An Integer Programming Placement Approach to FPGA Clock Power Reduction

Alireza Rakhshanfar and Jason Anderson

Dept. of Electrical and Computer Engineering
University of Toronto

IEEE/ACM ASP-DAC
Yokohama, Japan

January 2011

Mobile Devices

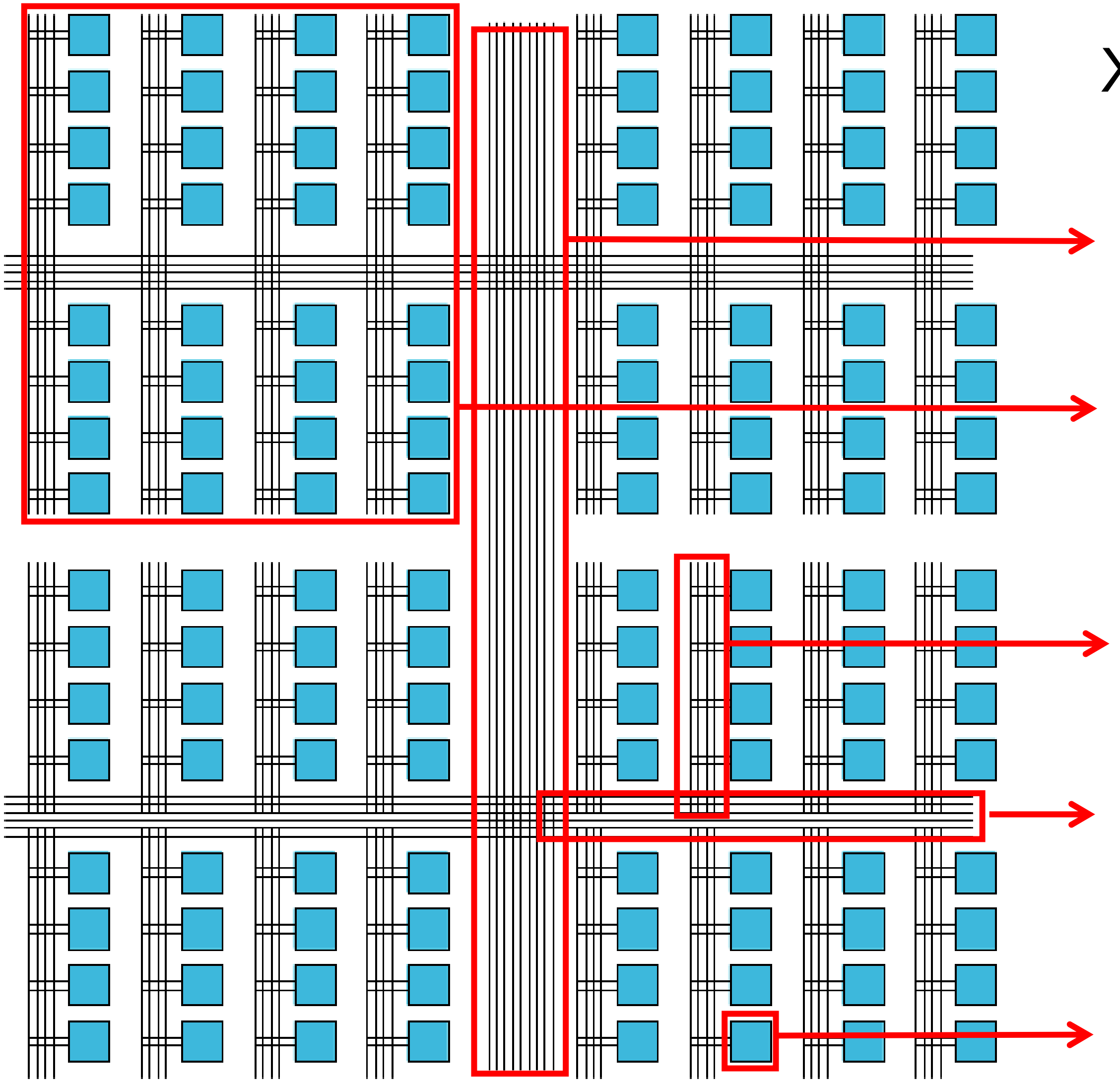
Motivation

- FPGA dynamic power is 7-14x higher than ASICs [Kuon TCAD'07].
  - Long metal wire segments.
  - Overhead of programmability.
- Power consumption excludes them from low-power applications such as mobile devices.
- Significant reductions are needed to close gap with ASICs.

FPGA Power Consumption

- Dynamic power is a function of toggle rates and capacitances of signals.
- Clock power is a major source of power dissipation due to their high fan-out and toggle frequency.
  - Accounts for 20-39% of dynamic power consumption [Degalahal ASP-DAC’05].
- **This work:** CAD technique to reduce FPGA clock power.

