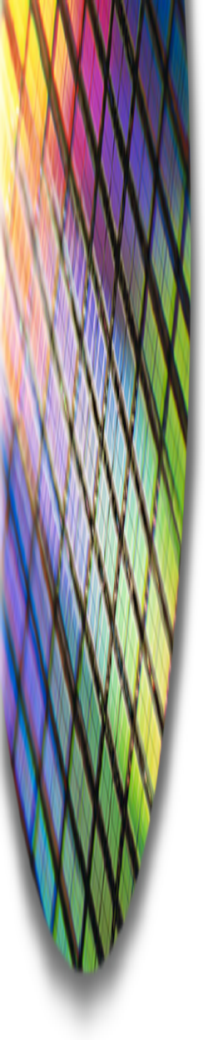


Mentor Graphics

ISPD 2014

UT DA

Clock Tree Resynthesis for Multi-corner Multi-mode Timing Closure

Subhendu Roy[1], Pavlos M. Mattheakis[2], Laurent Masse-Navette[2] and David Z. Pan[1]

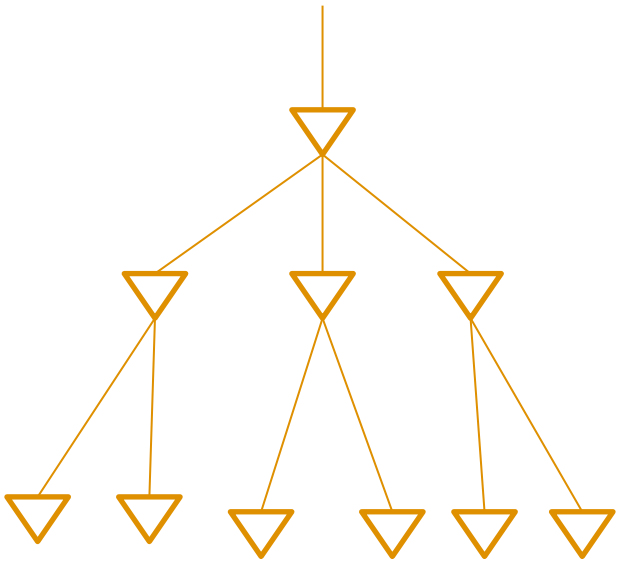
[1]ECE Department, The University of Texas at Austin

[2]Mentor Graphics, Fremont

Outline

- CTS Preliminaries
- Prior Work and Limitations
- Clock Tree Resynthesis
- Experimental Results
- Conclusion and Future Work

CTS-Preliminaries



- CTS – a fundamental step in physical design
- Modern designs – multi-corner, multi-mode (MCMM)
- Timing closure – extremely difficult in MCMM designs

CTS-Preliminaries

- If targeting global zero skew, that would
  - cost in area/power
  - limit achievable operating frequency
- Data-path optimization is not sufficient to handle timing violations
- Need for data path aware clock scheduling or useful clock skew optimization

