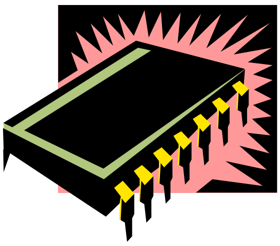
Power Optimization Techniques for FPGAs

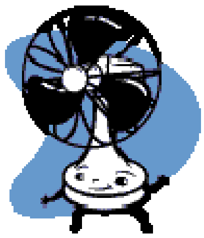
Outline

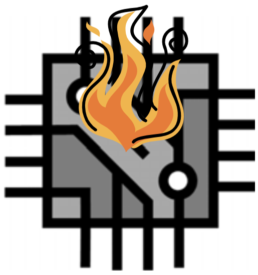
- Introduction
- Hardware Techniques
  - Selectable Core Voltage
  - Programmable Power Mode of Individual Tiles
- EDA Solutions
  - Dynamic Power Optimization in LUTs with Unused Input(s)
  - Leakage Power Optimization by LUT Output Polarity Selection
  - Leakage Power Optimization by LUT Input Vector Reordering
  - Power-Driven Synthesis, Place & Route
  - Clock Power Reduction by Power-Aware Placement and Clock Shutdown
  - Glitch Power Reduction by Don't Care Assignment
- Hardware + EDA
  - Interconnect Power Reduction by Effective Interconnect Capacitance Optimization

Introduction

- Power consumption is a key concern today.
- Reducing power will
  - Lower packaging cost and cooling costs
  - Improve reliability
  - Lengthen the battery life of mobile device

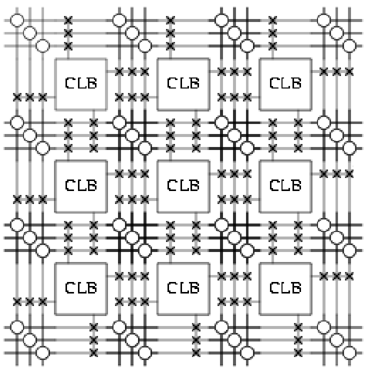






Introduction

- FPGA's programmability incurs extra power overhead in
  - More transistors are needed to implement a logic function than custom ASIC
  - Longer wire lengths
  - Inclusion of programmable routing switches



Power Reduction Techniques

- Combination of techniques to reduce
  - Dynamic power
  - Static power
- Combination of hardware techniques and EDA solutions

Dynamic Power vs Leakage Power

- Two major sources of power dissipation
  - Dynamic power – caused by signal transition
  - Static (leakage) power – caused by leakage currents in off transistors
- Dynamic power: $P_{avg} = \frac{1}{2} \sum_{i \,\in\, signals} C_i \cdot f_i \cdot V^2$
- Leakage power
  - proportional to transistor count
  - dependent on supply voltage and threshold voltage

Outline

- Introduction
- Hardware Techniques
  - Selectable Core Voltage
  - Programmable Power Mode of Individual Tiles
- EDA Solutions
  - Dynamic Power Optimization in LUTs with Unused Input(s)
  - Leakage Power Optimization by LUT Output Polarity Selection
  - Leakage Power Optimization by LUT Input Vector Reordering
  - Power-Driven Synthesis, Place & Route
  - Clock Power Reduction by Power-Aware Placement and Clock Shutdown
  - Glitch Power Reduction by Don't Care Assignment
- Hardware + EDA
  - Interconnect Power Reduction by Effective Interconnect Capacitance Optimization

Selectable Core Voltage

- Selectable core voltage allows user to choose lower core voltage if performance can be met
- Dynamic power:

$$P_{avg} = \frac{1}{2} \sum_{i \,\in\, signals} C_i \cdot f_i \cdot V^2$$

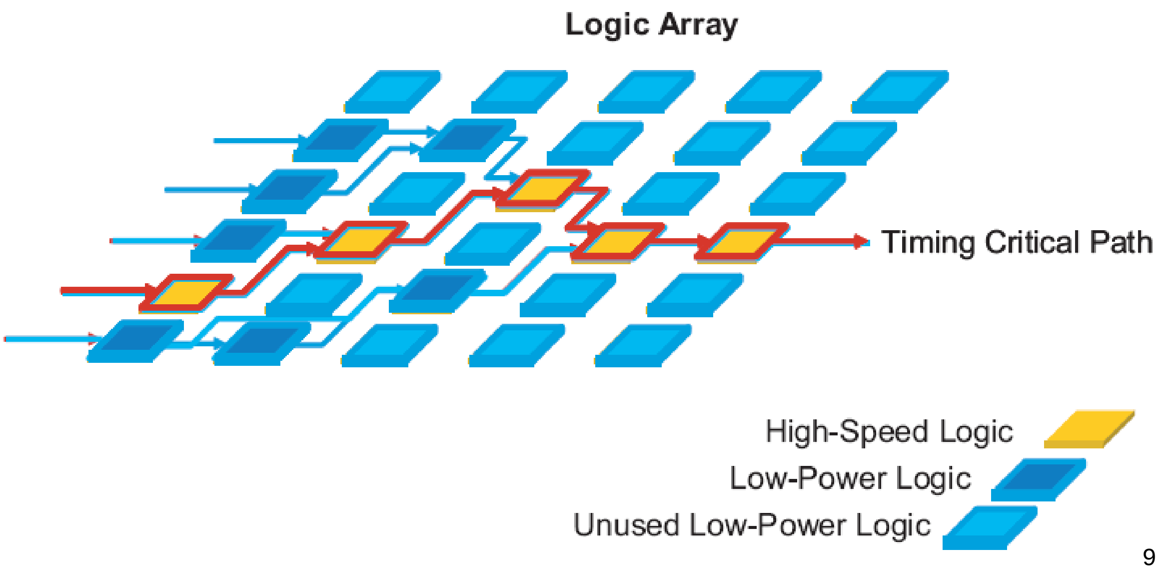
- Lower supply voltage reduces
  - dynamic power (quadratically)
  - Leakage power (more than quadratically)

Table 2. Stratix III Power Compared to Stratix II Power Across Selectable Core Voltage

Core Voltage	Dynamic Power Reduction From 1.2V	Static Power Reduction From 1.2V
1.1V	33%	52%
0.9V	55%	64%

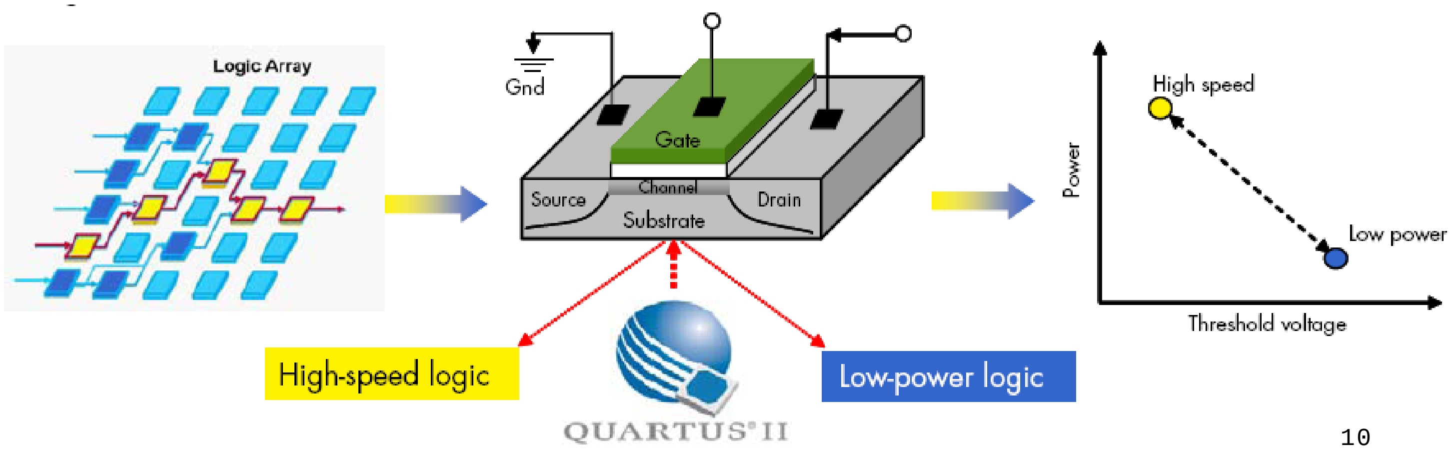
Programmable Power Technology in FPGA

- Only a small percentage of logic is timing-critical
- Reduce leakage power by running non-timing critical logic on low-power mode



