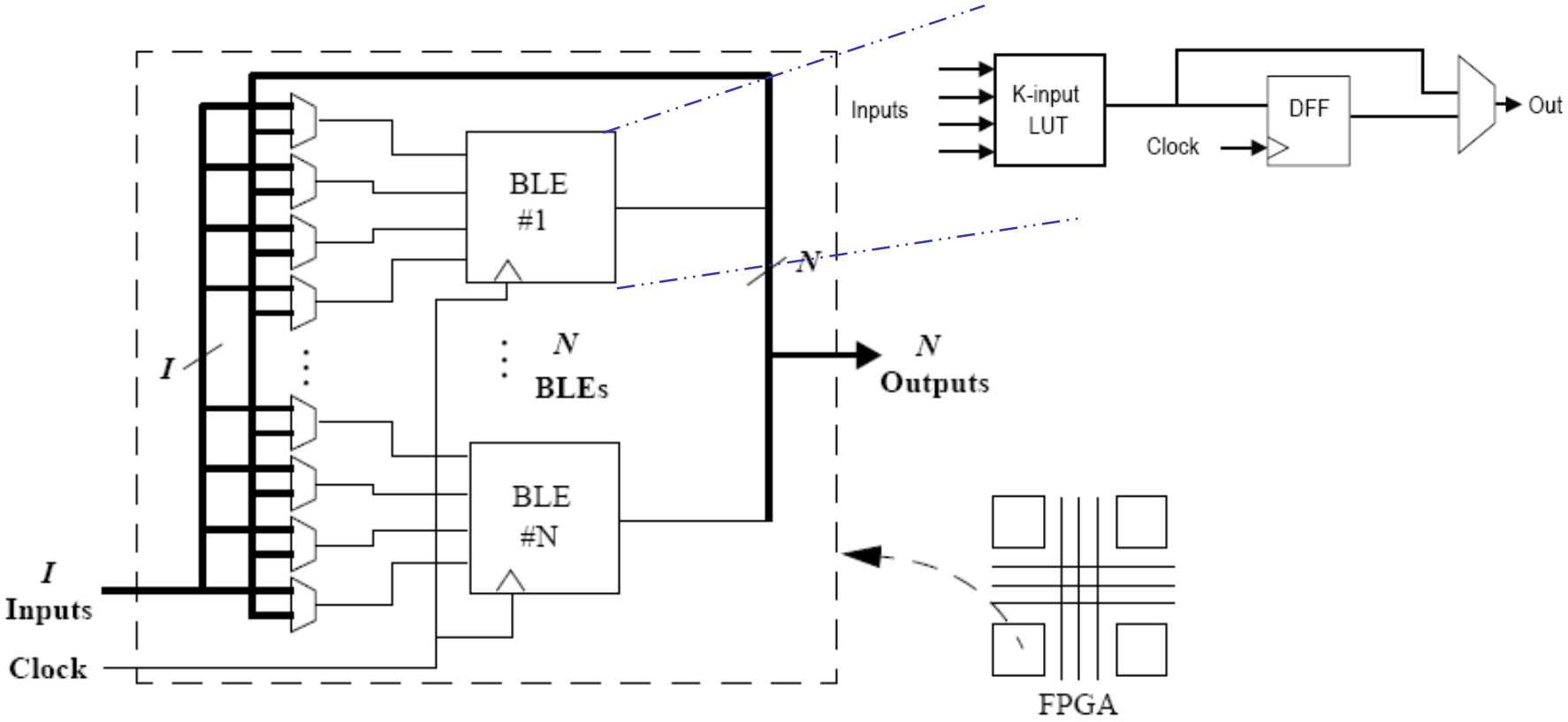
- Granularity of logic elements in the FPGA?
- LE structure:
  - What functions?
  - How many inputs?
  - Dedicated logic?
- What types of interconnect?
  - How much of each type?
- How long should interconnect segments be?
- How should we vary interconnect?
  - Uniform or non-uniform over chip?

FPGA Architecture Evaluation Methodology

- Empirical approach to explore different architectures is typical



Logic Block Structure

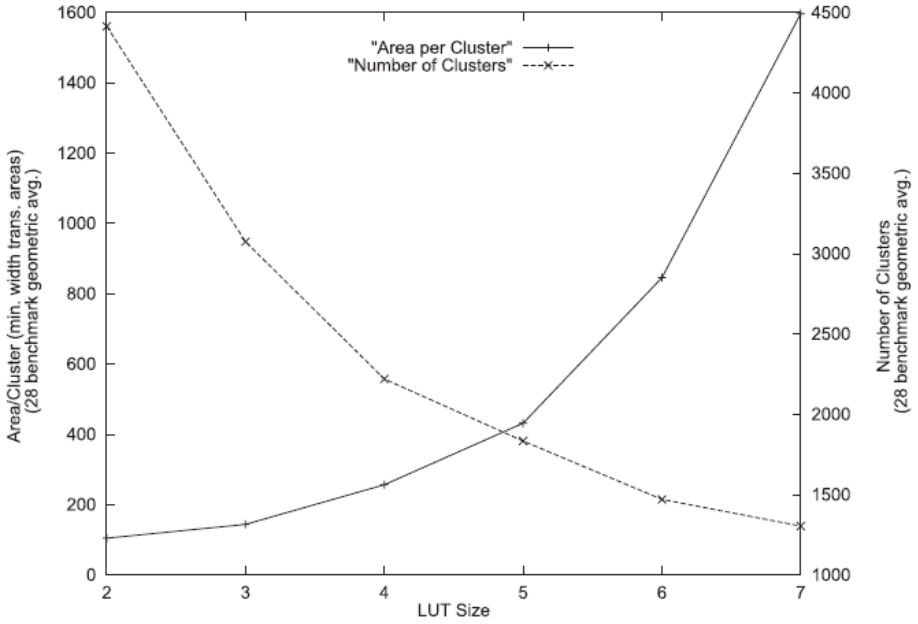


Logic Block Granularity Study

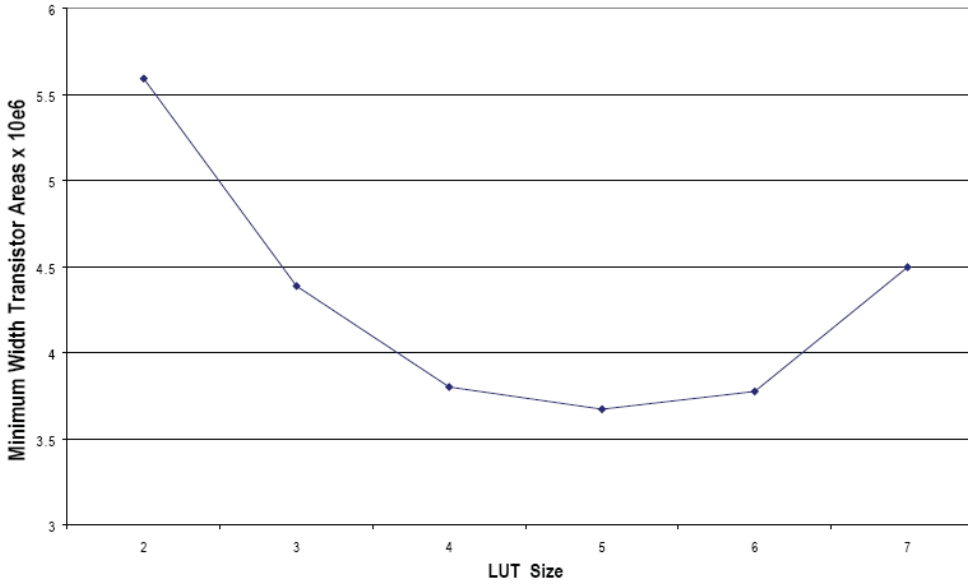
- *How large should the LUT size (K) be?*
- Effects on area & speed
  - Area
    - As *K* increases, fewer logic blocks are needed for a design but area per block increases (LUT's SRAM bits is 2^K)
  - Speed
    - As *K* increases, each critical path contains fewer blocks but delay per block increases

Effect of LUT Size on Area

- As LUT size (K) increases
  - Total FPGA area first decreases and then increases