FPGA Clock Network



Xilinx Virtex-5 family

Root Spines

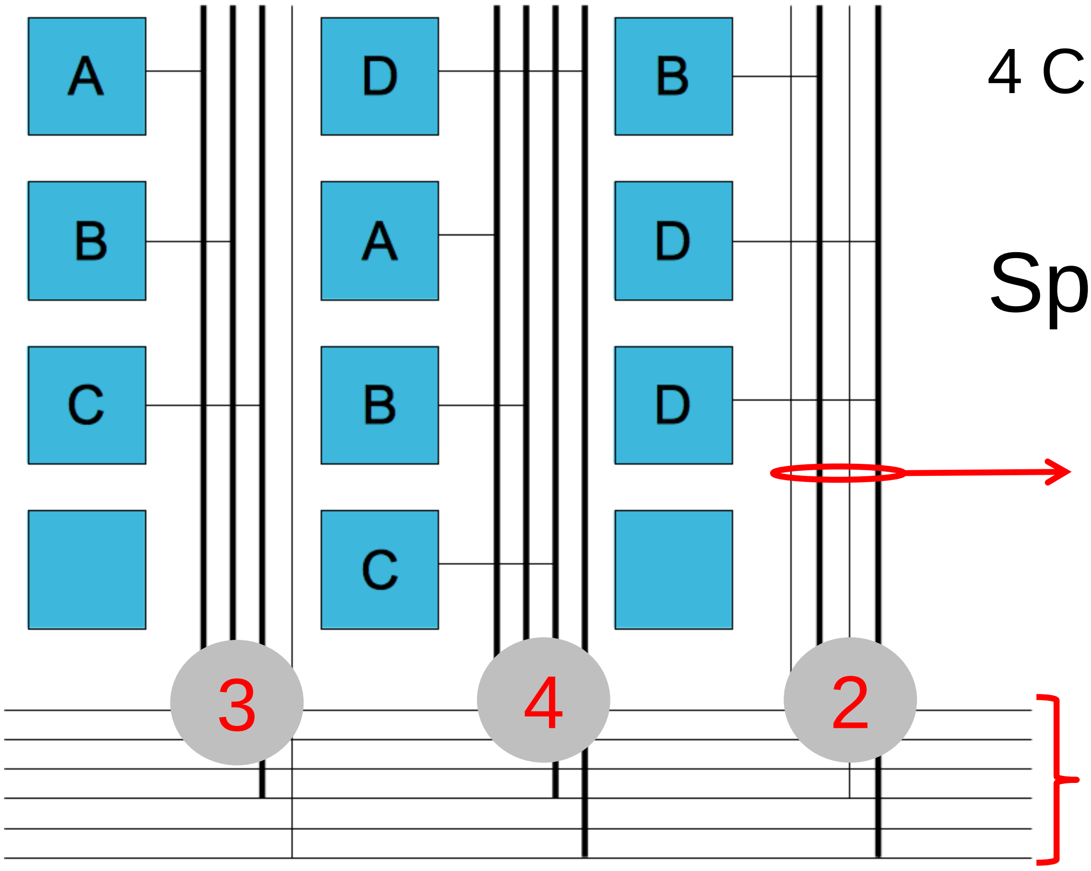
Clock Region

Vertical Half-Spines

Horizontal Spines

Logic Block

Spine Reduction



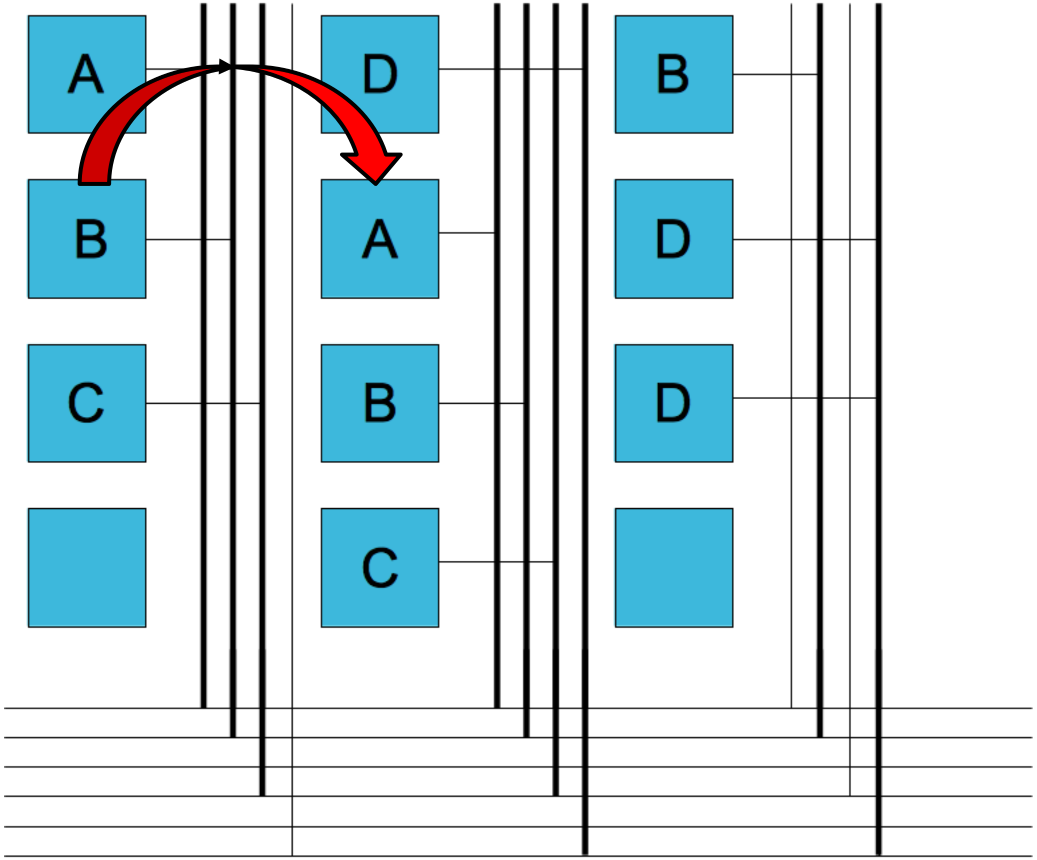
4 Clocks: A,B,C and D

Spine Count: 9

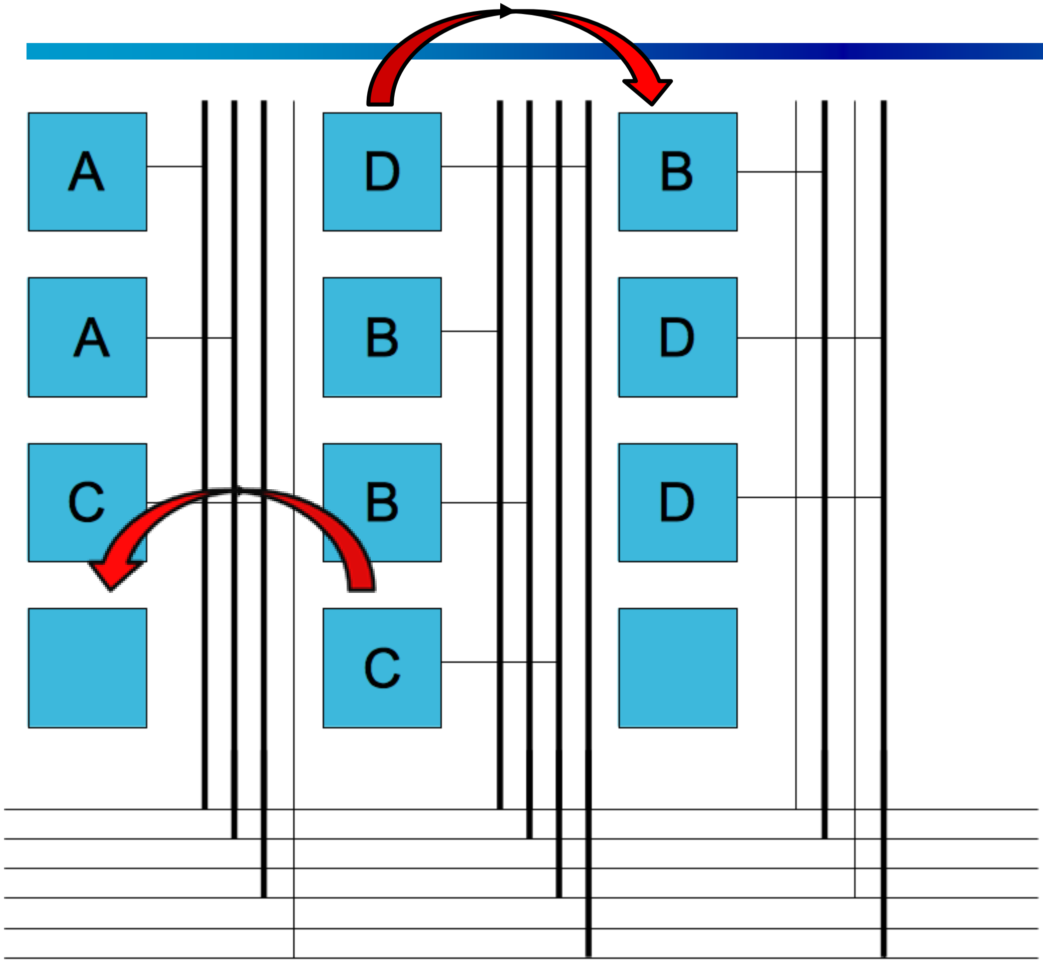
Vertical Half-Spines

Horizontal Spines

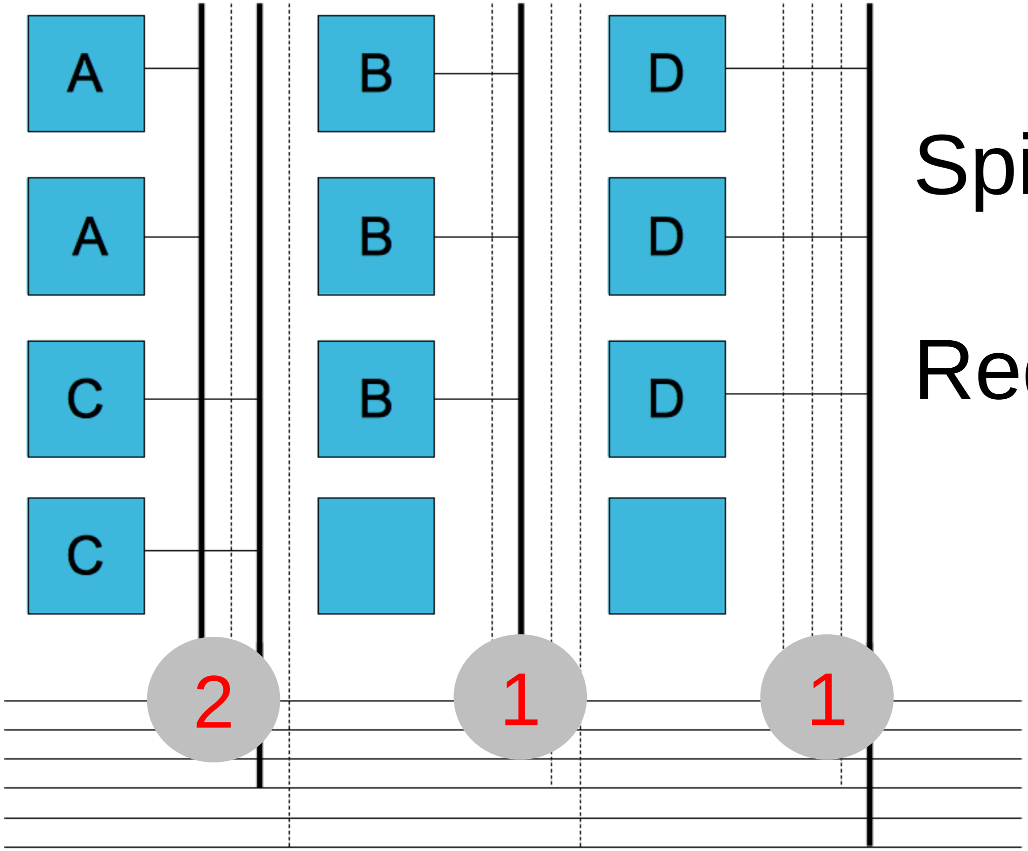
Spine Reduction



Spine Reduction



Spine Reduction



Spine Count: 4

Reduction:56%

Related Work

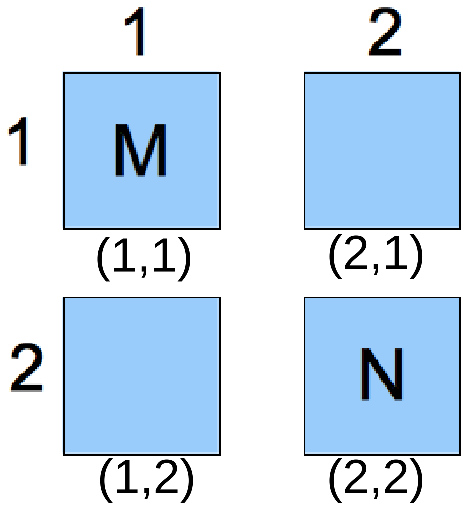
- Prior work at Actel Corp. [Hsu IET-CDT'10]

  has used this technique in annealing-based placement framework.
  - Annealing works with *any* cost function.
- However, many placement algorithms use analytical techniques.
- Our technique is a post-placement task:
  - Can be incorporated in **any placement algorithm**.

Integer Linear Programming

- Formal optimization technique with broad applications.
- Linear objective function.
- Linear equality and inequality constraints.
- NP-hard to solve.
- ILP Solver: lp_solve.