Programmable Power Technology in Stratix Series (since Stratix III)

- Timing analysis determines the slack available in each path of the circuit
- Individual tile programmability between high-performance and low-power modes

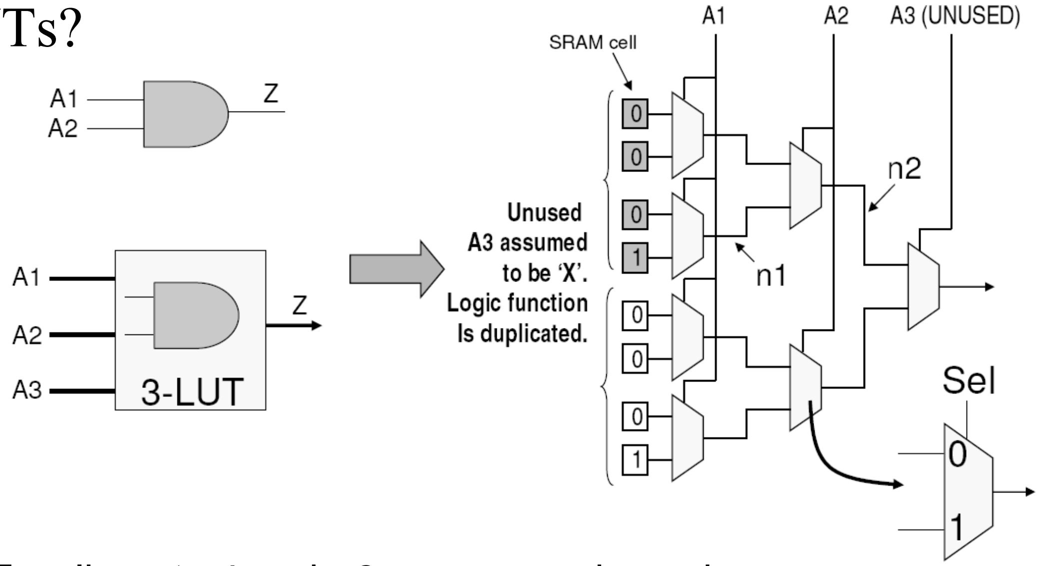


Outline

- Introduction
- Hardware Techniques
  - Selectable Core Voltage
  - Programmable Power Mode of Individual Tiles
- EDA Solutions
  - Dynamic Power Optimization in LUTs with Unused Input(s)
  - Leakage Power Optimization by LUT Output Polarity Selection
  - Leakage Power Optimization by LUT Input Vector Reordering
  - Power-Driven Synthesis, Place & Route
  - Clock Power Reduction by Power-Aware Placement and Clock Shutdown
  - Glitch Power Reduction by Don't Care Assignment
- Hardware + EDA
  - Interconnect Power Reduction by Effective Interconnect Capacitance Optimization

Dynamic Power Optimization in LUT with Unused Input(s)

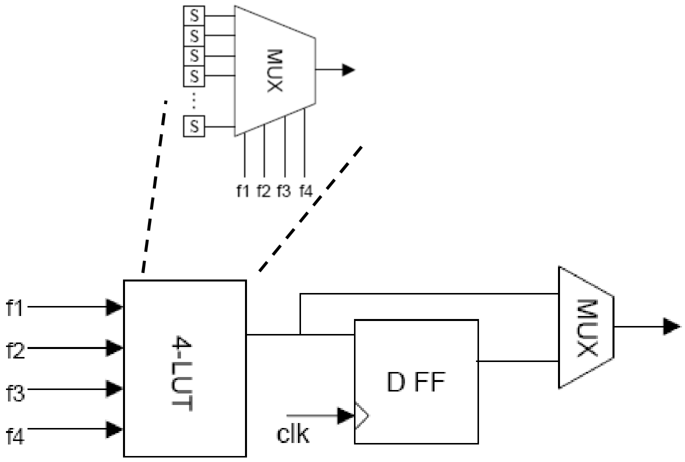
- A mapped design has many LUTs with unused input(s)
- How to optimize dynamic power consumption of such LUTs?



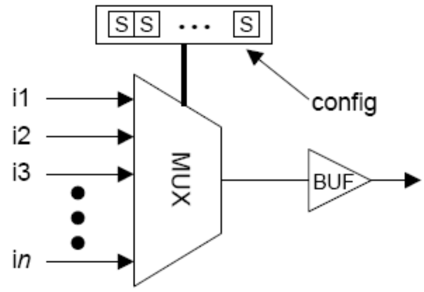
• Toggling at *n1* and *n2* consumes dynamic power.
• Setting shaded cells to logic-0 and A3 to 1 will eliminate unnecessary switching.

Leakage Power in FPGA

- Many MUXes and buffers in FPGA, they consume leakage power



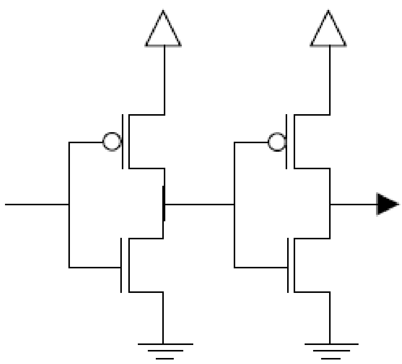
Logic block



Routing switch

Buffer Leakage Characteristic

- Buffer leakage power is smaller when input = 1
  - due to different leakage characteristics of N and P transistors and transistor sizing for delay