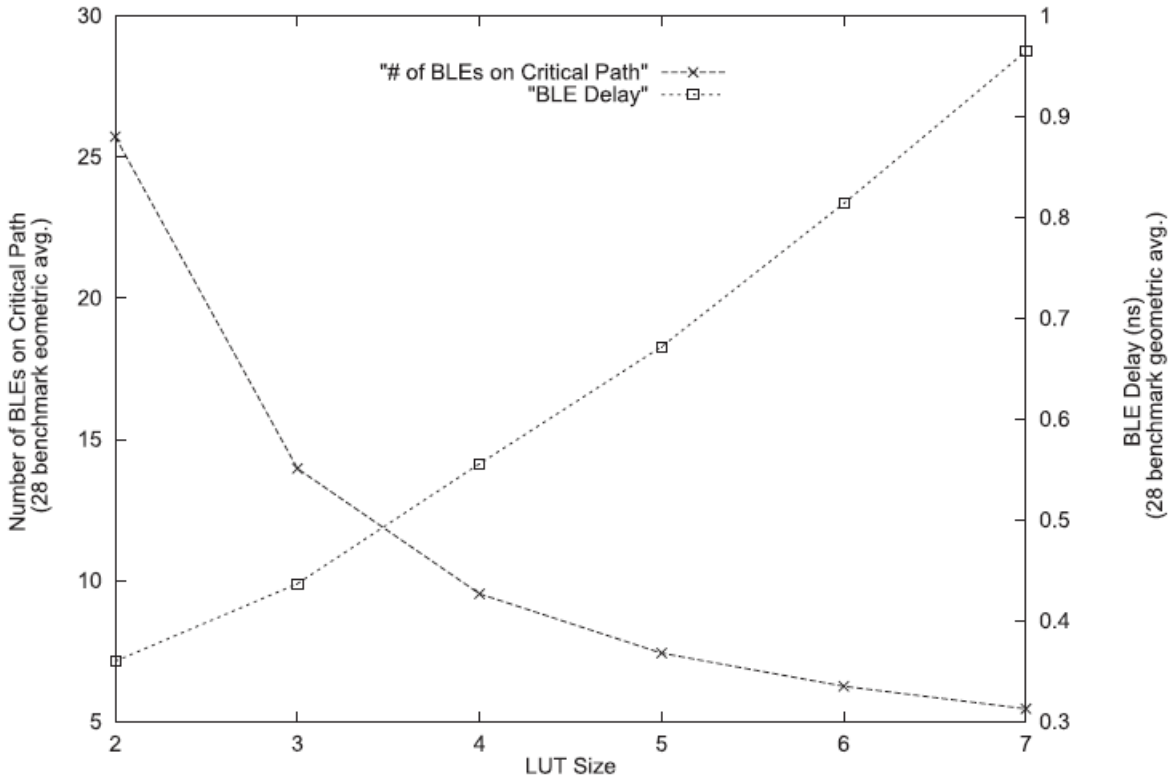
blocks required & area per block for different LUT sizes



Total FPGA area for different LUT sizes

Effect of LUT Size on Speed

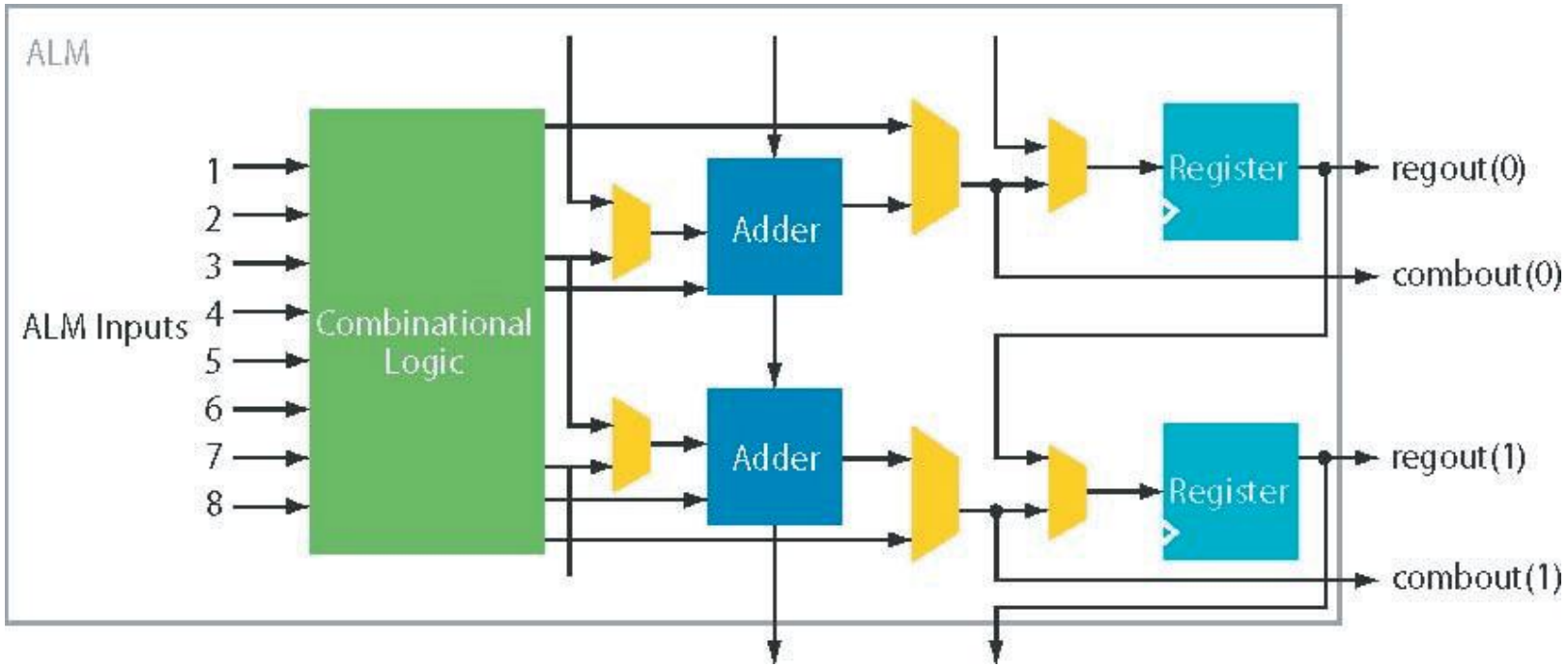
- As *K* increases, each critical path contains fewer blocks but delay per block increases



#LUTs on a critical path & delay per LUT for different LUT sizes

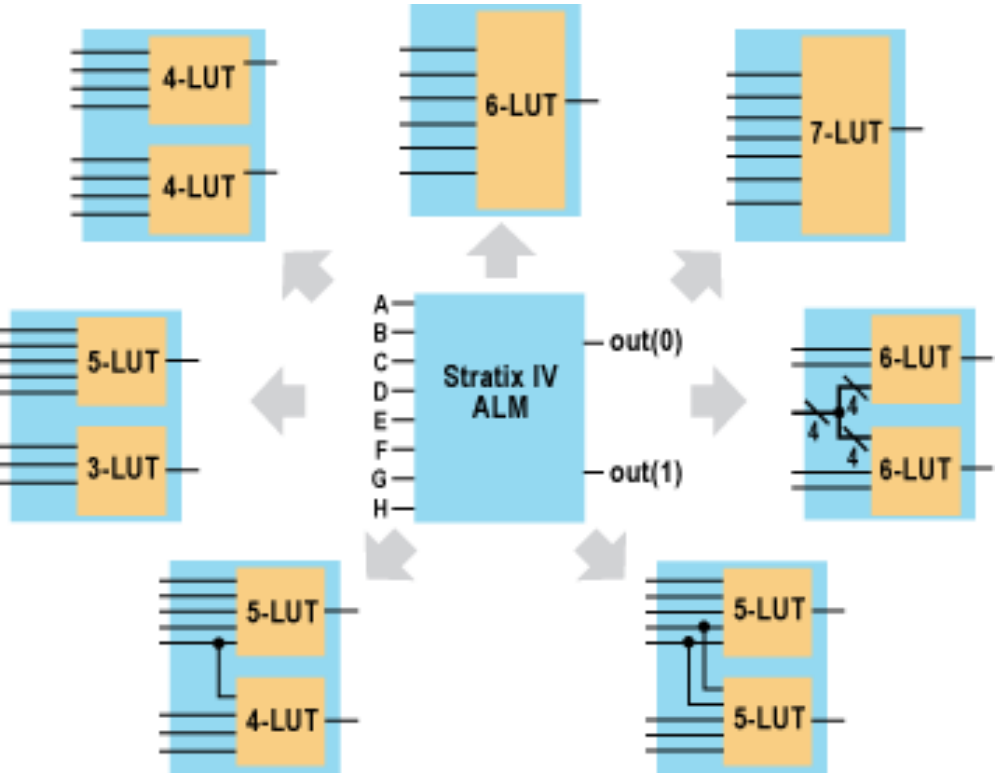
Innovative Idea – Adaptive Logic Module

- Altera Stratix ALM (adaptive logic module)