Our Approach

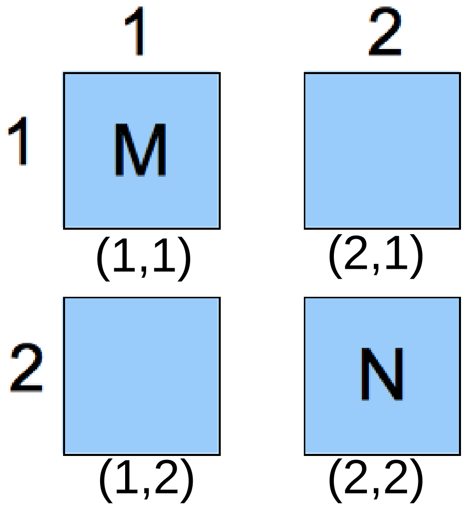


Block M: Clock A
Block N: Clock B

Binary Variables: clkA_1, clkA_2
clkB_1, clkB_2

clkA_1 is true if clock A is used in the first column

Our Approach



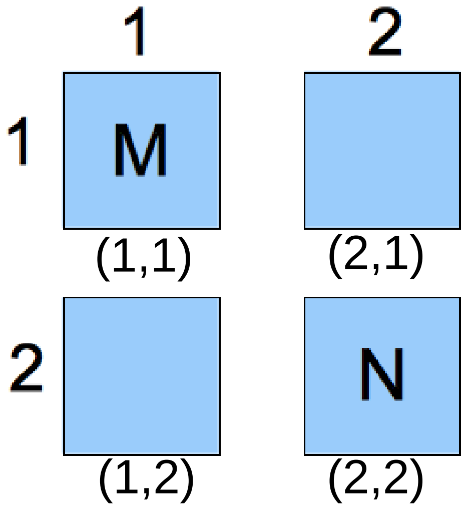
Block M: Clock A
Block N: Clock B

Binary Variables: clkA_1, clkA_2
clkB_1, clkB_2

blkM_11, blkM_12, blkM_21, blkM_22
blkN_11, blkN_12, blkN_21, blkN_22

blkM_11 is true if block M is placed in location 1,1

Our Approach



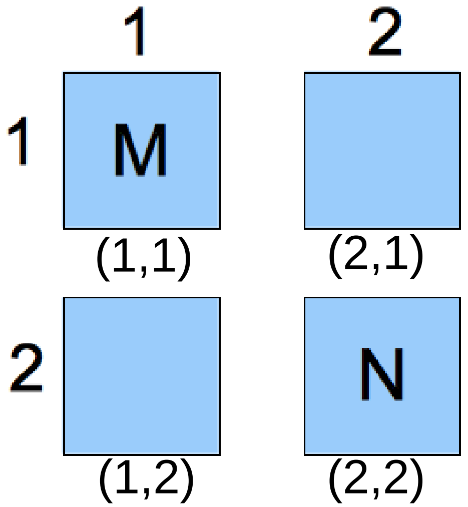
Block M: Clock A
Block N: Clock B

Binary Variables: clkA_1,clkA_2,clkB_1,clkB_2
blkM_11, blkM_12, blkM_21, blkM_22
blkN_11, blkN_12, blkN_21, blkN_22

Objective Function:

min clkA_1 + clkA_2 + clkB_1 + clkB_2

Our Approach



Block M: Clock A

Block N: Clock B

Binary Variables: clkA_1,clkA_2,clkB_1,clkB_2

blkM_11, blkM_12, blkM_21, blkM_22

blkN_11, blkN_12, blkN_21, blkN_22

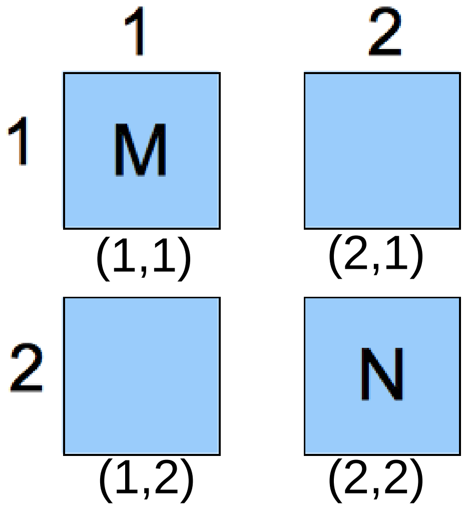
Objective Function: min clkA_1+clkA_2+clkB_1+clkB_2

Placement:

blkM_11+blkM_12+blkM_21+blkM_22=1

blkN_11+blkN_11+blkN_21+blkN_22=1

Our Approach



Block M: Clock A
Block N: Clock B

Binary Variables: clkA_1,clkA_2,clkB_1,clkB_2
blkM_11, blkM_12, blkM_21, blkM_22
blkN_11, blkN_12, blkN_21, blkN_22

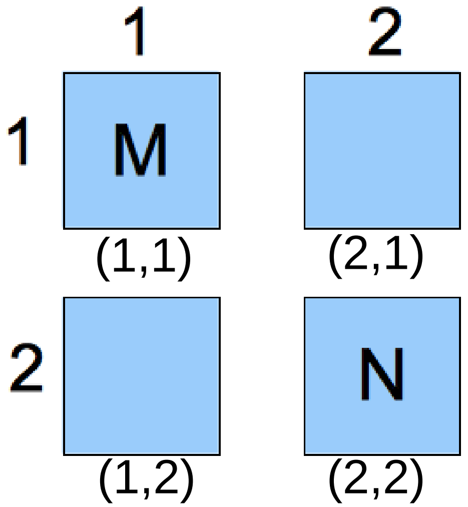
Objective Function: min clkA_1+clkA_2+clkB_1+clkB_2

Placement: blkM_11+blkM_12+blkM_21+blkM_21=1
blkN_11+blkN_11+blkN_21+blkN_21=1

Exclusivity: blkM_11+blkN_11≤1

blkM_12+blkN_12≤1
blkM_21+blkN_21≤1
blkM_22+blkN_22≤1

Our Approach



Block M: Clock A
Block N: Clock B

T: large constant

Binary Variables: clkA_1,clkA_2,clkB_1,clkB_2
blkM_11, blkM_12, blkM_21, blkM_22
blkN_11, blkN_12, blkN_21, blkN_22

Objective Function: min clkA_1+clkA_2+clkB_1+clkB_2

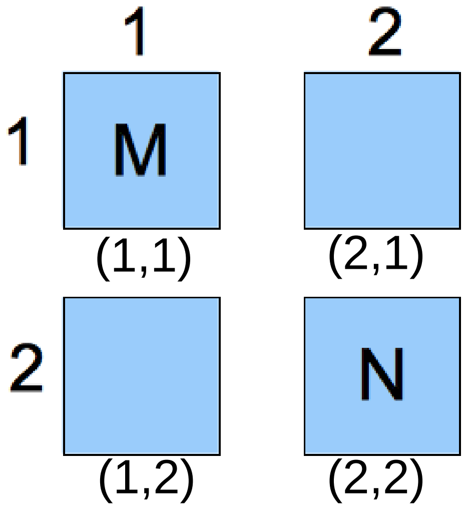
Placement: blkM_11+blkM_12+blkM_21+blkM_21=1
blkN_11+blkN_11+blkN_21+blkN_21=1

Exclusivity: blkM_11+blkN_11≤1
blkM_12+blkN_12≤1
blkM_21+blkN_21≤1
blk_M22+blkN_22≤1