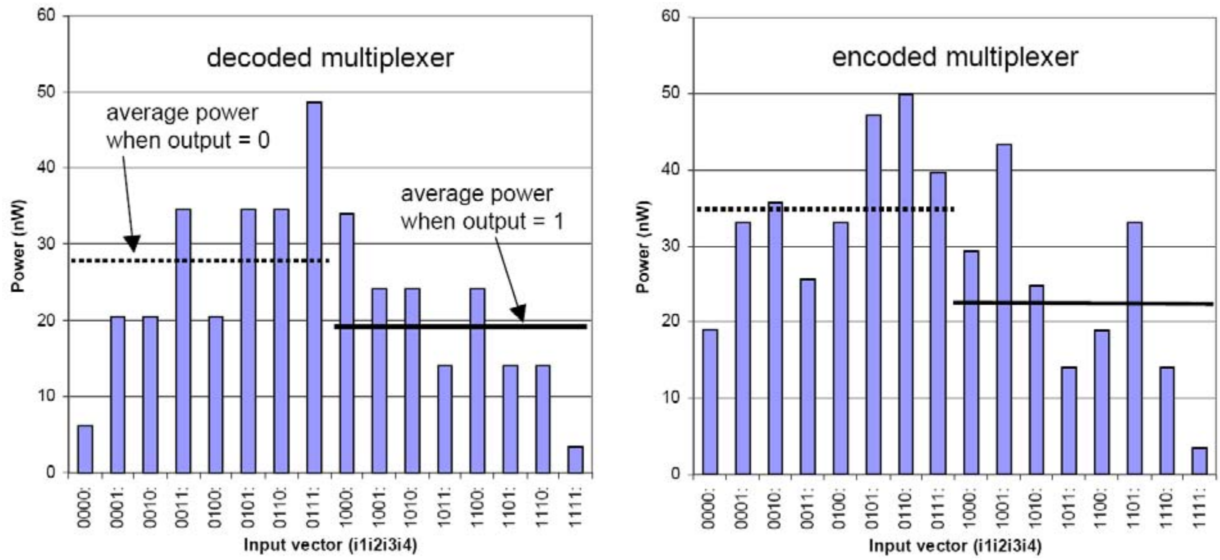


Input	Power (nW)
0	56.1
1	46.6

MUX Leakage Characteristic

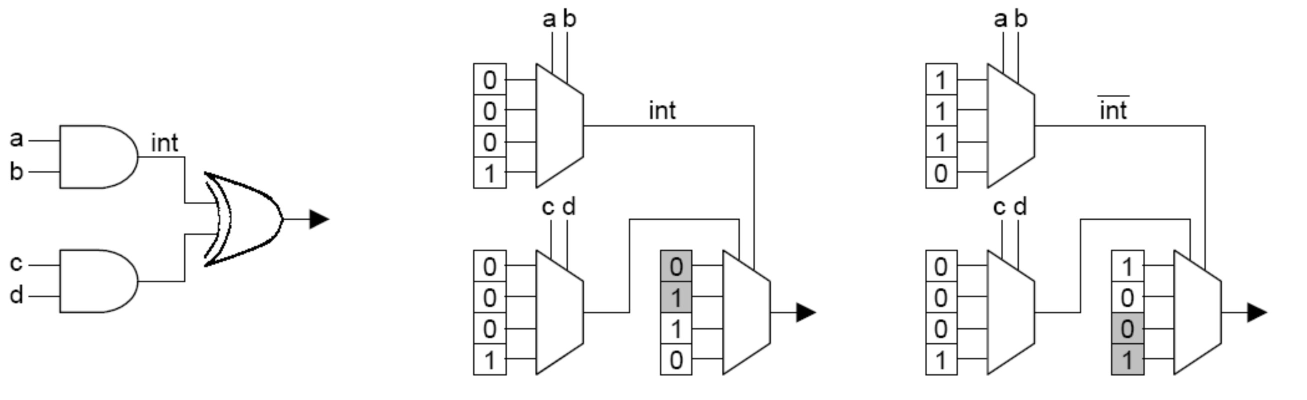
- MUX leakage power is smaller when output = 1

When input i1 is passed to the output:



Leakage Power Optimization by LUT Output Polarity Selection

- Want signals to spend most of their time in logic 1 state
- Signals spending more time in logic 0 state are candidates for inversion
- Most signal can be inverted like below:



a) original circuit b) 2-LUT implementation c) after signal inversion

Polarity Selection Algorithm for Leakage Power Optimization

function OptimizeLeakage(design, signal static probabilities)

  for each signal *n* in the design **do**

    if static_probability(*n*) < 0.5 **then**

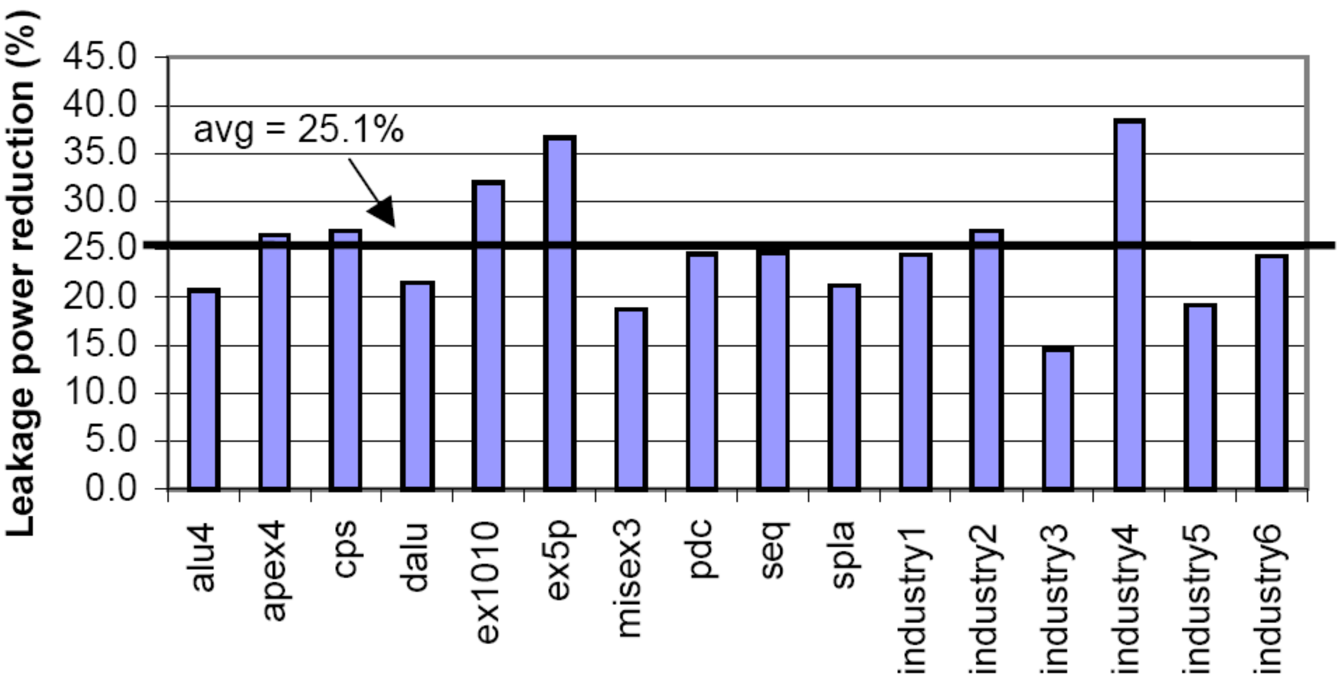
      if signal *n* can be inverted **then**

        invert(*n*)
        // FPGA is re-programmed; *n* replaced with $\overline{n}$

  return new design

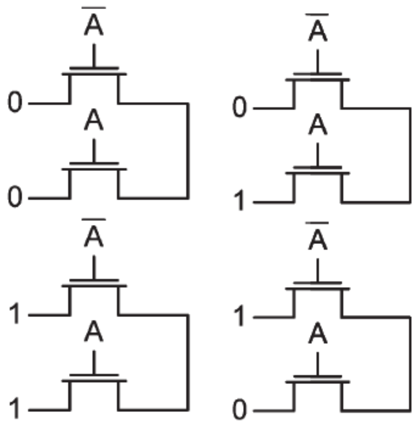
Experimental Results

- Leakage power reduction by polarity selection



Leakage Characteristic of MUX Transistor Pair

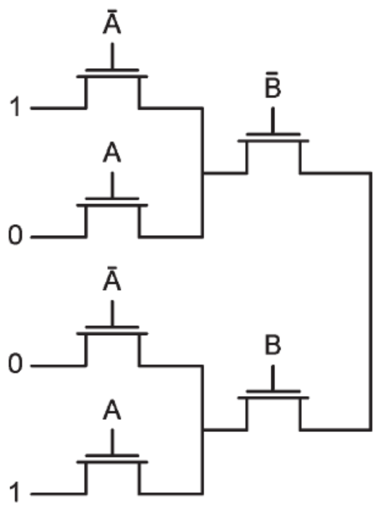
- Leakage of transistor pair in a MUX depends on values of input pair
  - (a) shows low-leakage multiplexer configurations
  - (b) shows high-leakage multiplexer configurations

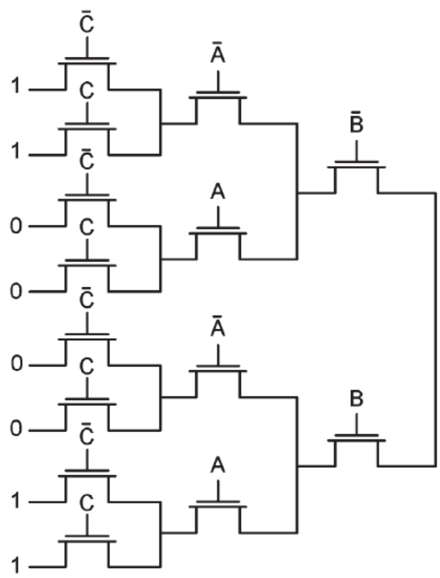


(a) (b)

Leakage Power Optimization by LUT Input Vector Reordering

- How to optimize leakage power for LUT with unused input(s)?