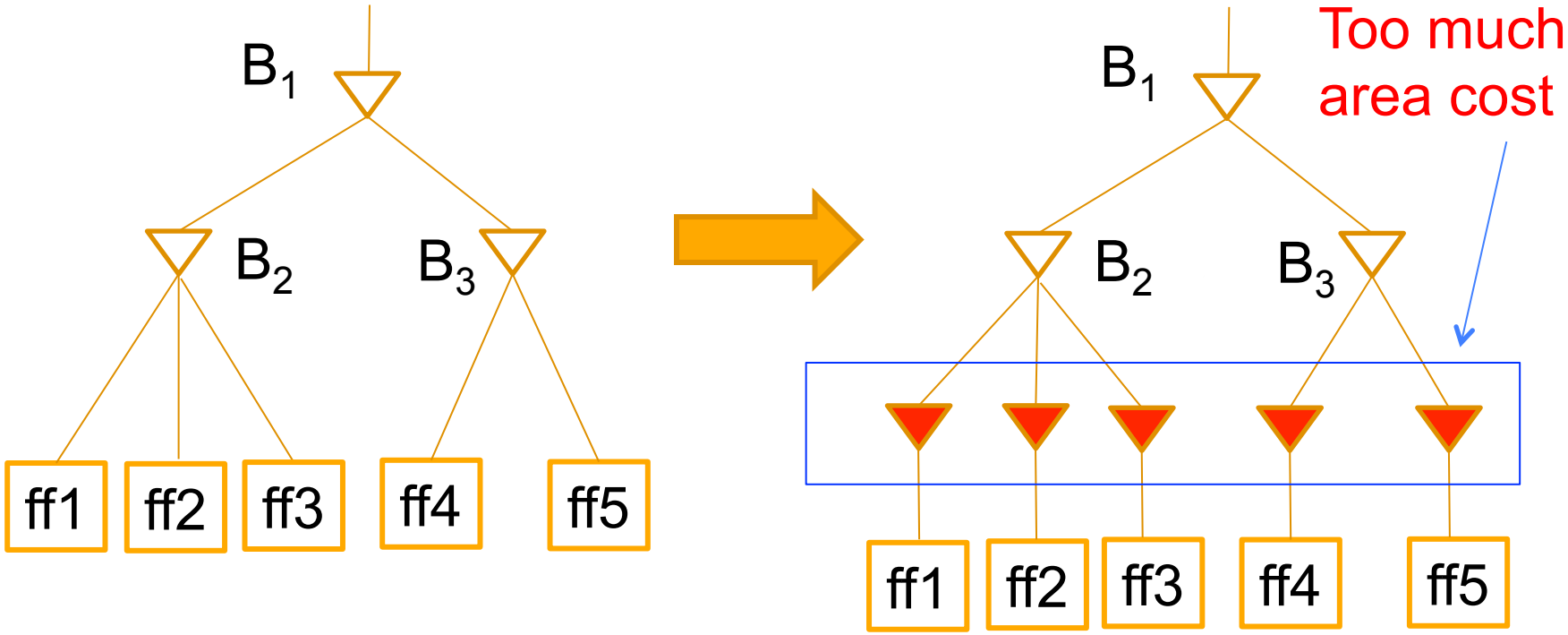
Prior Work and Limitations(1)

Useful Skew Optimization

- [Kourtav+, ICCAD'99], [Nawale+, ICCAD'06] –
  - Solve LP or Quadratic problem
  - Calculate clock skew in pre-CTS stage
  - Actual implementation difficult to achieve in later design stage
  - No support for MCMM

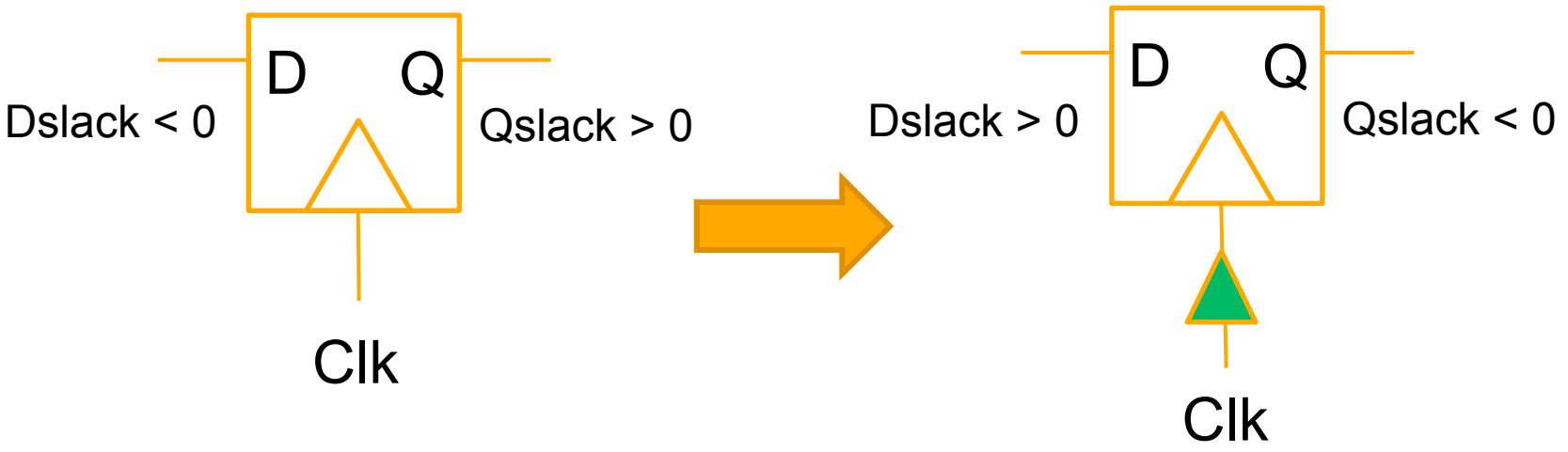
Prior Work and Limitations(2)