Spine Count: blkM_11+blkM_12 ≤ T*clkA_1

blkM_21+blkM_22 ≤ T*clkA_2
blkN_11+blkN_12 ≤ T*clkB_1
blkN_21+blkN_22 ≤ T*clkB_2

Our Approach



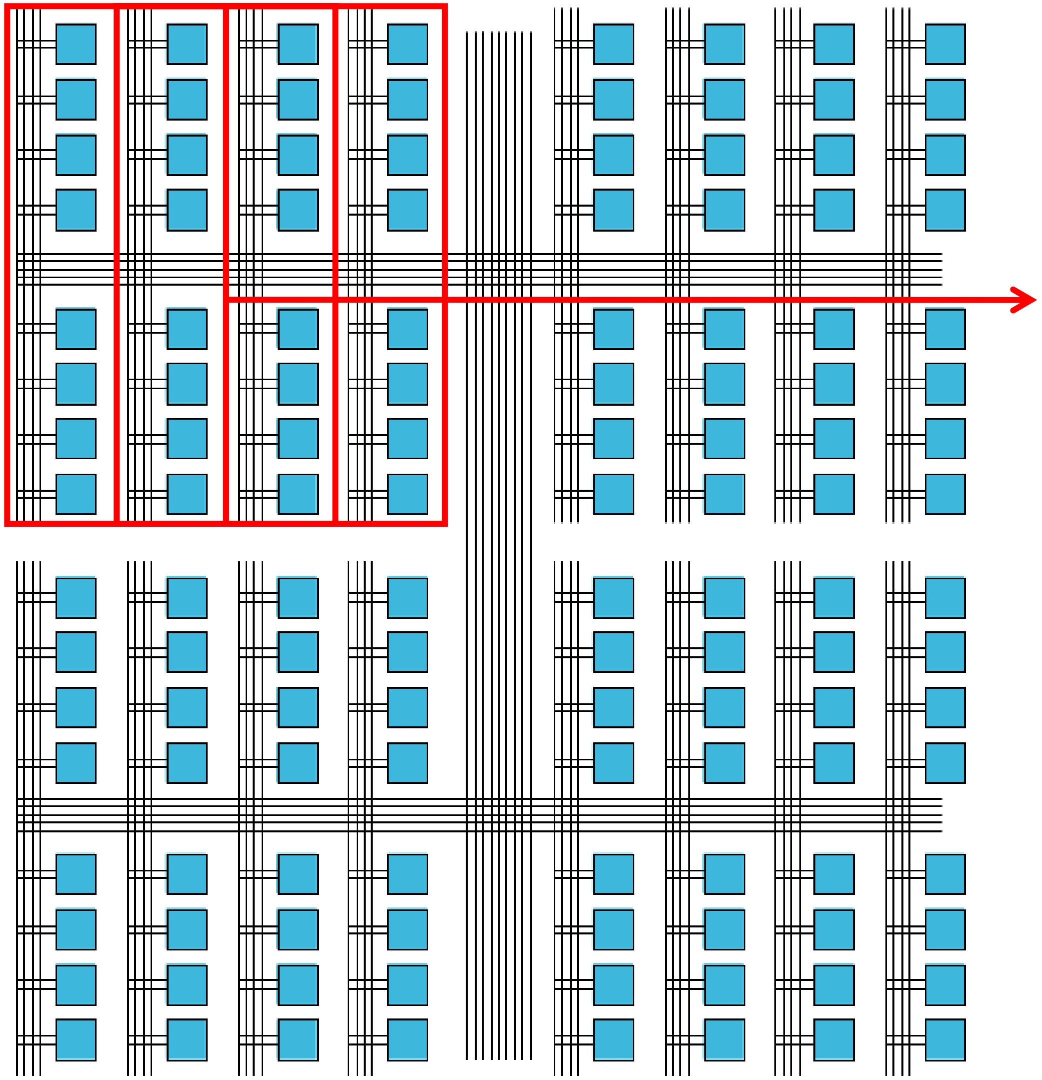
Block M: Clock A
Block N: Clock B

- Multiple solutions to an ILP problem.
- Original placement is optimized for wirelength and timing
- Anchoring term: encourages blocks to remain in their original position.

Original Objective Function: min clkA_1+clkA_2+clkB_1+clkB_2

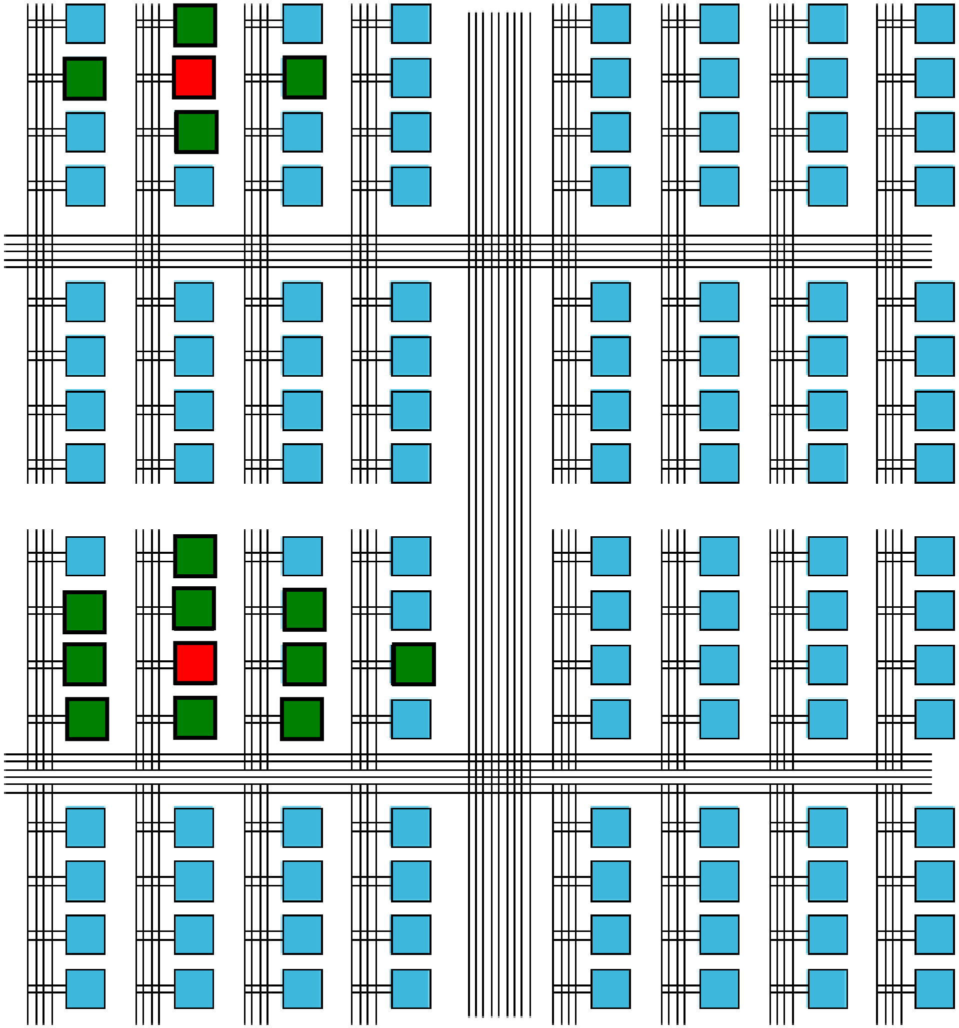
Augmented Objective Function: min clkA_1+clkA_2+clkB_1+clkB_2
+ 0.1*(blkM_12+blkM_21+blkM_22
+blkN_11+blkN_12+blkN_21)