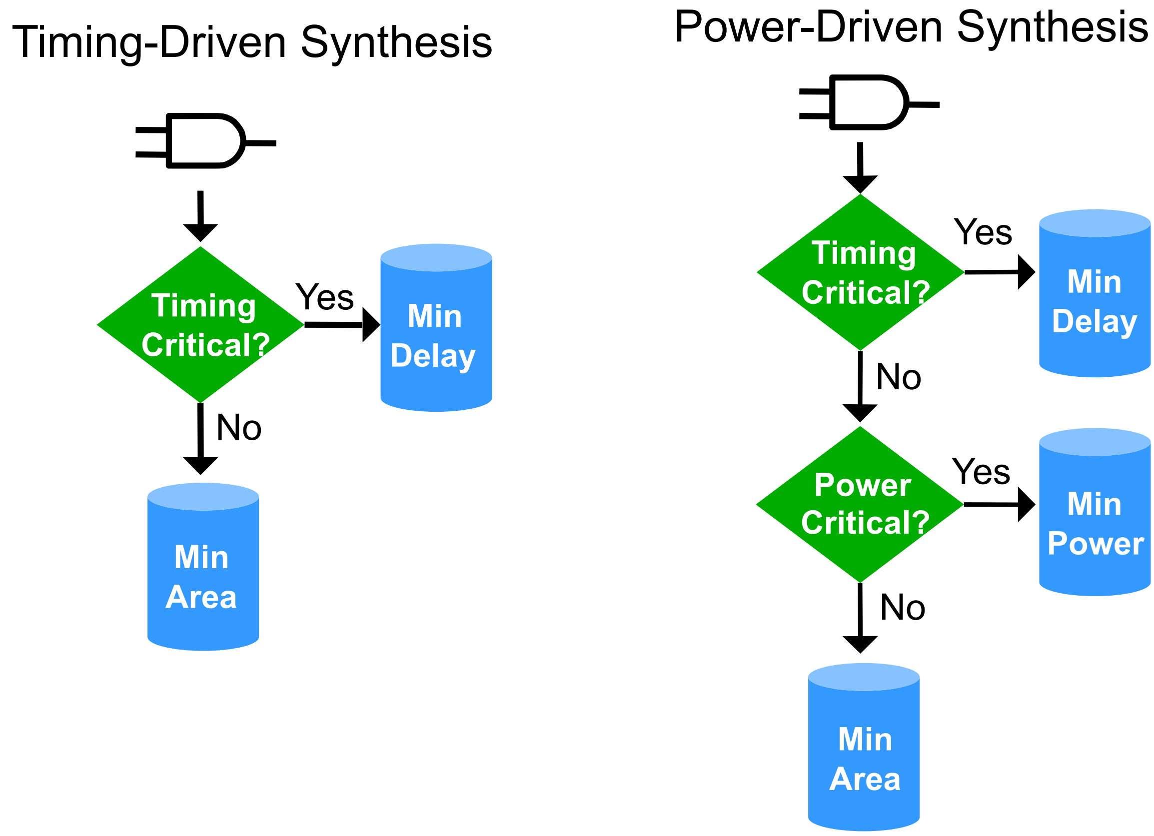




(a) A 3-LUT with one unused input.

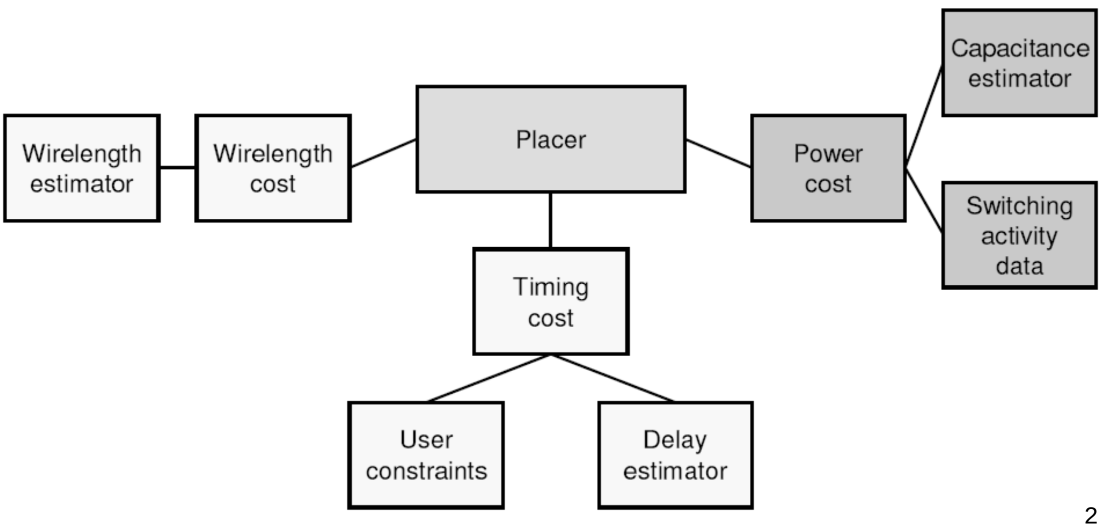
(b) Input padding to create largest # of low-leakage transistor pairs.

Power-Driven Synthesis



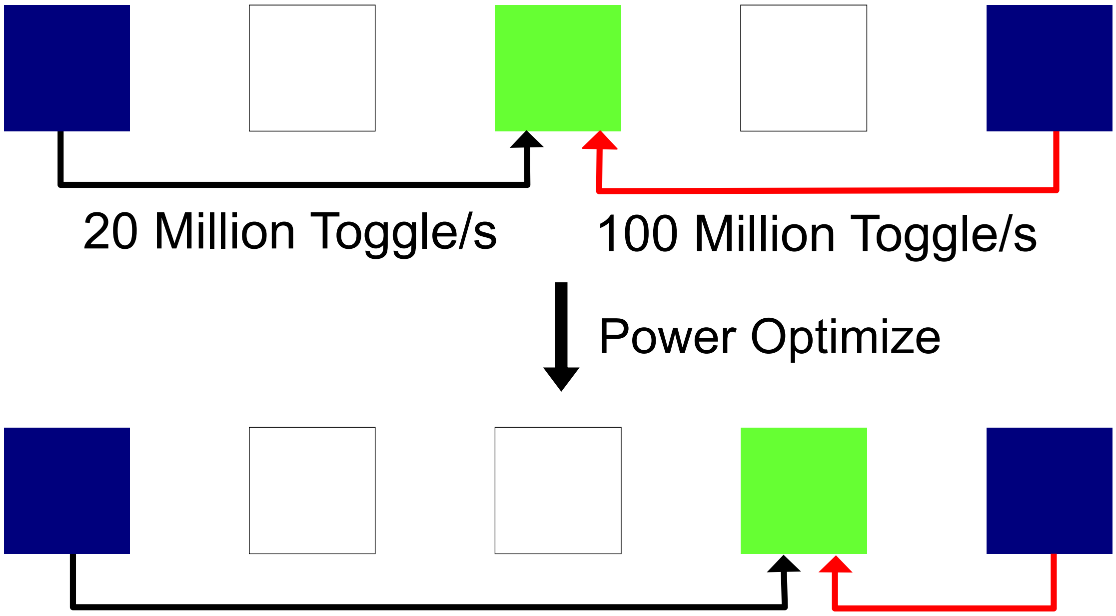
Power-aware Placement

- Use cost function including estimated dynamic power: $Cost = a \cdot W + b \cdot T + c \cdot P_{avg}$
- Dynamic power consumption of a signal estimated based on its switching activity, fanouts, X-span and Y-span.