Flexibility of Adaptive Logic Module

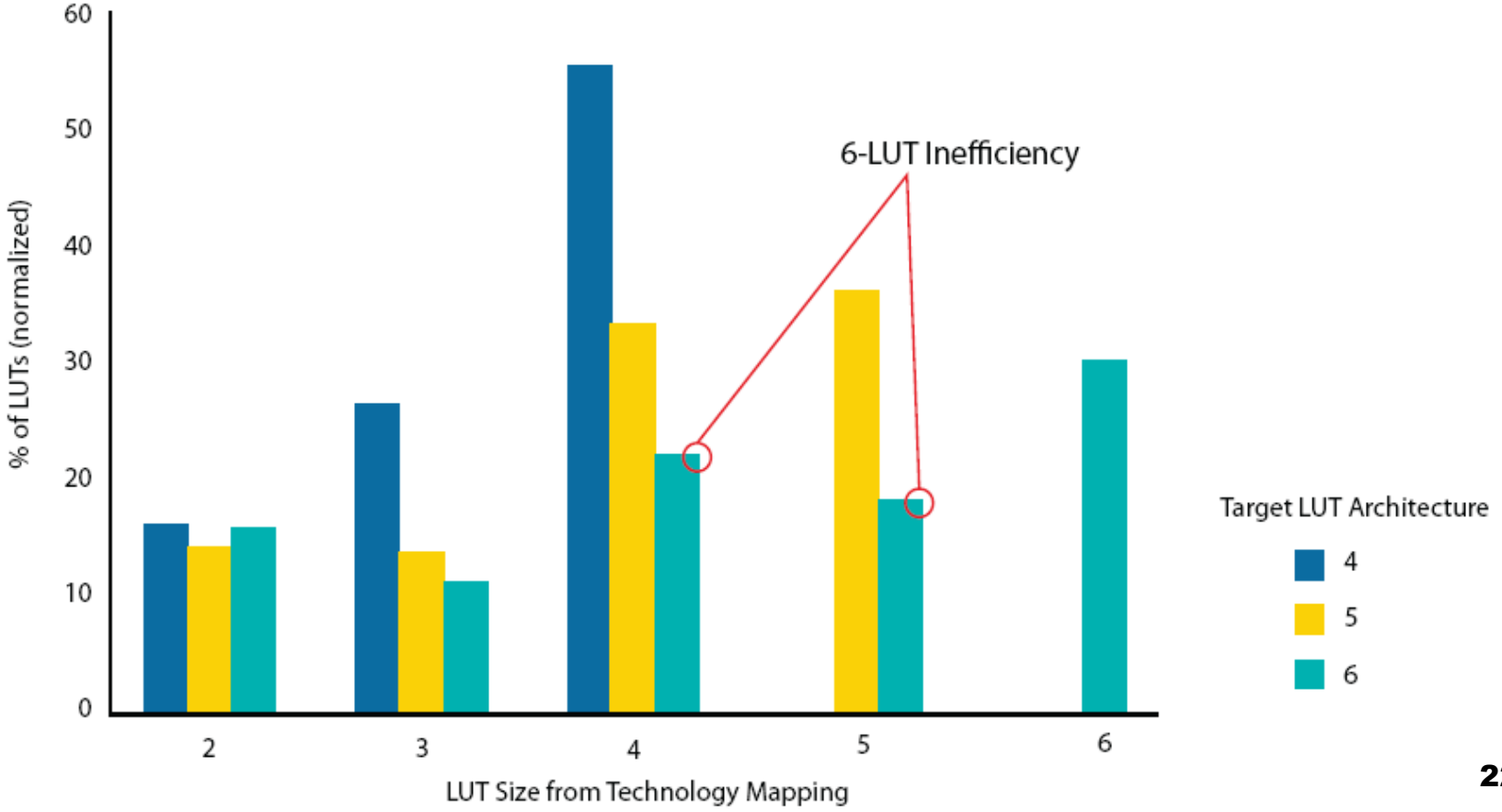
- Fracturable into two



Output 1	Output 2	Shared inputs (min)
6-LUT	-	-
5-LUT	5-LUT	2
5-LUT	4-LUT	1
5-LUT	3-LUT	0
4-LUT	4-LUT	0
4-LUT	3-LUT	0
3-LUT	3-LUT	0

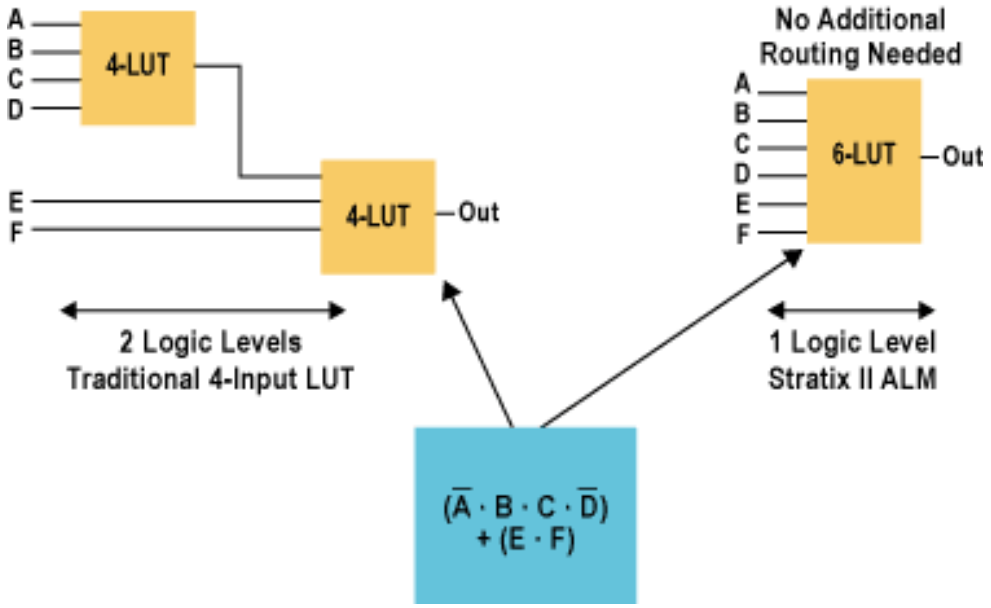
Adaptive Logic Module

- *Observation*: Functions generated by synthesis have different input sizes.