- [Lu+, IMSCS'09] – Post-CTS bounded delay buffering at leaves
  - Buffering at leaves ⟶ high area/power cost
  - Does not tackle MCMM scenario



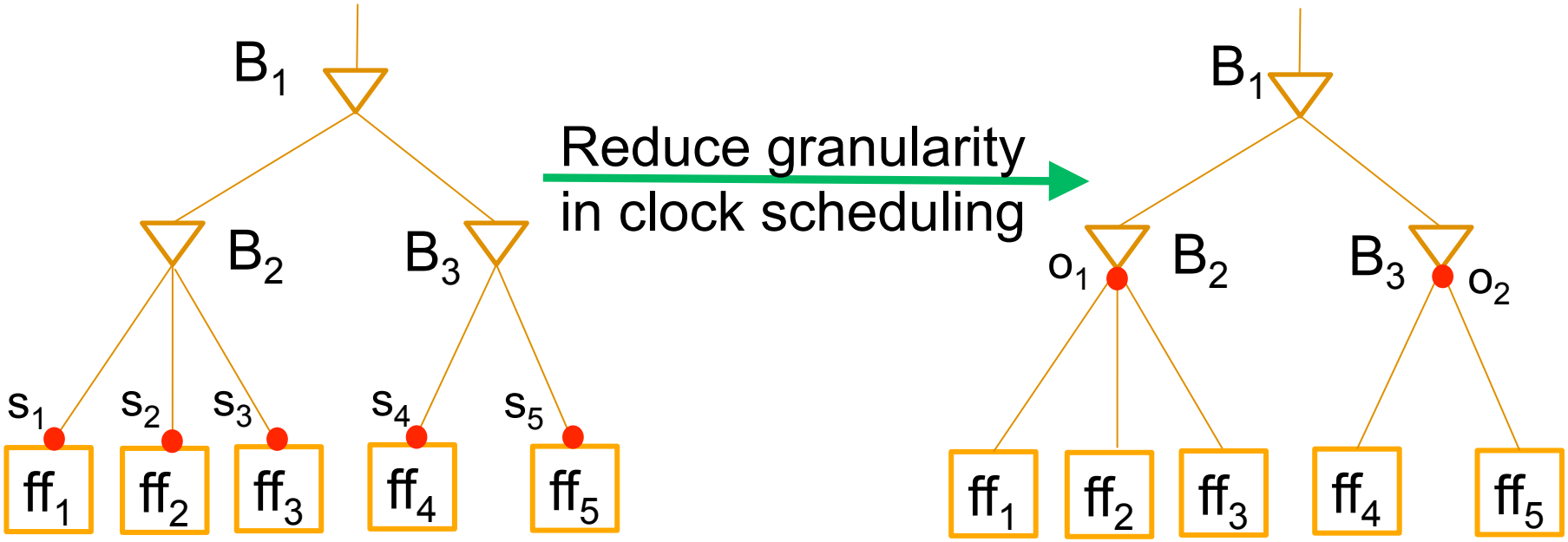
Prior Work and Limitations(3)

- [Shen+, ISQED'10] – Post-CTS useful skew implementation in MCMM
  - Local transformation at leaf-level ⟶ greedy, high area/power cost
  - Insert/remove buffer to delay/speed up clock arrival at flop inputs
  - Speed up by buffer removal may not be practically realizable