Power-Driven Place & Route

- Minimize capacitance of high-toggling signals
- Without violating timing constraints

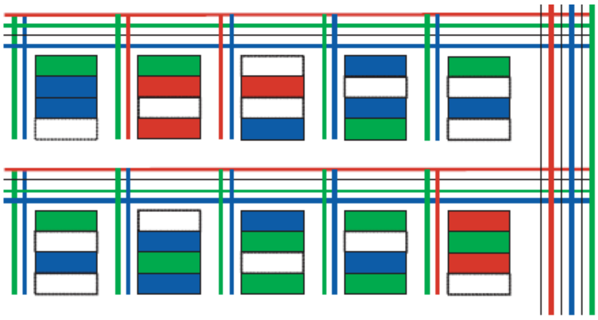


Power-Driven Routing

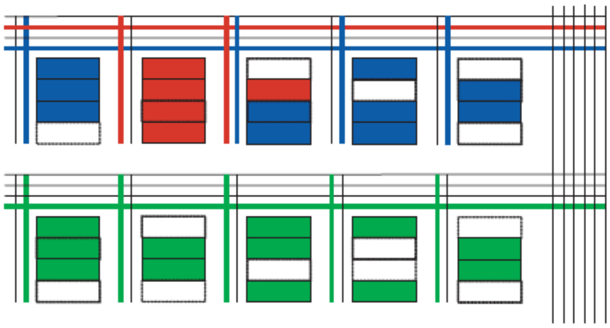
- Timing-critical nets
  - route with minimum delay
- Non-timing-critical nets
  - route with a cost considering capacitance and switching activities
- In iterative negotiation-based routing
  - high activity nets are given preference to retain low-capacitance routing resources

Reducing Clock Power by Power-aware Placement and Shutdown of Clocks

- Shut down unused clock signals to reduce power
- Group logic with common clock into same LAB in power-driven placement



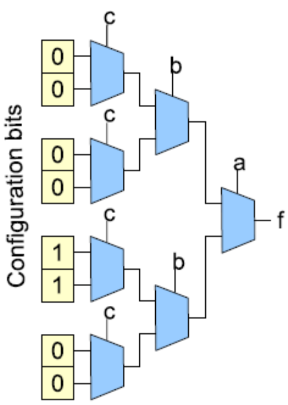
Clocking with a timing-driven placement

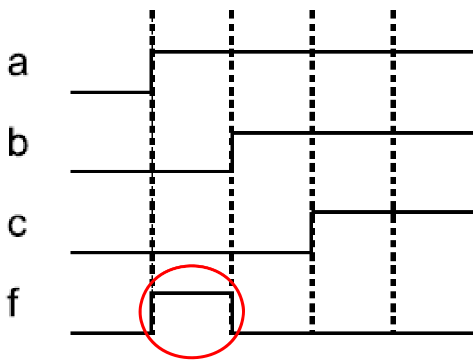


Clocking with a power-driven placement

Glitch Power

- *Glitches* at gate output are unwanted signal transitions due to unbalanced arrival times at gate inputs.
- E.g. Input transition from 000 to 111:

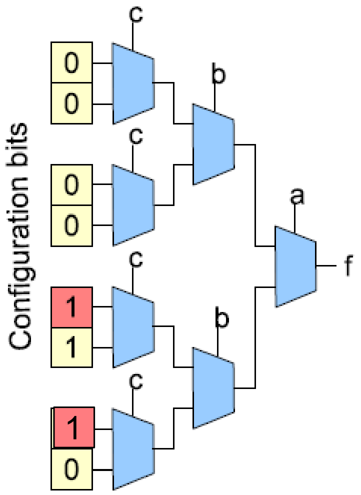




- For FPGA, glitch power accounts for a significant portion of dynamic power (>20%)

Don't Cares in Logic Circuit

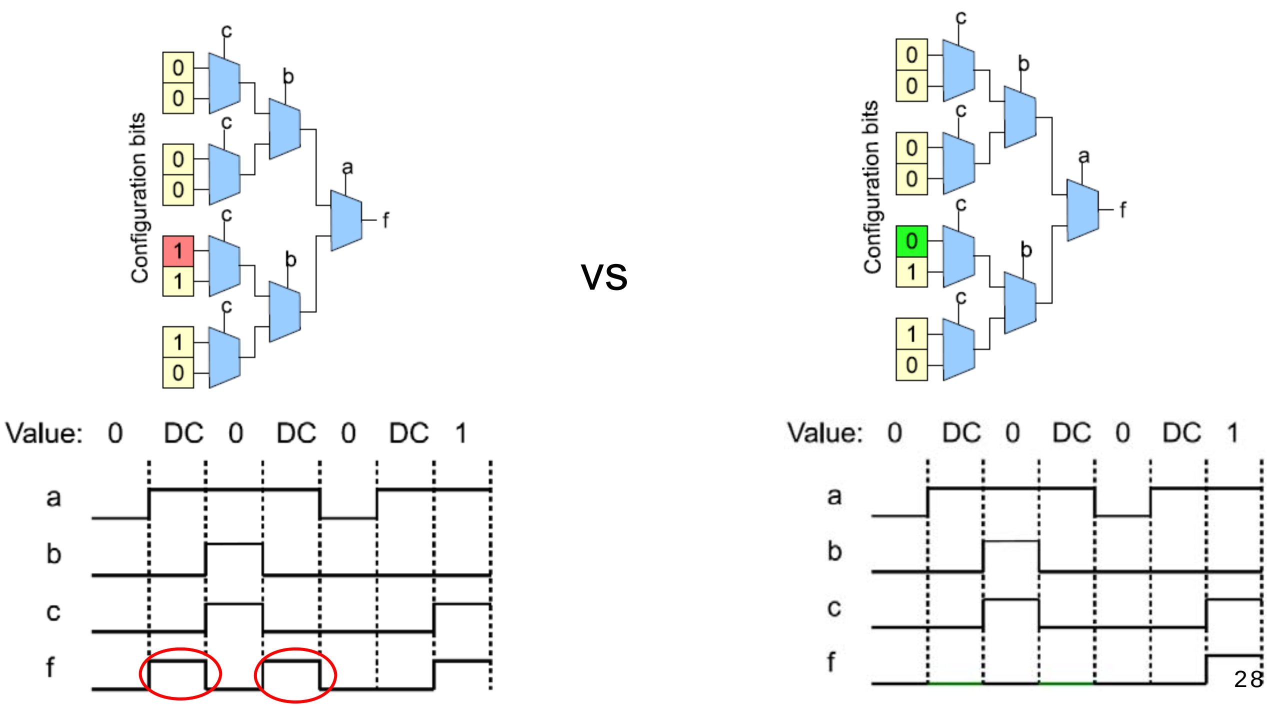
- A mapped LUT may have *don't care* entries
- Don't care entry: an input pattern can never occur or output cannot propagate to POs
- E.g.



abc	f	Care
000	0	Y
001	0	Y
010	0	Y
011	0	Y
100	1	N
101	1	Y
110	0	N
111	0	Y

Glitch Reduction by Don't Care Assignment

- Glitch reduction by proper logic value assignment for don't cares (use a simple majority vote heuristic)



Outline

- Introduction
- Hardware Techniques
  - Selectable Core Voltage
  - Programmable Power Mode of Individual Tiles
- EDA Solutions
  - Dynamic Power Optimization in LUTs with Unused Input(s)
  - Leakage Power Optimization by LUT Output Polarity Selection
  - Leakage Power Optimization by LUT Input Vector Reordering
  - Power-Driven Synthesis, Place & Route
  - Clock Power Reduction by Power-Aware Placement and Clock Shutdown
  - Glitch Power Reduction by Don't Care Assignment
- Hardware + EDA
  - Interconnect Power Reduction by Effective Interconnect Capacitance Optimization