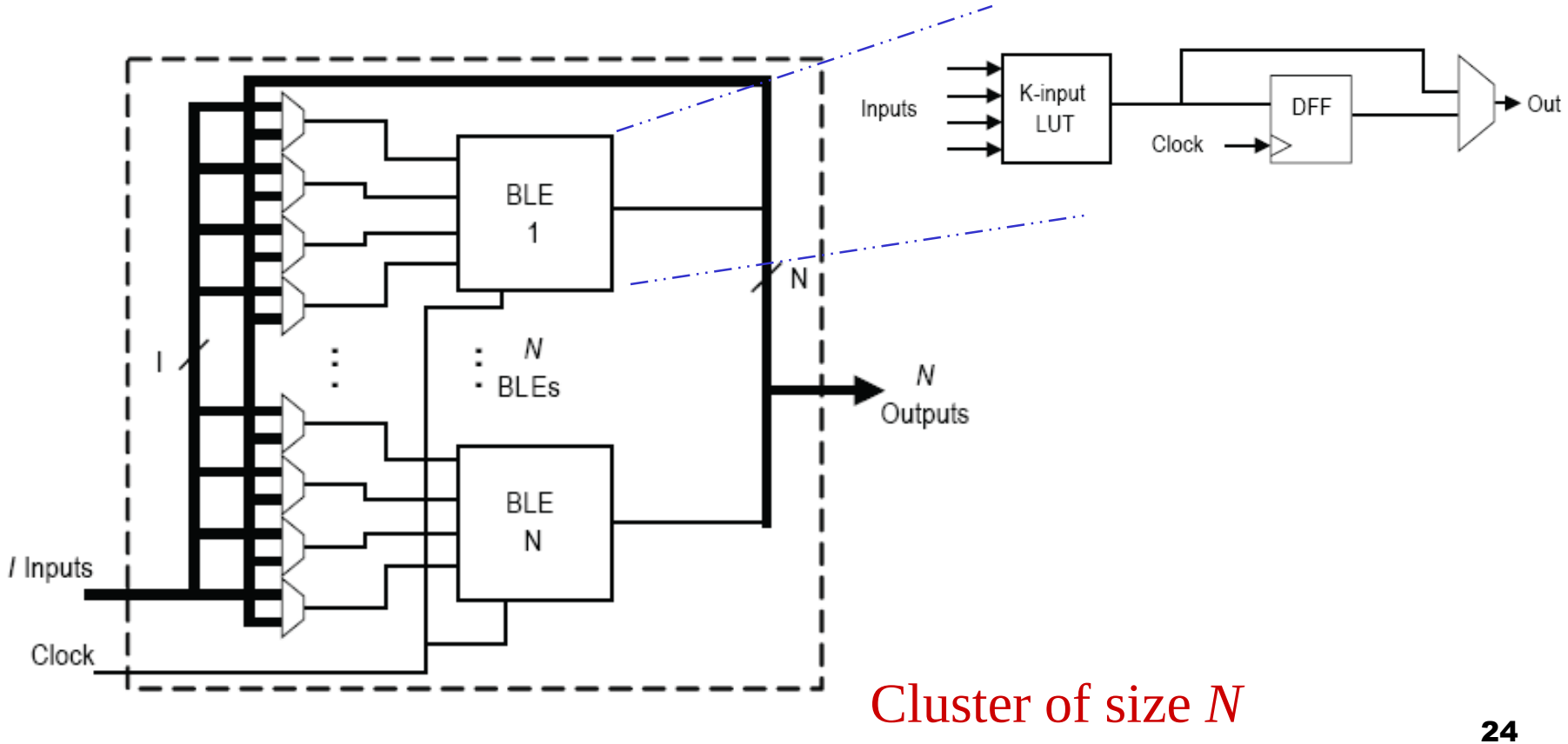
Advantages of Adaptive Logic Module

- ALM-based architecture vs traditional 4-LUT-based architecture
  - Improved area efficiency
  - Improved timing performance



Logic Block Clustering

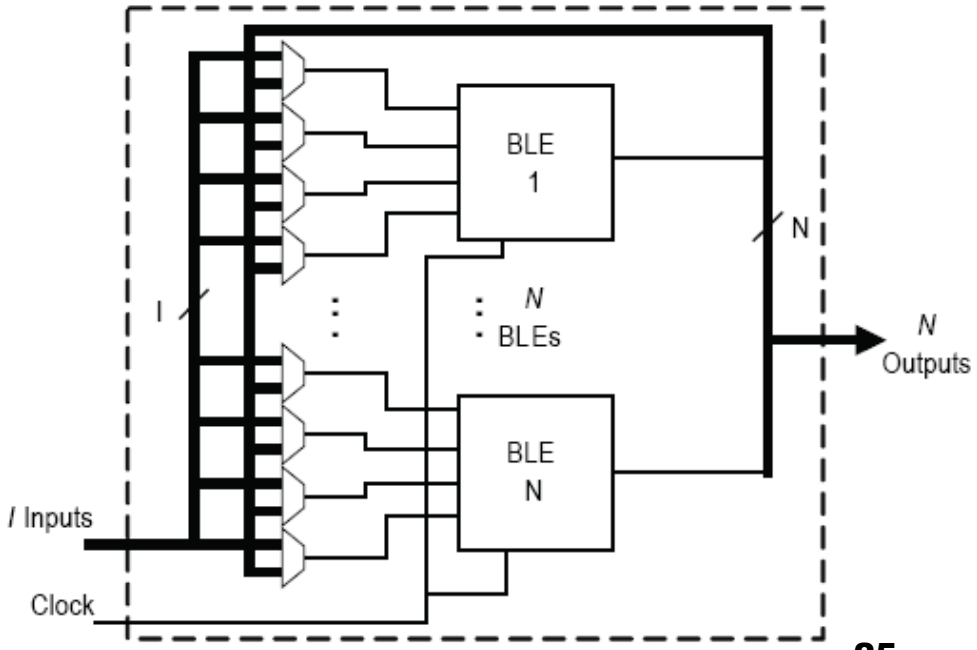
- Logic block made up of a cluster of LUTs and FFs



Cluster of size *N*

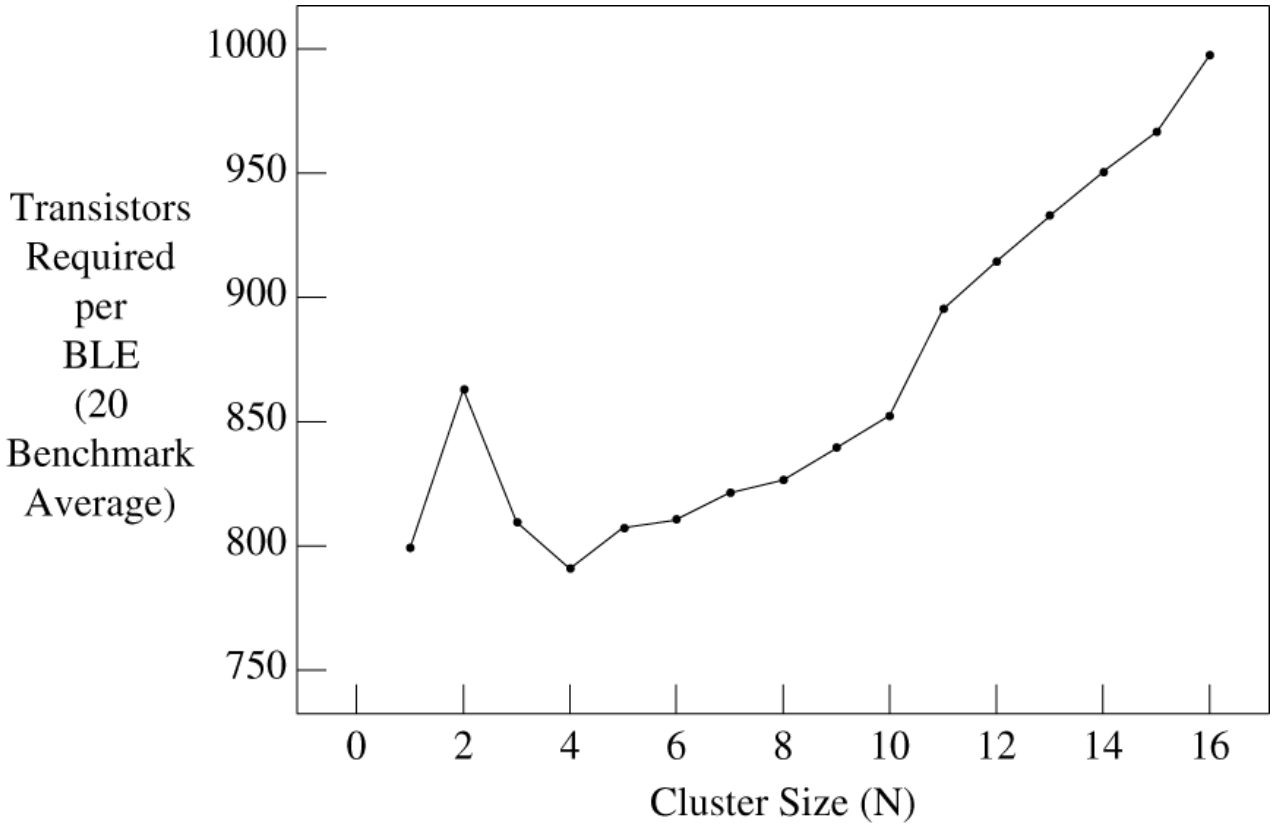
Logic Cluster Study

- *How many cluster input pins (I) are needed?*
  - BLEs in a cluster often share many input signals
  - Empirically, # input pins *I* needed to fully utilize a cluster of *N* *K*-LUT is
    - $I = K(N+1)/2$