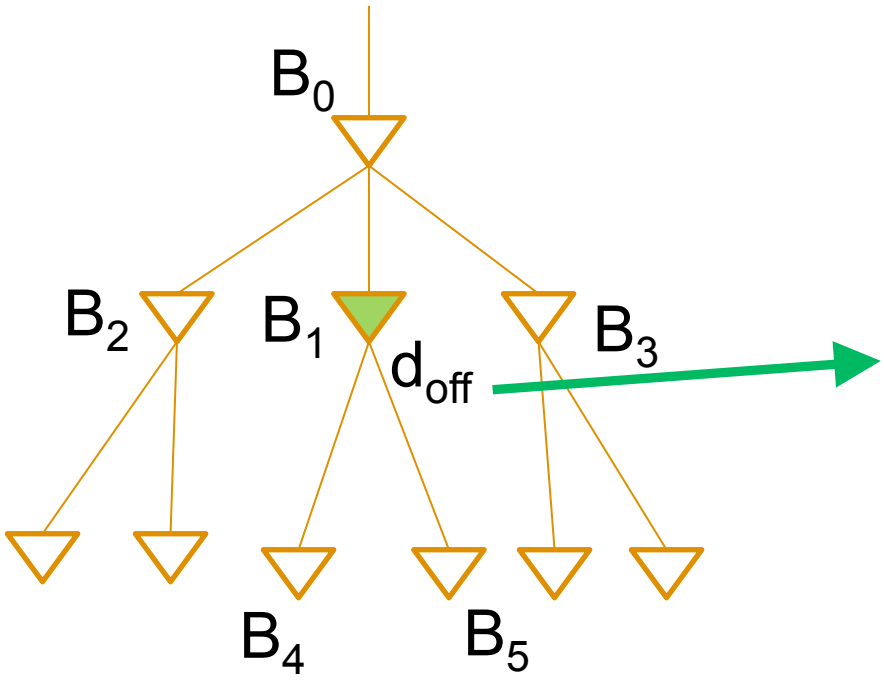
Notion of Offset

- Pre-CTS useful skew ⟶ Difficult to implement
- Post-CTS useful skew ⟶ greedy, high area cost, may not support MCMM



Clock scheduling moved up to driver pins of clock-tree buffers

Notion of Offset



- Positive offset if $d_{off} > 0$, clock-arrival at B_1's output to be delayed by d_{off}
- Negative offset if $d_{off} < 0$, clock-arrival at B_1's output to be expedited by d_{off}