Interconnect Power Consumption

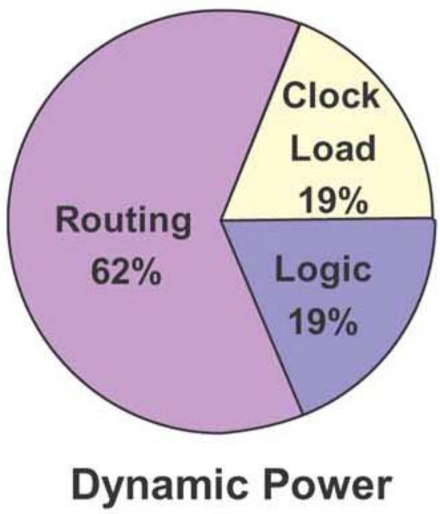
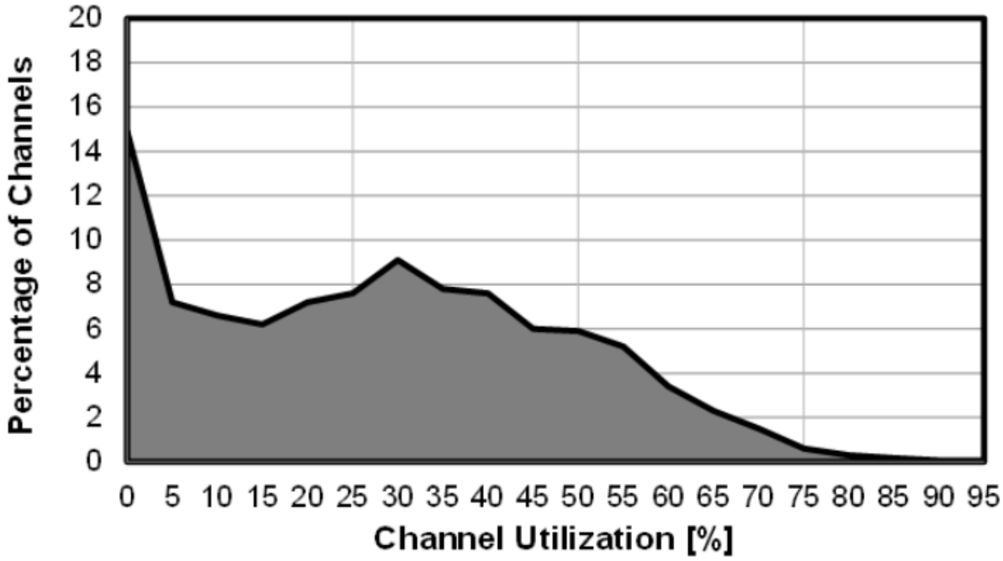


*Figure taken from [Tuan07]

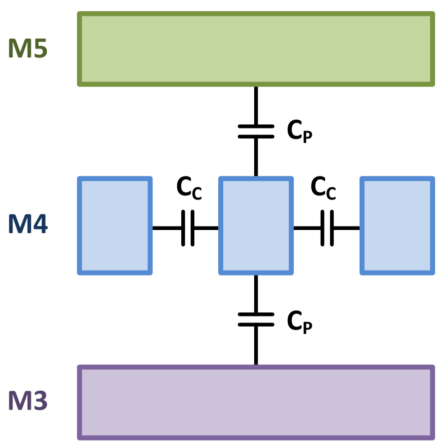
- Routing power is prime component of FPGA dynamic power
- Large wire capacitance results in high power consumption

Unused Wires in FPGA



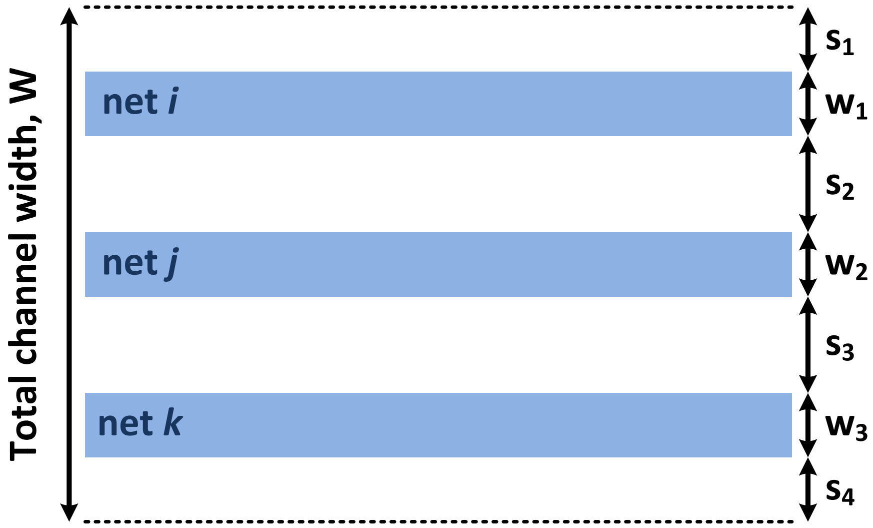
- FPGAs typically have underutilized wires
- Can we take advantage of unused wires?

Wire Capacitance



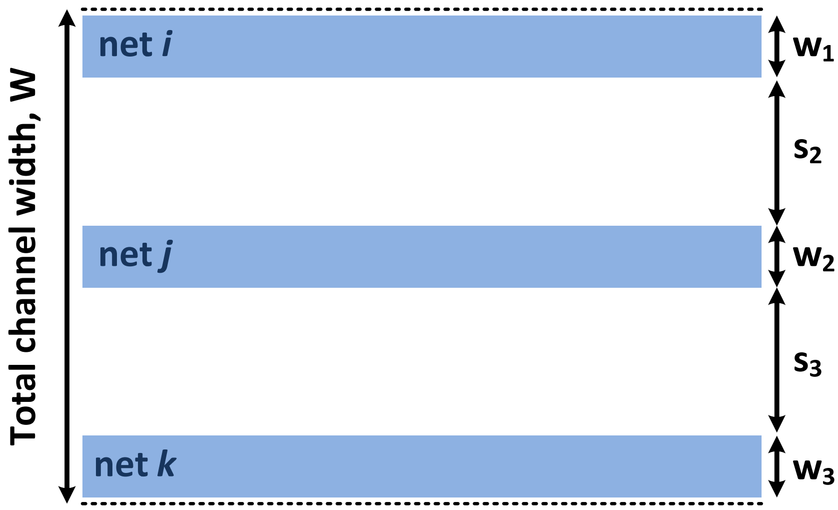
- Wire capacitance consists of:
  - Coupling capacitance (C_C) – between adjacent wires on same layer
  - Plate capacitance (C_P) – between adjacent wires on different layers
- Due to aspect ratio of wires, C_C is dominant

Wire Capacitance Optimization in ASICs



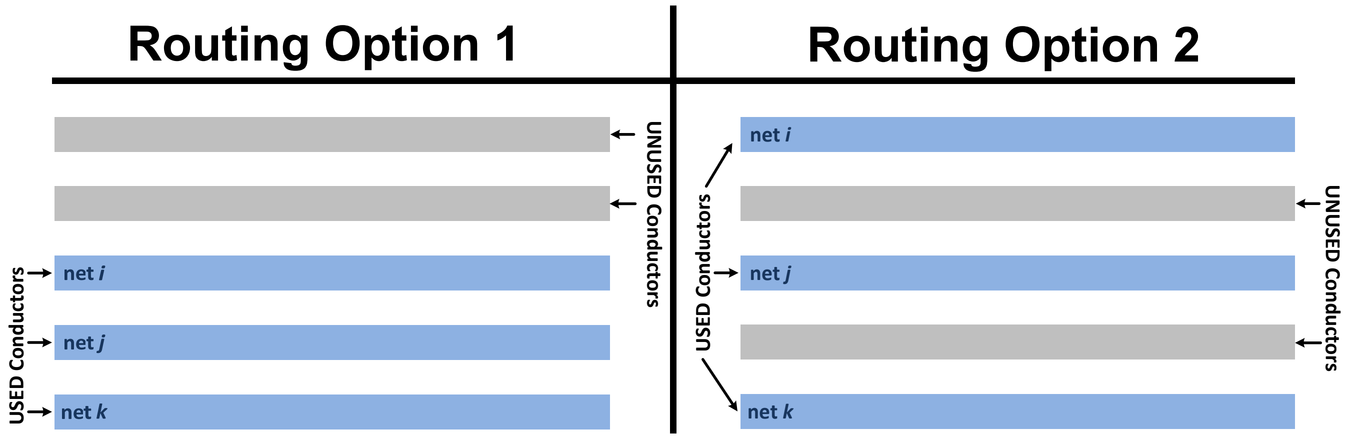
- In ASICs, have freedom to optimize wire width and spacing
  - Can optimize w_i and s_i to maximize timing, minimize power
  - Optimize w_i and s_i subject to $\Sigma w_i + \Sigma s_i = W$