Area Efficiency of Different Cluster Sizes

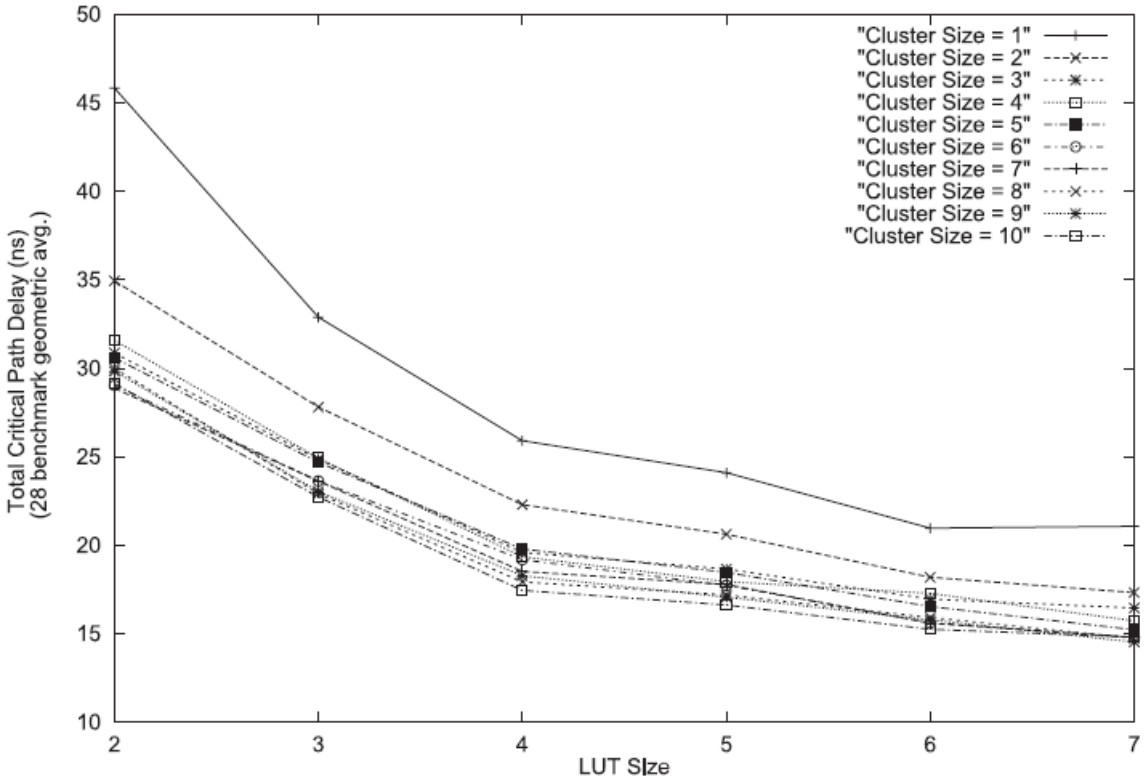
- Clusters in size 1-8 are area-efficient.



Transistors per BLE vs. cluster size (includes overhead circuits)

Effect of Cluster Size and LUT Size on Speed

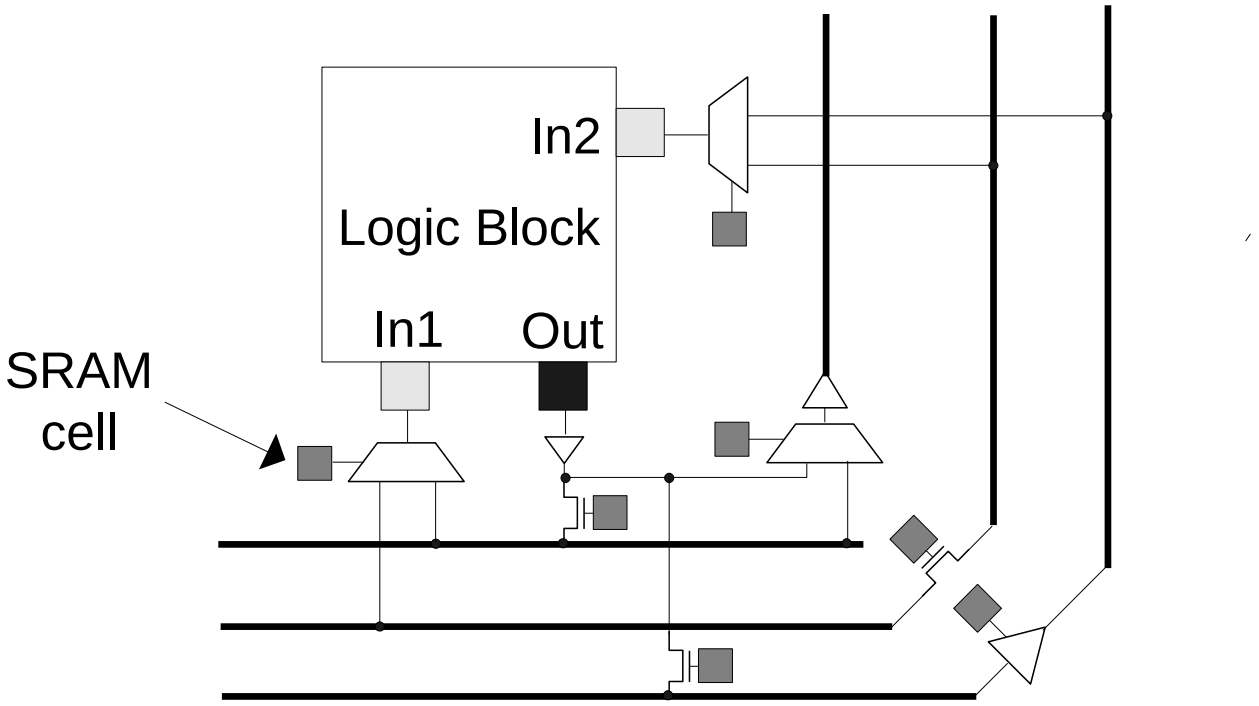
- As LUT and cluster size increase, critical path delay monotonically decreases with diminishing returns
- Significant returns to increase LUT size up to 6 and cluster size up to 3 or 4



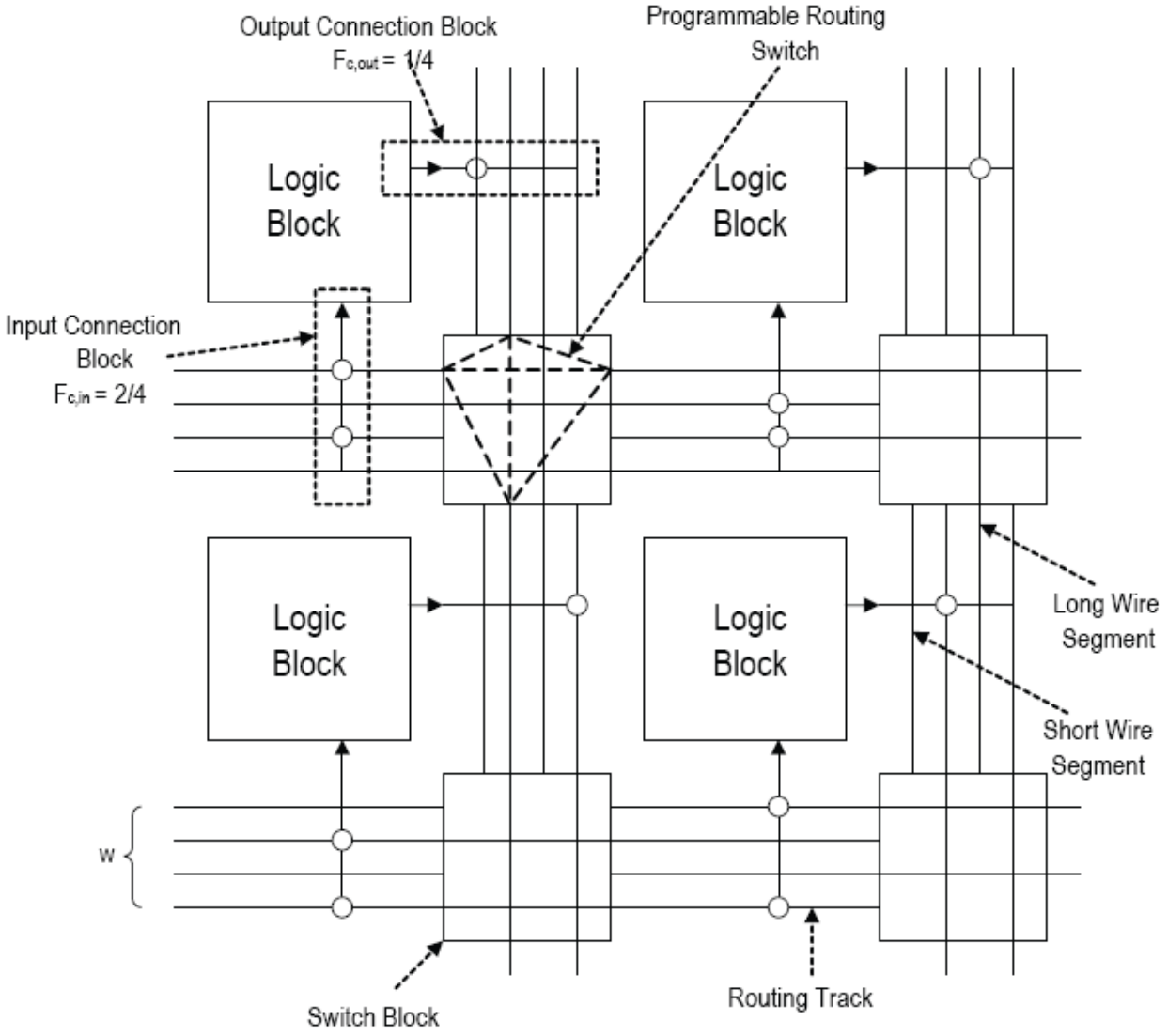
Critical path delay for different LUT and cluster sizes