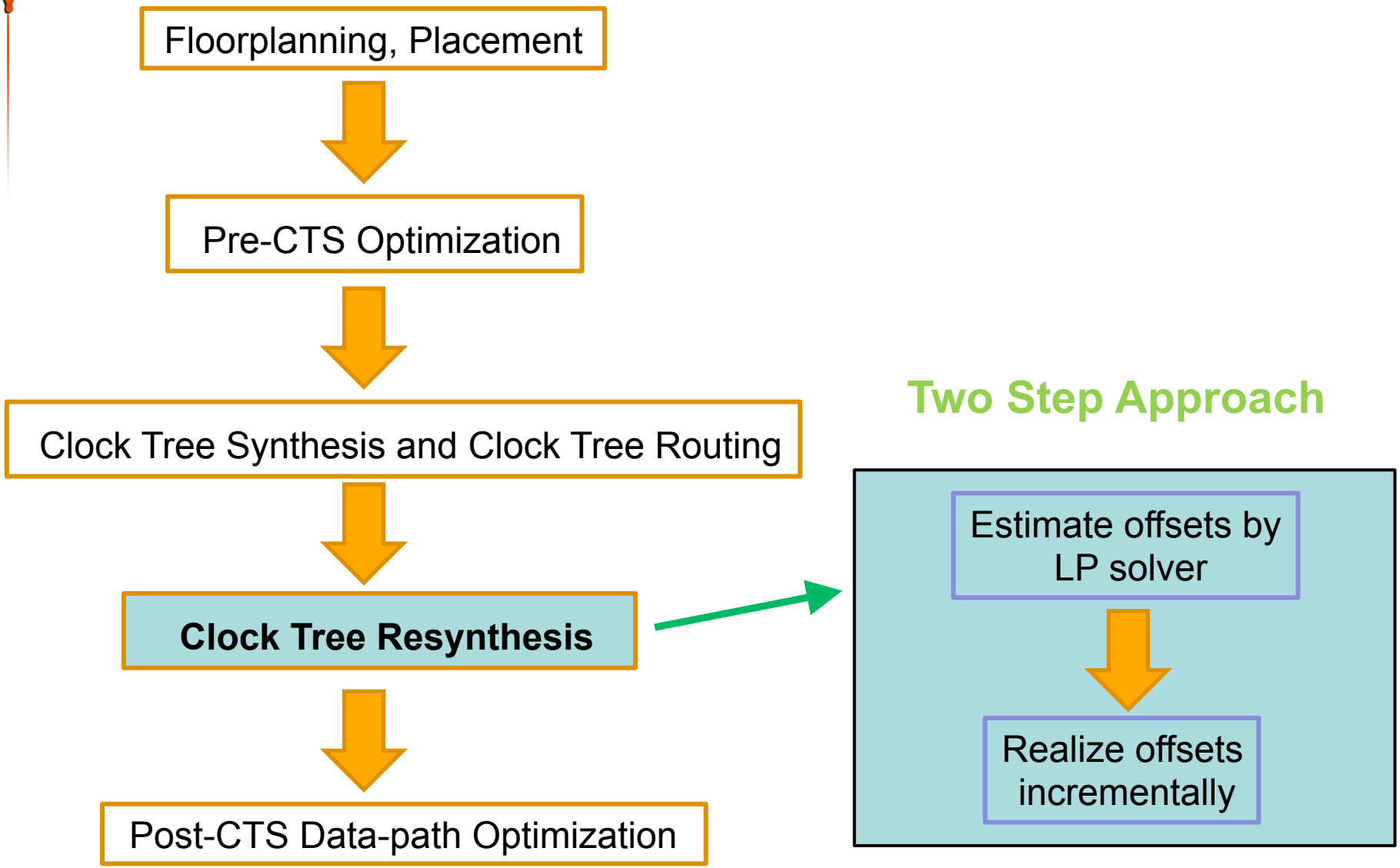
Our Contributions

- First work to consider offsets at output pins of clock tree cells
  - In a placed design with already routed clock tree
- An area-efficient and non-intrusive algorithm is presented
  - To realize negative offsets
- A methodology for clock tree resynthesis presented
  - Significantly improved timing metrics in large-scale industrial designs under MCMM scenarios

Outline

- CTS Preliminaries
- Prior Work and Limitations
- Clock Tree Resynthesis
- Experimental Results
- Conclusion and Future Work