Wire Capacitance Optimization in ASICs



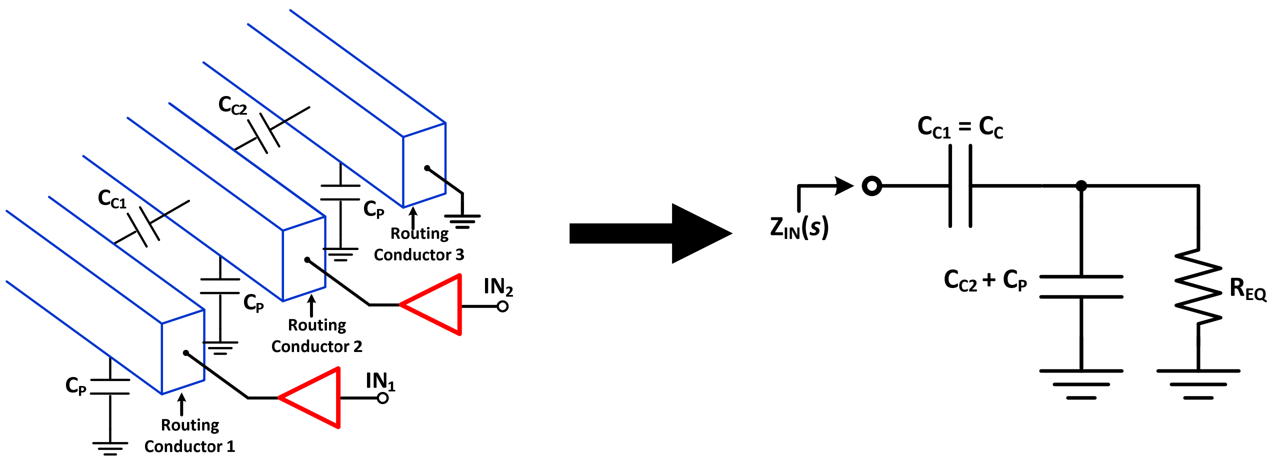
- If net *j* is timing/power critical:
  - Can increase s_2 and s_3 to reduce C_C
  - Reduces capacitance on net *j*, improves speed and reduces power
- Can also optimize w_1, w_2, w_3 for speed and power

In FPGAs?



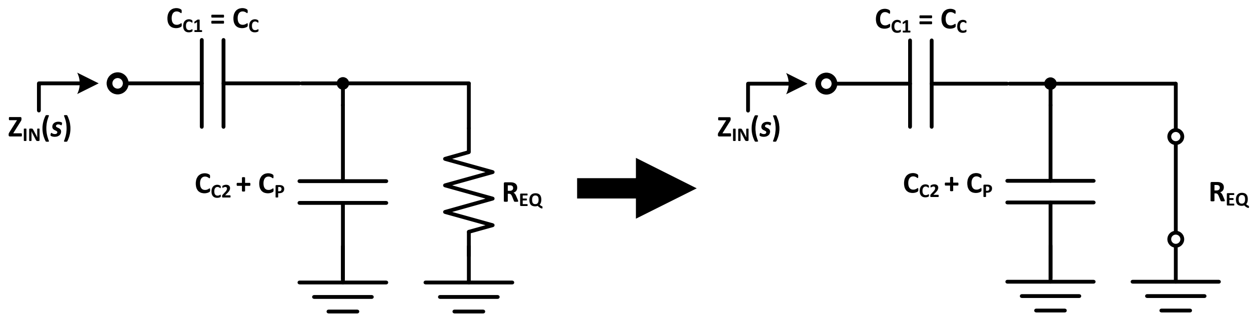
- FPGA wiring prefabricated, width and spacing fixed
- Can't space wires used wires apart, unused wires in the way
- Capacitance on wires in two routing options the same
  - Despite the fact that nets *i,j,k* are now spaced further apart

Wire Cap. Optimization (1)



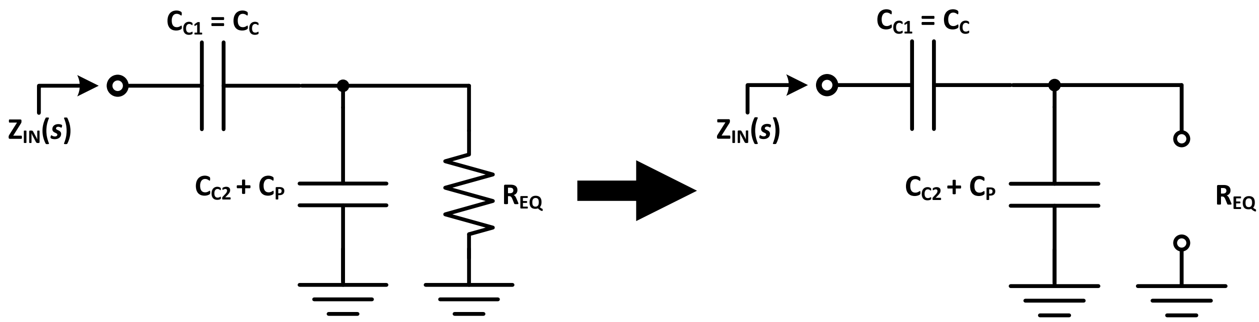
- What's the total impedance seen by Routing Conductor 1, looking towards Routing Conductor 2?

Wire Cap. Optimization (2)



- If R_{eq} is small, capacitor $C_{C2} + C_P$ is shorted out
- Impedance looking towards Routing Conductor 2 is the capacitor C_c

Wire Cap. Optimization (3)

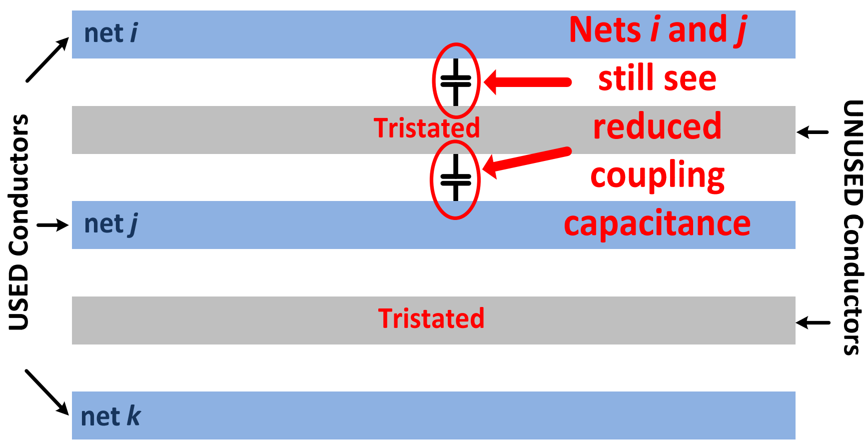


- If R_{eq} is large, we approximate as an open circuit
- Z_{IN} equal to series combination of C_C and C_{C2} + C_P

Wire Cap. Optimization (3)

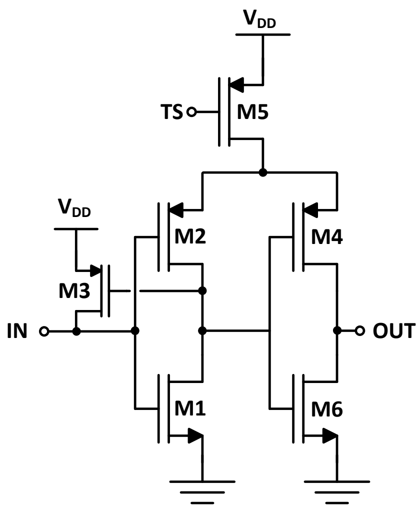
- Series combinations of capacitors result in reduced capacitance:
  - If C_1 in series with C_2, eq. capacitance $C_{eq} = C_1C_2/(C_1 + C_2) < C_1$
- So, we can reduce capacitance if R_{eq} is large enough
- Making R_{eq} large is bad…
  - buffer delay ~ $R_{eq}C_{wire}$ --> increase in R_{eq} increases delay
- What if we made R_{eq} large only for unused conductors?
  - Would not result in increased delay of used conductors
  - Neighbouring used conductors would see benefit of reduced cap.
- Need to be able to set R_{eq} large for unused conductors, but small for used conductors
  - Use tri-state buffers!