Programmable Routing

- Programmable switches connect fixed metal wires

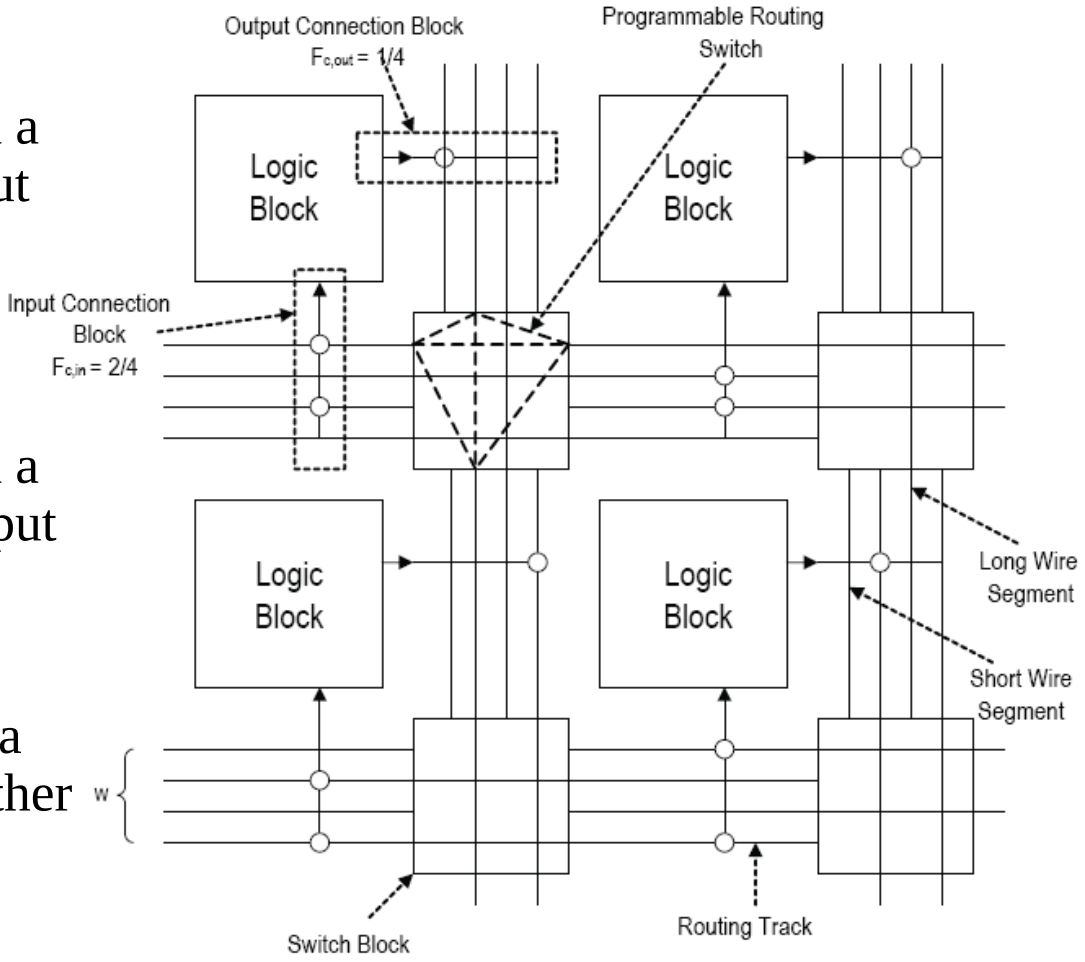


Routing Architecture



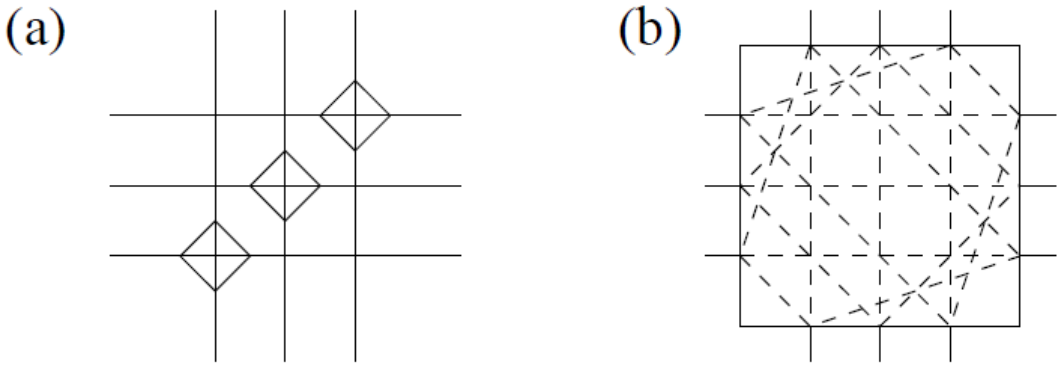
Some Parameters of Routing Architecture

- Input connection block flexibility $F_{c,in}$
  - Fraction of wire segments in a channel connected to an input pin of a block
- Output connection block flexibility $F_{c,out}$
  - Fraction of wire segments in a channel connected to an output pin of a block
- Switch block flexibility F_s
  - No. of possible connections a wire segment can make to other wire segments

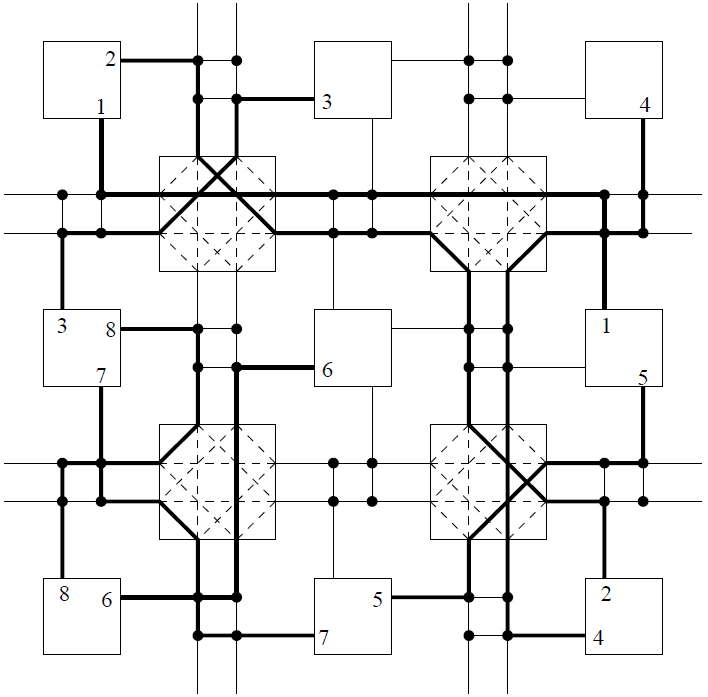


Switch Block Structure

- E.g. Xilinx XC4000 switch block and its abstract representation

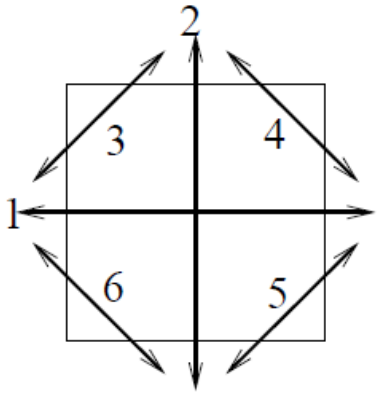


- Switch block structure dictates which wire segments can be connected, and hence influences routability