Managing Runtime



Optimization window

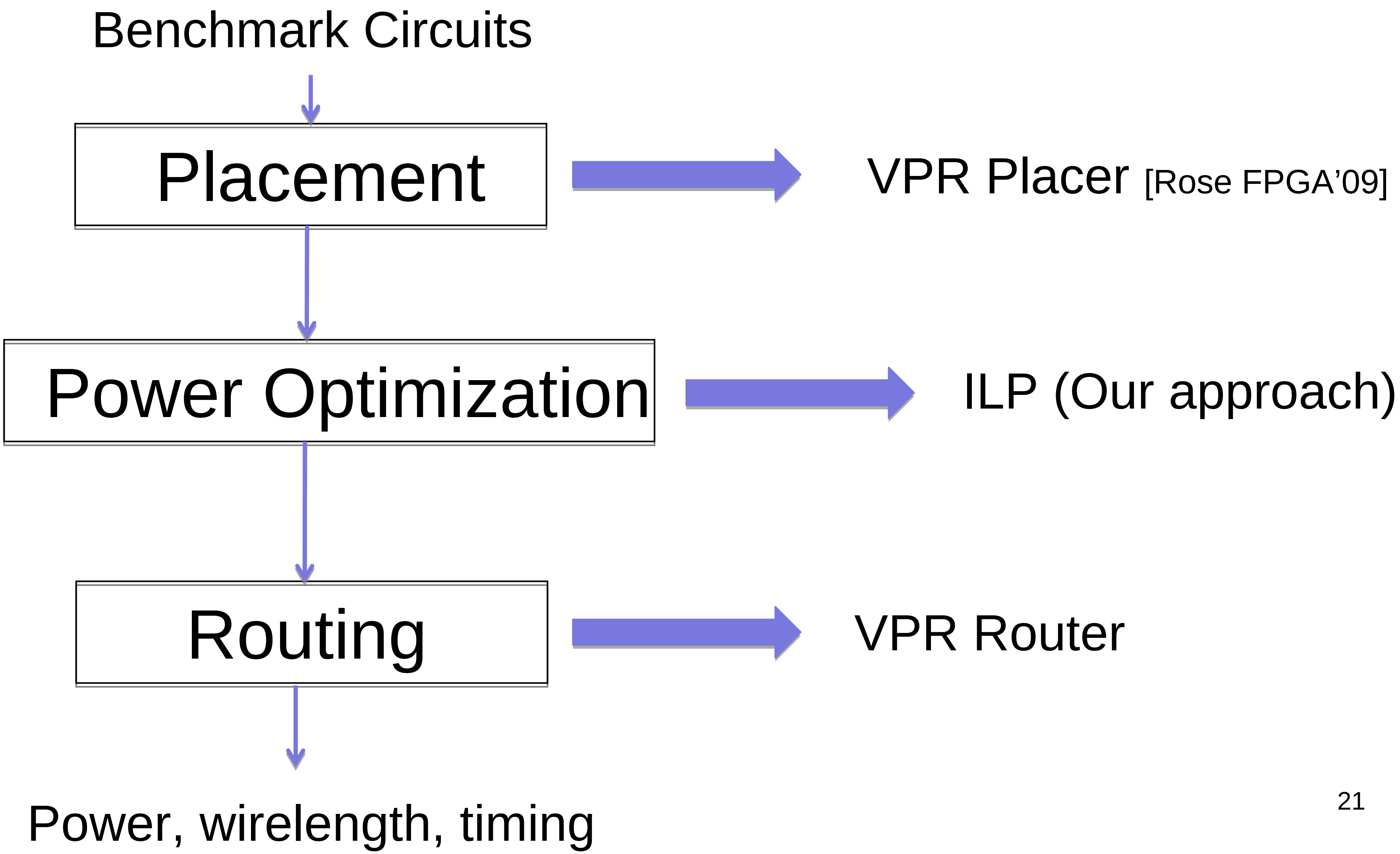
Permissible Move Distance



Manhattan distance of 1

Manhattan distance of 2

Methodology



Benchmark Circuits

Placement → VPR Placer [Rose FPGA'09]

Power Optimization → ILP (Our approach)

Routing → VPR Router

Power, wirelength, timing

Measurement Metrics

- Clock signals are routed from horizontal spines to vertical half-spines on an as-needed basis.
- Clock signals usually span many columns and require many vertical half-spines.
- Clock network capacitance mainly determined by the # of vertical half-spines used.
- **Our metrics:** number of vertical half-spines (spine count), wirelength, timing

Comparative Baseline

- Aims to reduce spine count in an annealing-based placer (Actel Corp.) [Hsu IET-CDT'10].
- Cost function:

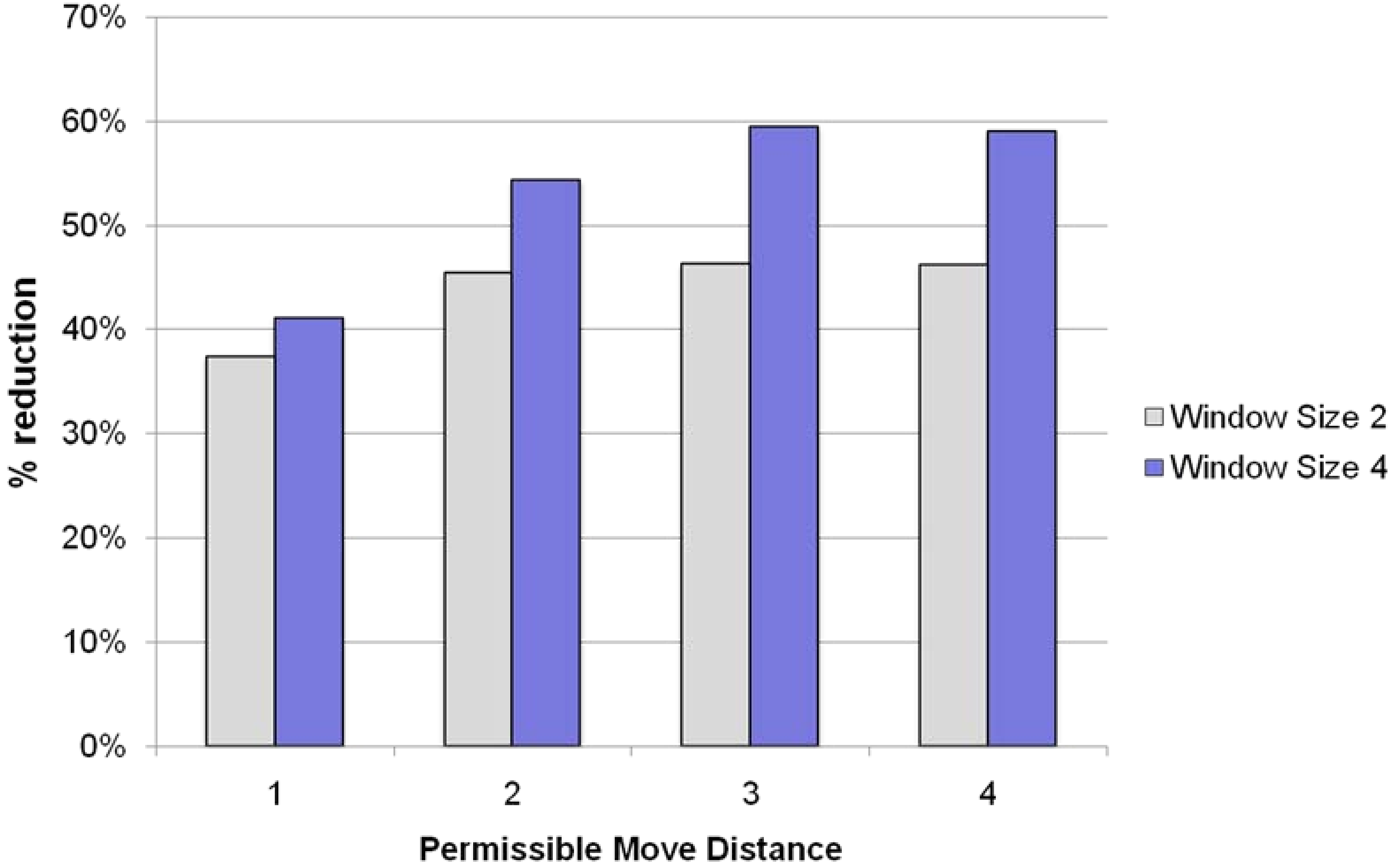
  Cost= a•WireLength + b•Timing + c•ClockPowerCost

- ClockPowerCost = sum of spine costs
- Spine cost:
  - when no logic blocks use a spine, spine cost=0.
  - when blocks are initially added to a spine, spine cost ramps quickly.
  - when spine is ½ full, spine cost increases slowly up to a maximum when the spine is full.

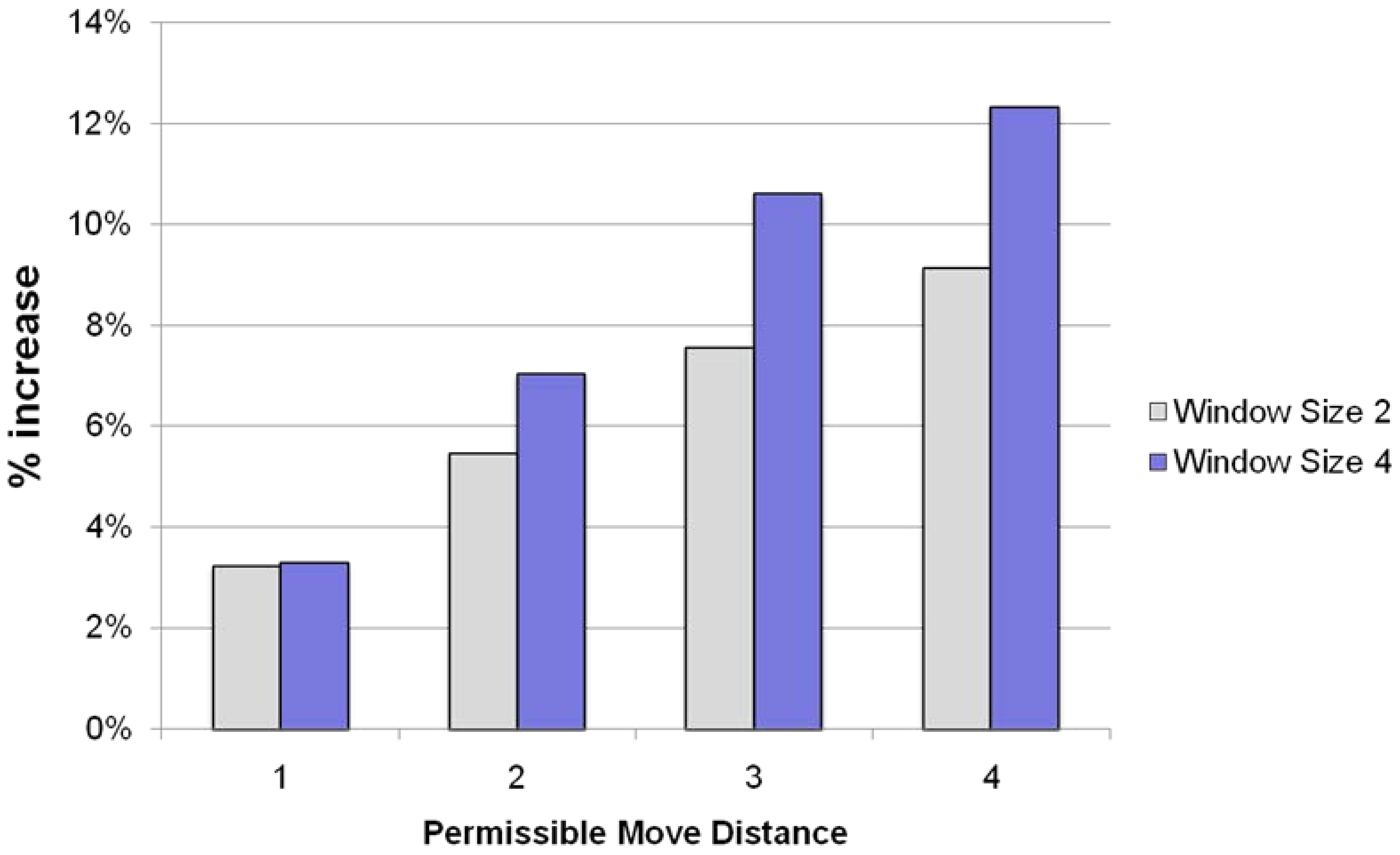
Experimental Methodology

- FPGA architecture: 10 4-LUT/flip-flop pairs per logic block and length-4 wire segments.
- Altered 20 combinational and sequential benchmark circuits to contain multiple clock signals.
- For each circuit, the logic blocks were arbitrarily assigned one of the four clock domains.