Optimize Wire Cap. by TSB and Routing



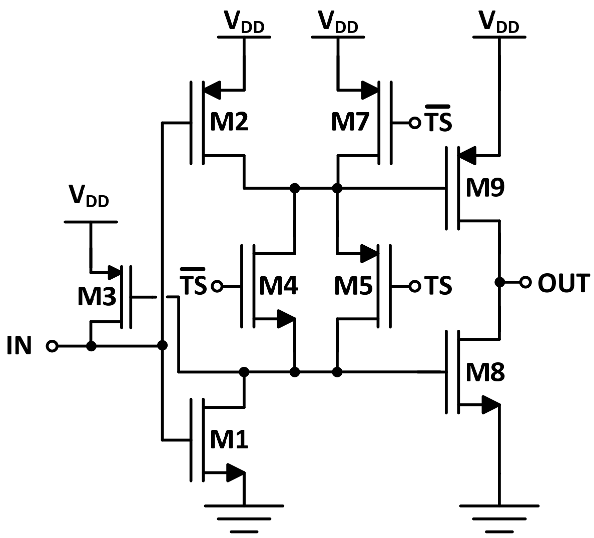
- If intermediate wires are tristated, see reduced C_C !!
- In this work we tristate unused wires to reduce wire cap
  - Proposed a novel, lightweight TSB topology
  - Proposed CAD techniques to space wires out, reduce effective cap.

Traditional Tri-state Buffers



- Header transistor M5 cuts off pull up path to output
- Unused buffer would have IN at VDD
  - M_1 pulls gate of M_6 to GND
- Large area cost: size of M_2, M_4 and M_5 must be doubled to maintain same delay as a conventional buffer

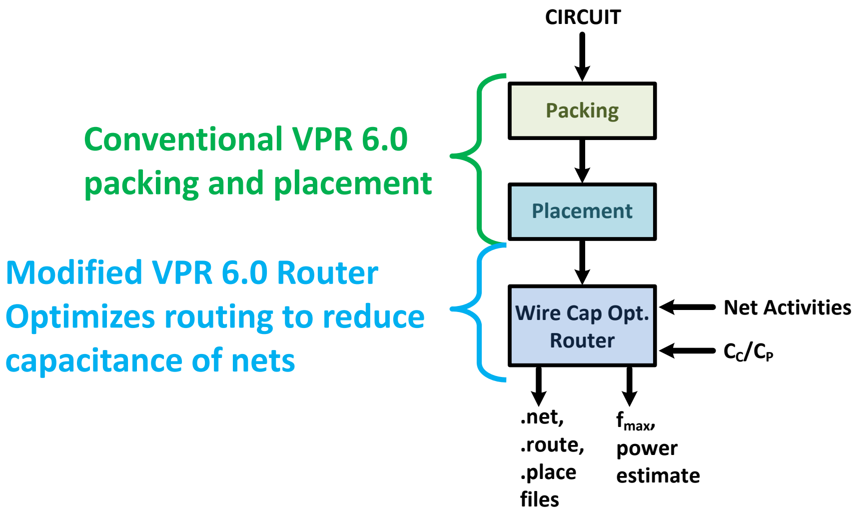
Proposed Tri-state Buffer



N.B. Tri-state mode is achieved without transistor stacking in the output stage

Buffer Topology	Area	TS Mode Leakage Reduction [%]
Conventional	99	45
Proposed	3	25.4

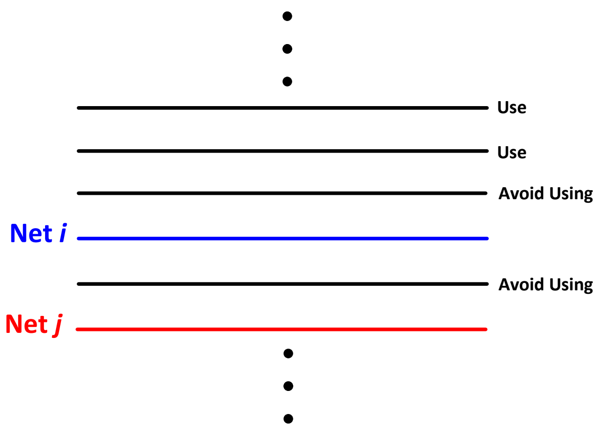
Proposed CAD Flow



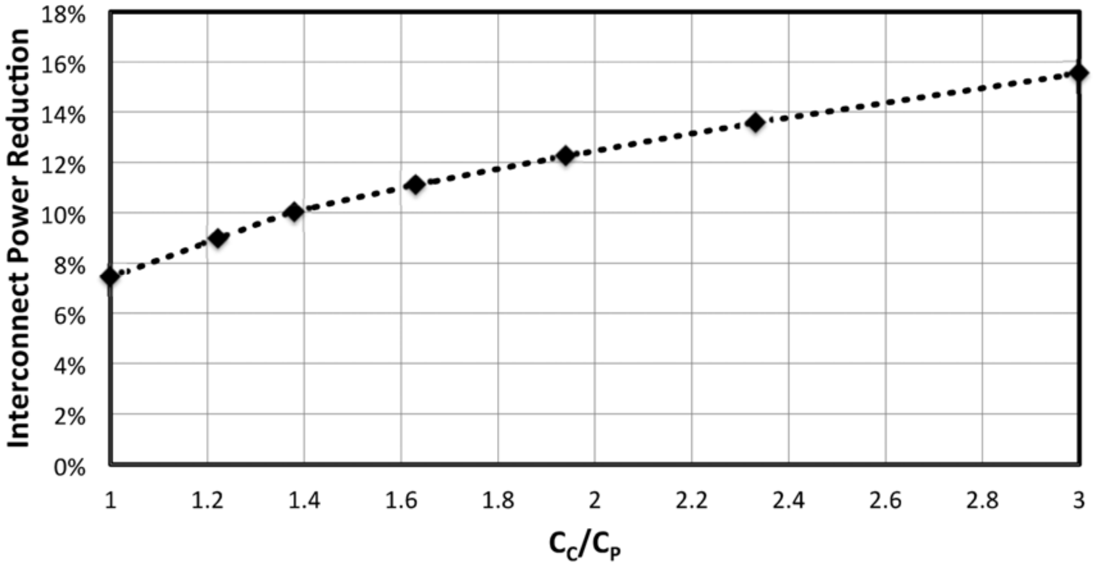
- Power and speed of a conductor can be optimized if adjacent conductor(s) unused
- For capacitance reduction we need CAD which ensures conductors adjacent to power/timing critical nets are unused

Modifications to VPR Router

- VPR router cost function for expanding net *i* to node *n*:
  - Cost(n) = f(congestion(n), criticality(i), delay(n))
  - If *i* is timing critical focus on using fastest resources
  - If *i* is not timing critical use uncongested resources
- To maximize capacitance reduction:
  - Want to route high activity nets with unused adj. conductors
  - Want to avoid using routing conductors adj. to high activity nets



Results



- Dynamic power reduction exceeds 15% for $C_C/C_P \approx 3$
- Get additional 14.6% leakage power savings from TSB
- Critical path degradation ~1%
- Total area overhead ~2.1%

References

- *Stratix-III FPGA Family Data Sheet*, 2008.
- "Active Leakage Power Optimization for FPGAs", in *FPGA'04*
- "Input Vector Reordering for Leakage Power Reduction in FPGAs", *TCAD, Sept. 2008*
- "CAD Techniques for Power Optimization in Virtex-5 FPGAs", in *CICC'07*
- "Clock-aware placement for FPGAs", in *FPL'07*
- "FPGA glitch power analysis and reduction", in *ISLPED'11*
- "Optimizing Effective Interconnect Capacitance for FPGA Power Reduction", in *FPGA'14*