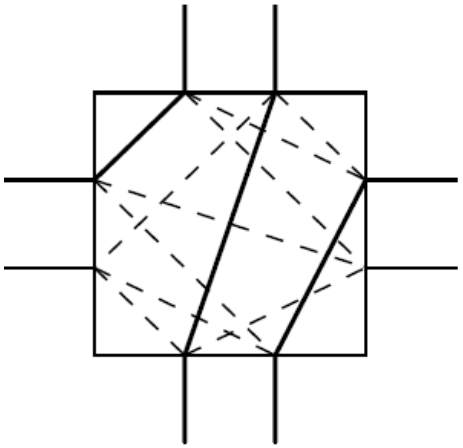


Switch Block Structure

- 6 types of connections

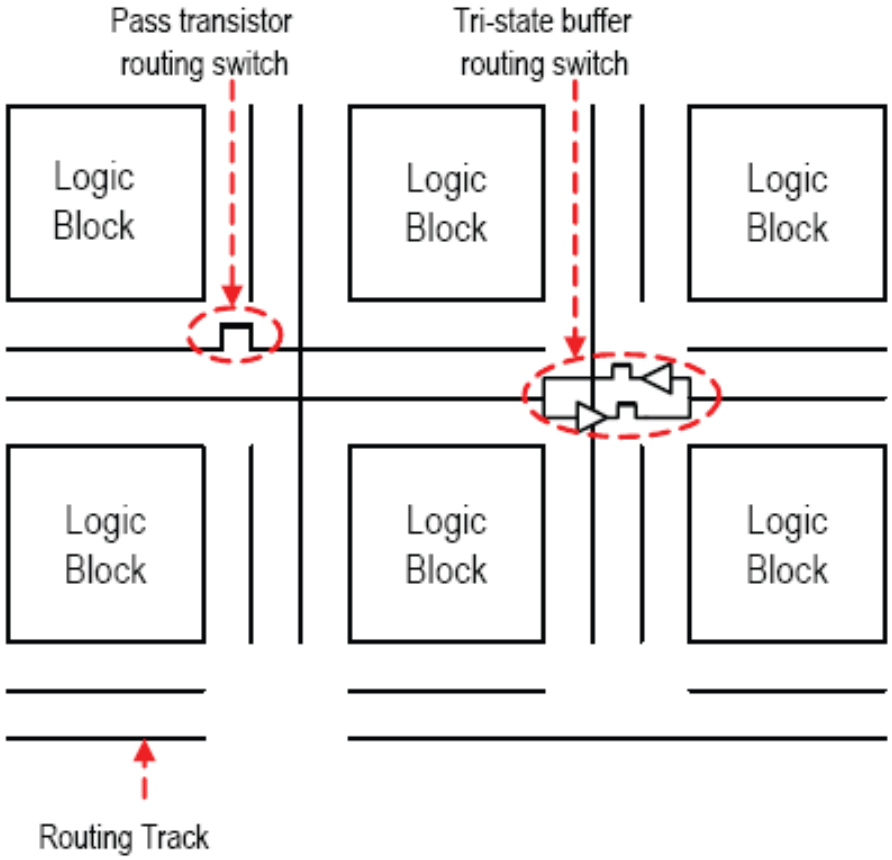


- Routing requirement vector (*n1*, *n2*, *n3*, *n4*, *n5*, *n6*) where *ni* denotes #type-*i* connections
  - e.g. (0,1,1,0,1,0) is routable but (2,0,0,0,0,0) is not with the switch block below
- Want a switch block structure with max no. of routable RRVs



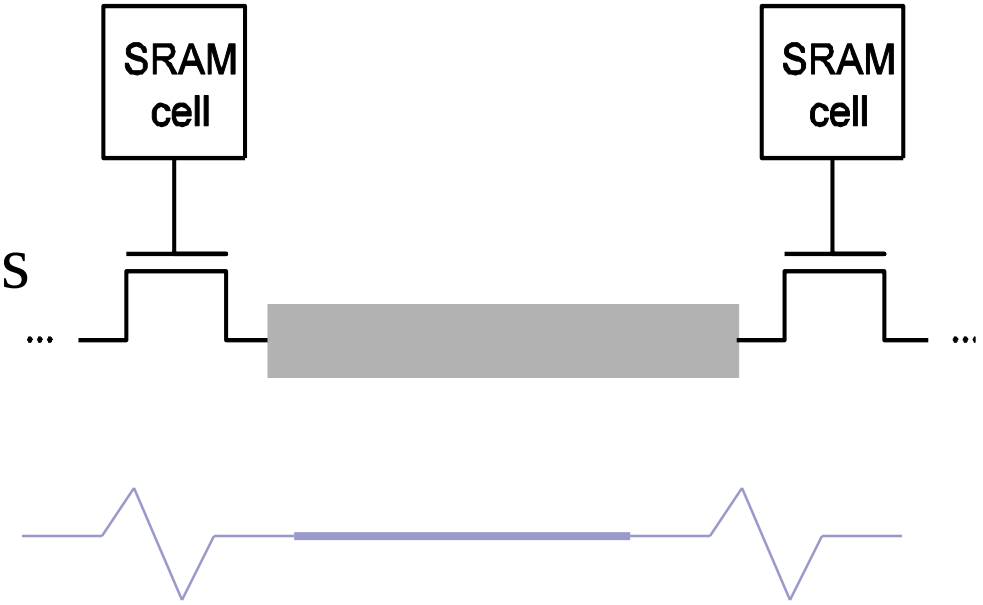
2 Types of Routing Switches

- Typically, mix pass transistor switches & tri-state buffer switches (*why*?)



Pass Transistor Routing Switch

- Small area
- Resistive switch
- Faster for short paths
- Delay grows as the square of no. of switches



Tri-state Buffer Routing Switch

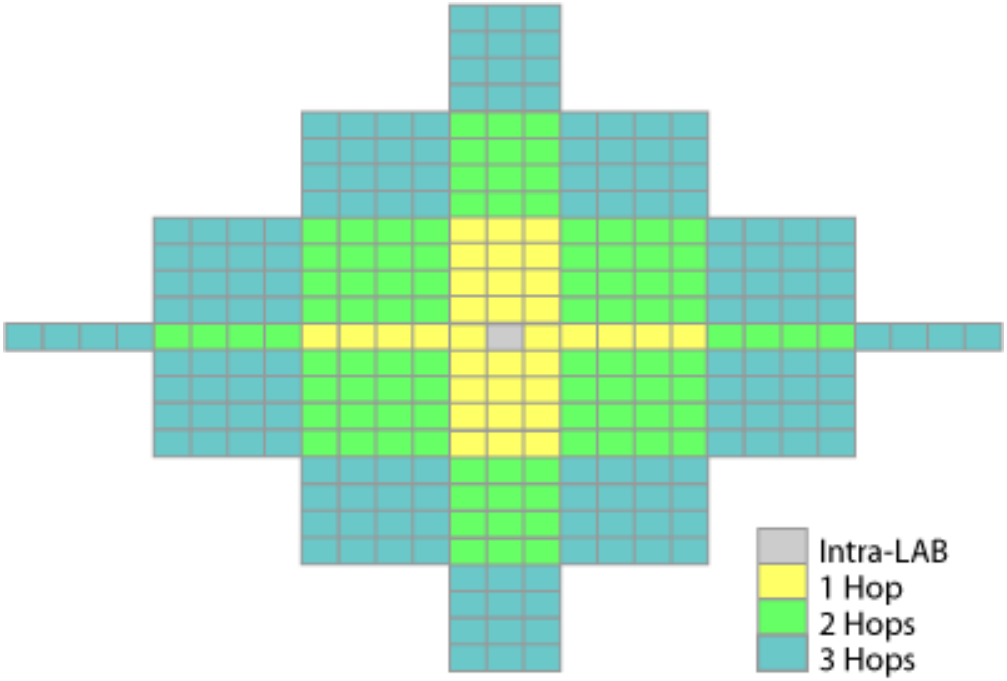
- Larger area
- Regenerative driver
- Faster for long paths passing through many switches
- Delay grows linearly as no. of switches

Other Routing Architecture Factors and Parameters

- Speed, Area and Power also depend on
  - Channel segmentation
  - Transistor size
  - Buffer size
  - Ratio of pass transistor switches & tri-state buffer switches
  - Metal width
  - Wire spacing
  - …