Spine Count Reduction

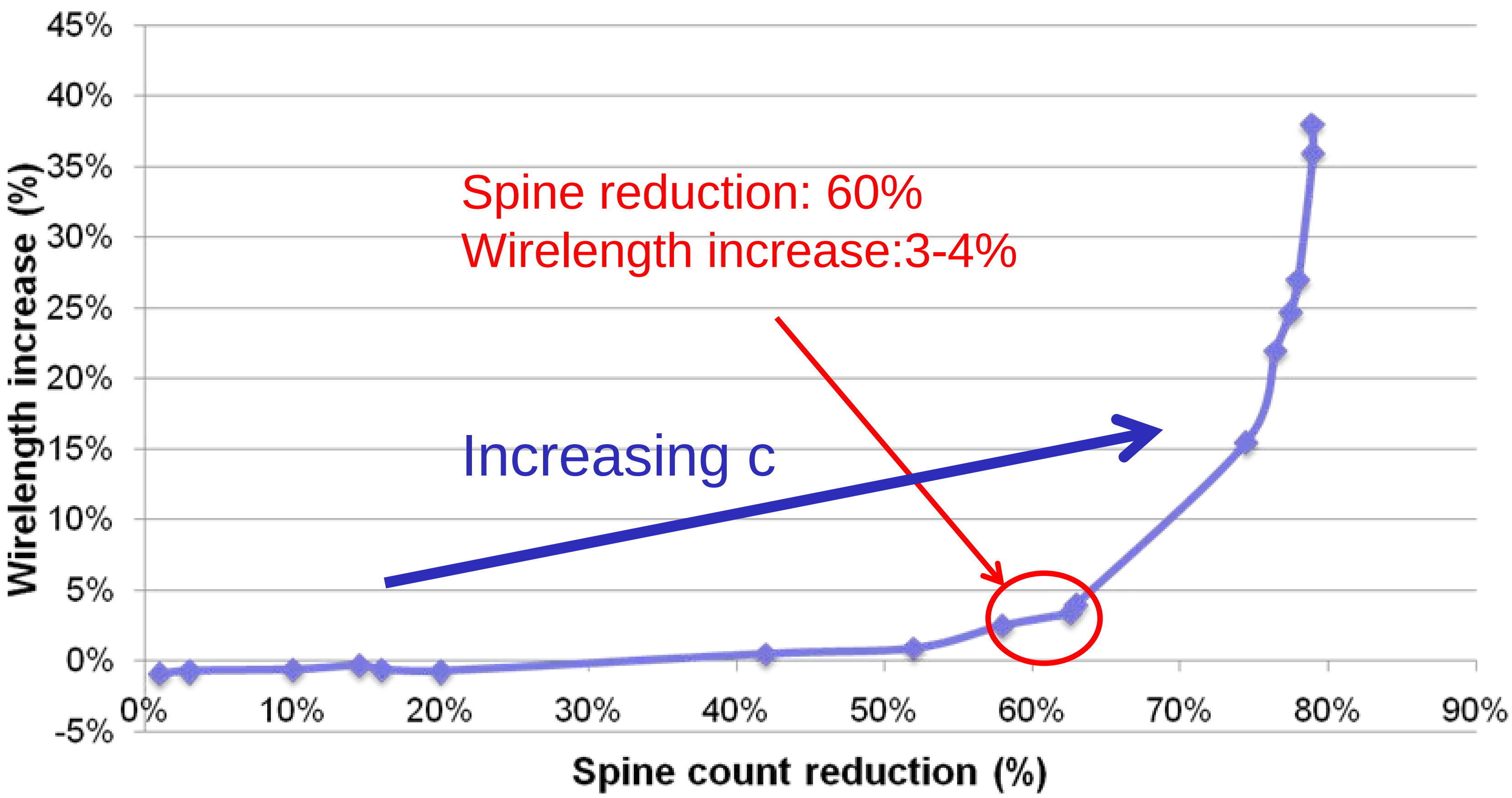


Post Routing Wirelength Increase



Annealing Based Approach

Cost= a•WireLength + b•Timing + c•ClockPowerCost



Spine reduction: 60%
Wirelength increase:3-4%

Comparison

Optimization Approach	**Spine Reduction**	**Wirelength Increase**
Our approach-Permissible move distance 2	52-54%	6-7%
Our approach-Permissible move distance 3	58-61%	10-11%
Annealing-based [Hsu IET-CDT’10]	60%	3-4%

Projected Power Benefit

	% Total Dynamic Power	**% Reduction**	**Projected power reduction**
Clock power	25%	50%	12.5%
Logic-signal power	50%	-6%	-3%
Overall			~9%

Conclusions

- Post-placement ILP based clock power optimization technique.
- Can be used in conjunction with placement algorithm.
- Over 50% reduction of clock spine resources with minimal damage to traditional placement metrics.

Future Work

- Evaluate proposed ILP-based technique within comprehensive power-aware FPGA CAD system that optimizes power throughout flow, including synthesis, packing, place and route.

Questions

	Baseline VPR		VPR+ clock power optimization	
Benchmark	Clk Spine Count	# Wire Segs	Clk Spine Count	# Wire Segs
alu4	73	8148	34	8664
apex2	92	13021	39	13758
apex4	74	9740	32	10410
bigkey	66	13870	32	14306
clma	217	64527	109	68509
des	78	17913	42	18927
diffeq	65	5578	31	6340
dsip	67	13056	30	13733
elliptic	113	18857	46	20433
ex1010	154	36901	71	39430
ex5-p	60	9404	27	10185
frisc	118	22549	48	24023
misex3	73	8537	34	8848
pdc	164	42780	70	44606
s298	97	7716	45	8123
s38417	93	26605	88	29296
s38584.1	119	30842	92	33685
seq	86	11830	40	12884
spla	147	28673	65	30117
tseng	49	4981	22	5599
Geomean	99.0	15563.9	45.1	16660.7

Window size 4, move distance 2