Connectivity

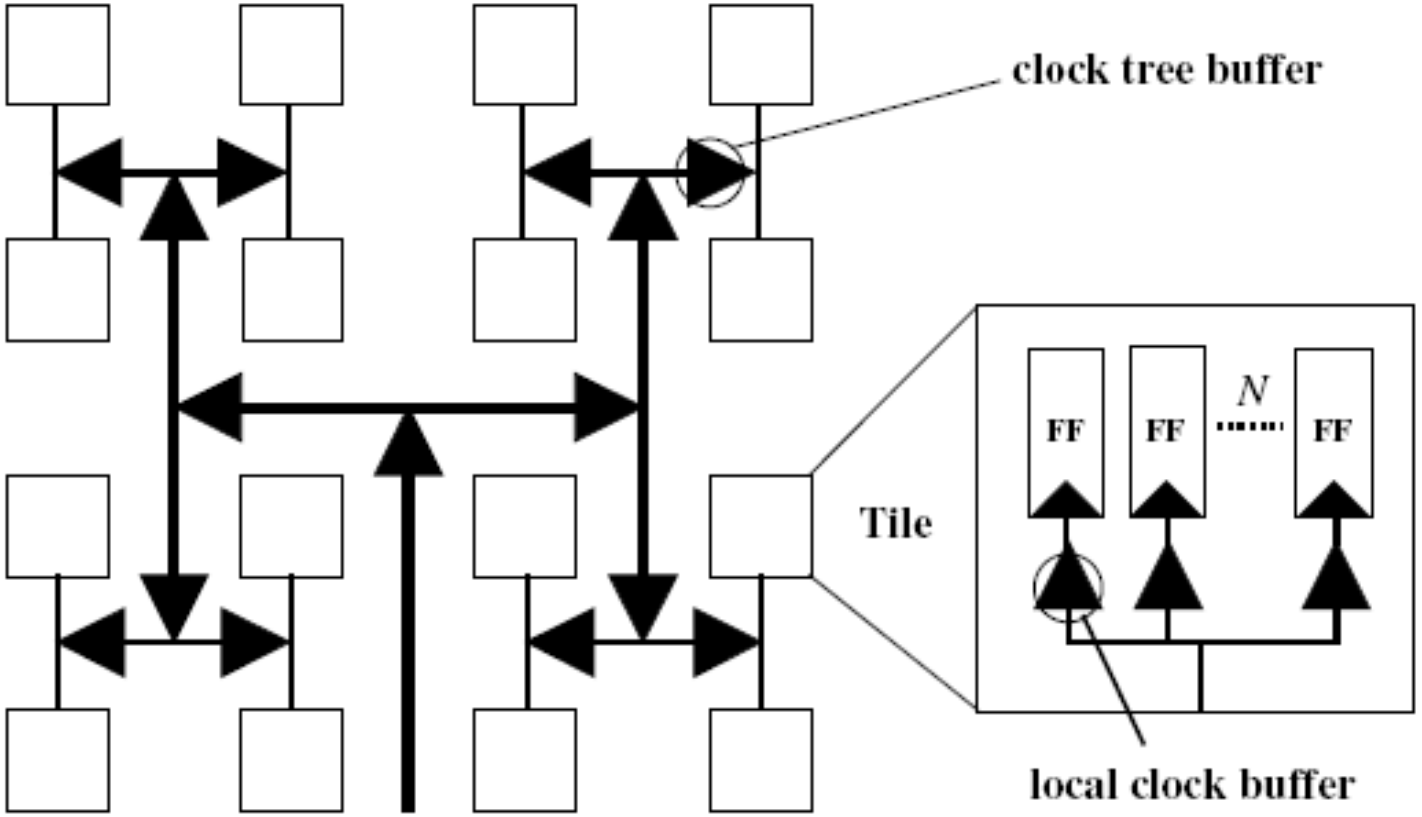
- What is the # hops required to get from one logic block to another?
- Fewer hops → better performance
- More predictable pattern → easier CAD tool optimization



Stratix FPGA series connectivity

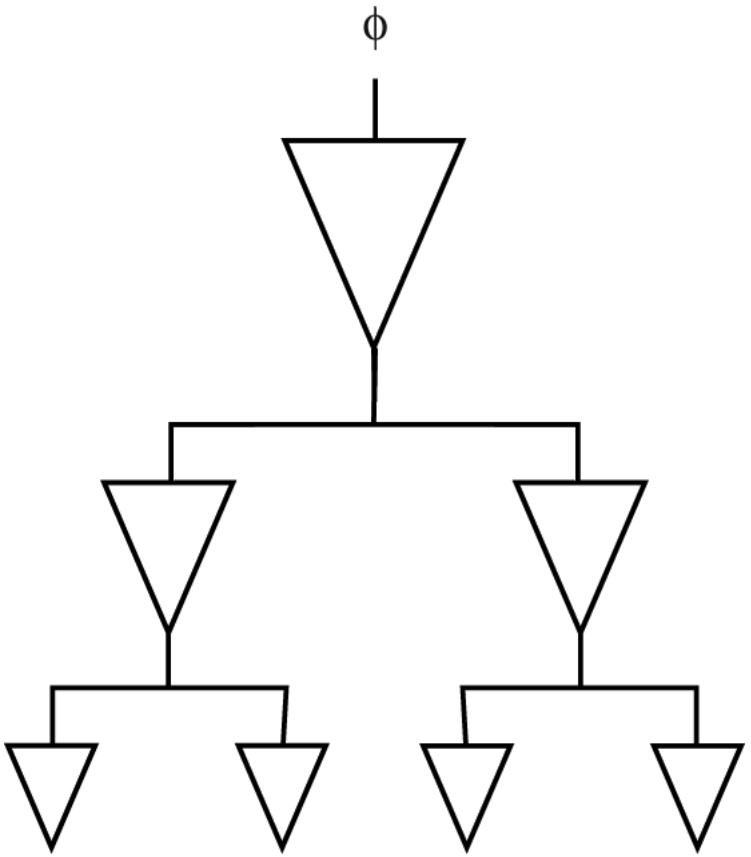
Clock Nets

- Must drive all LEs.



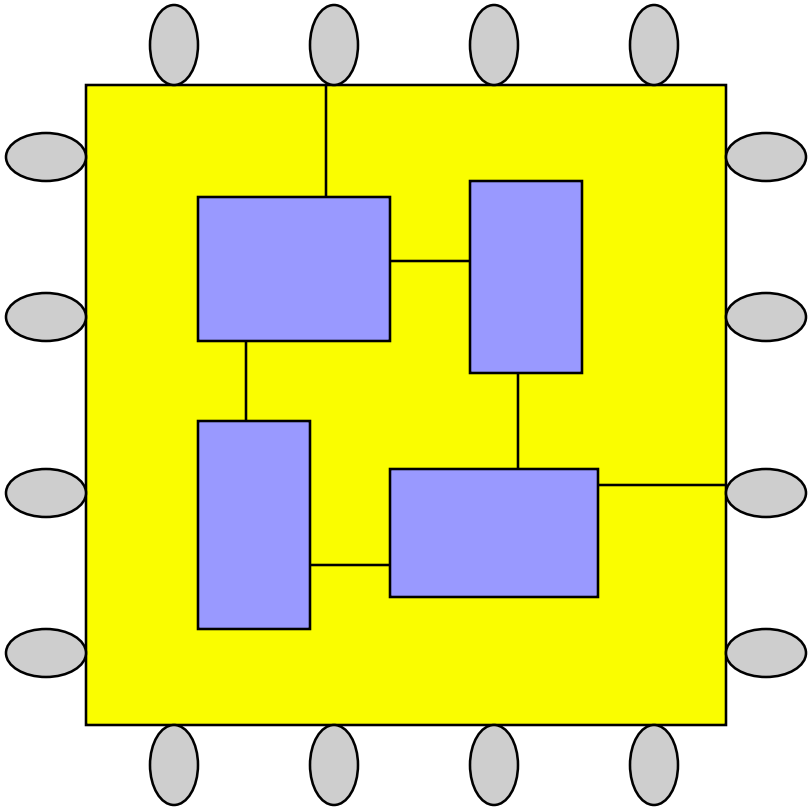
Clock Drivers

- Clock driver tree.
- Determine optimal buffer sizes.



Pinout

- How many pins?
  - Limited by technology.
  - Too much logic, not enough pins means we can't get signals off-chip.



Rent's Rule

- Developed by E. F. Rent (IBM) in 1960.
  - Experimentally derived from sample designs.
- Number of pins vs. number of components is a line on a log-log plot:
  - $N_p = K_p N_s^{\beta}$
- Parameters may vary based on technology:
  - Rent measured $\beta = 0.6$, $K_p = 2.5$.
  - Modern microprocessor has $\beta = 0.455$, $K_p = 0.82$.

FPGAs and Pins

- Chip capacity is growing faster than package pinout.
- Harder to use logic in a multi-FPGA design
  - must try to fit a large function with a small interface into the FPGA
  - may use time-division multiplexing for I/Os

References

- "Flexibility of interconnection structures for field programmable gate arrays", *IEEE J. Solid-State Circuits*, vol. 26(3), 1991.
- "Mixing Buffers and Pass Transistors in FPGA Routing Architectures", in *FPGA'01*.
- "The Effect of LUT and Cluster Size on Deep-Submicron FPGA Performance and Density", *IEEE Trans. VLSI Syst.* 12(3), 2004, 288-298.
- "Improving FPGA Performance and Area Using an Adaptive Logic Module", in *FPL'04*.
- "Leakage Control in FPGA Routing Fabric", in *ASP-DAC'05*.
- "FPGA Architecture: Survey and Challenges", *Foundations and Trends in Electronic Design Automation*, vol.2(2), 2007.
- "VPR 5.0: FPGA CAD and Architecture Exploration Tools with Single-Driver Routing, Heterogeneity and Process Scaling", in *FPGA'09*.
- "Should FPGAs abandon the pass-gate?", in *FPL'13*.
- "FPGA Architecture: Principles and Progression", *IEEE Circuits and Systems Magazine*, vol.21(2), 2021.