How CT-Resynthesis Fit in the Flow



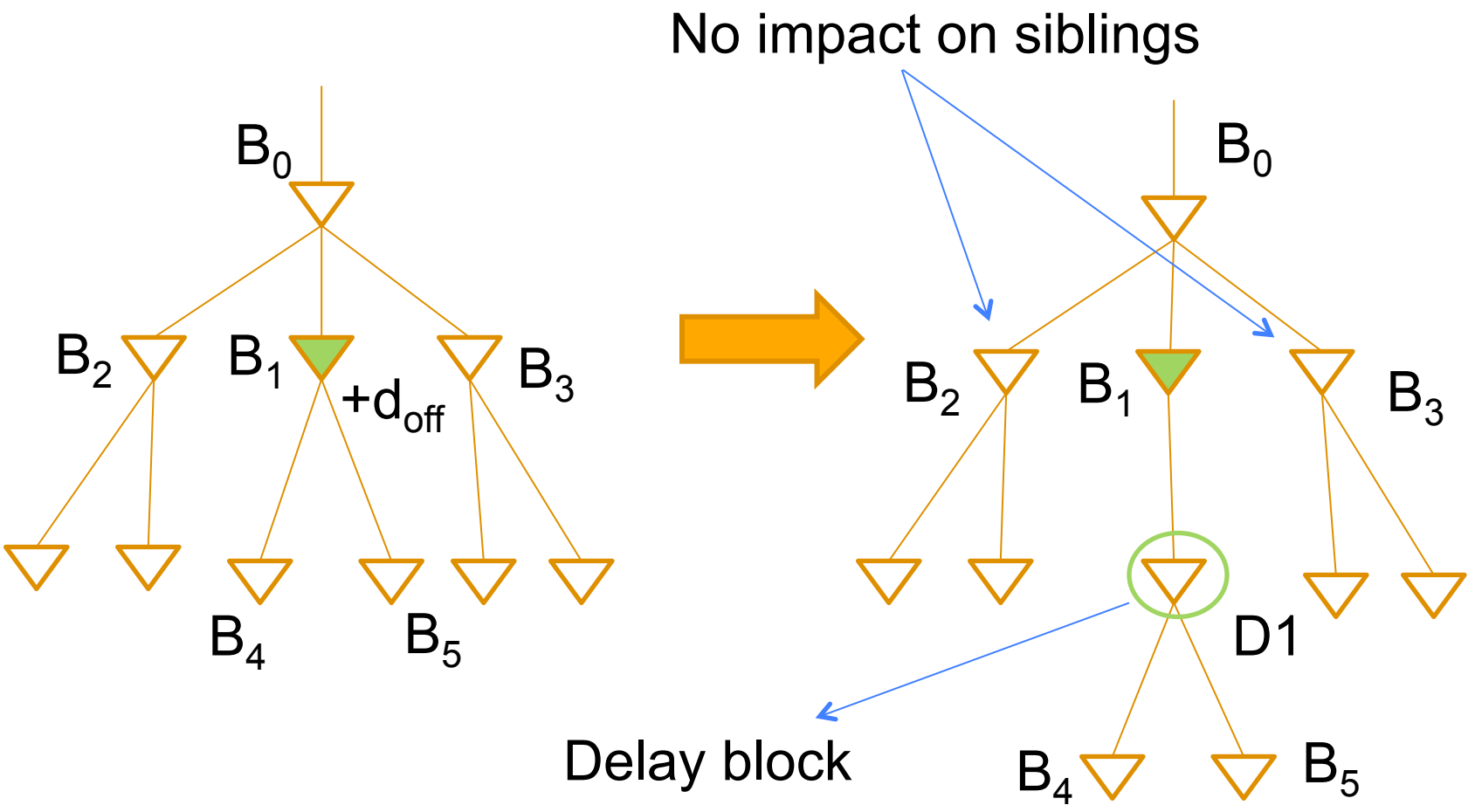
MCMM Offset Estimation

Synthesized/routed clock tree
User specified Offset Range

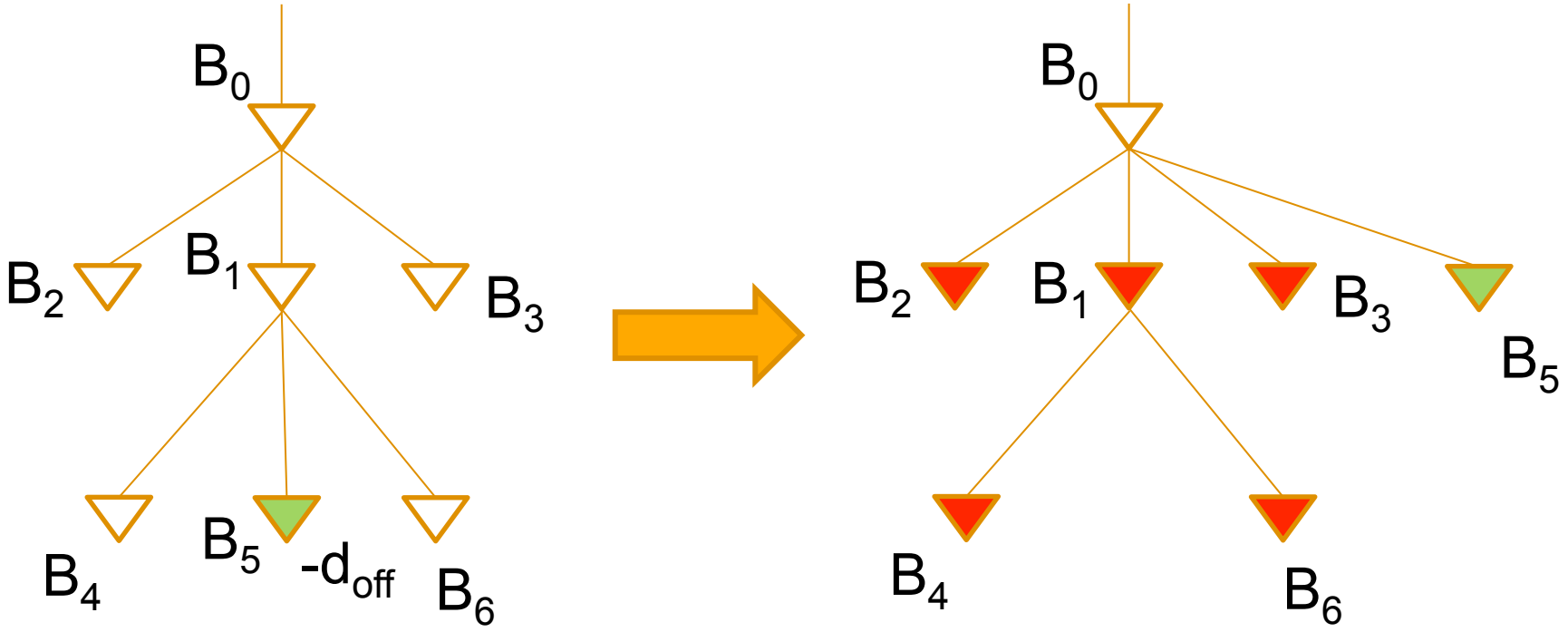
LP Solver [Rama, ISPD'12]

Multi-corner offsets &
TNS/THS improvement prediction

Positive Offset Realization



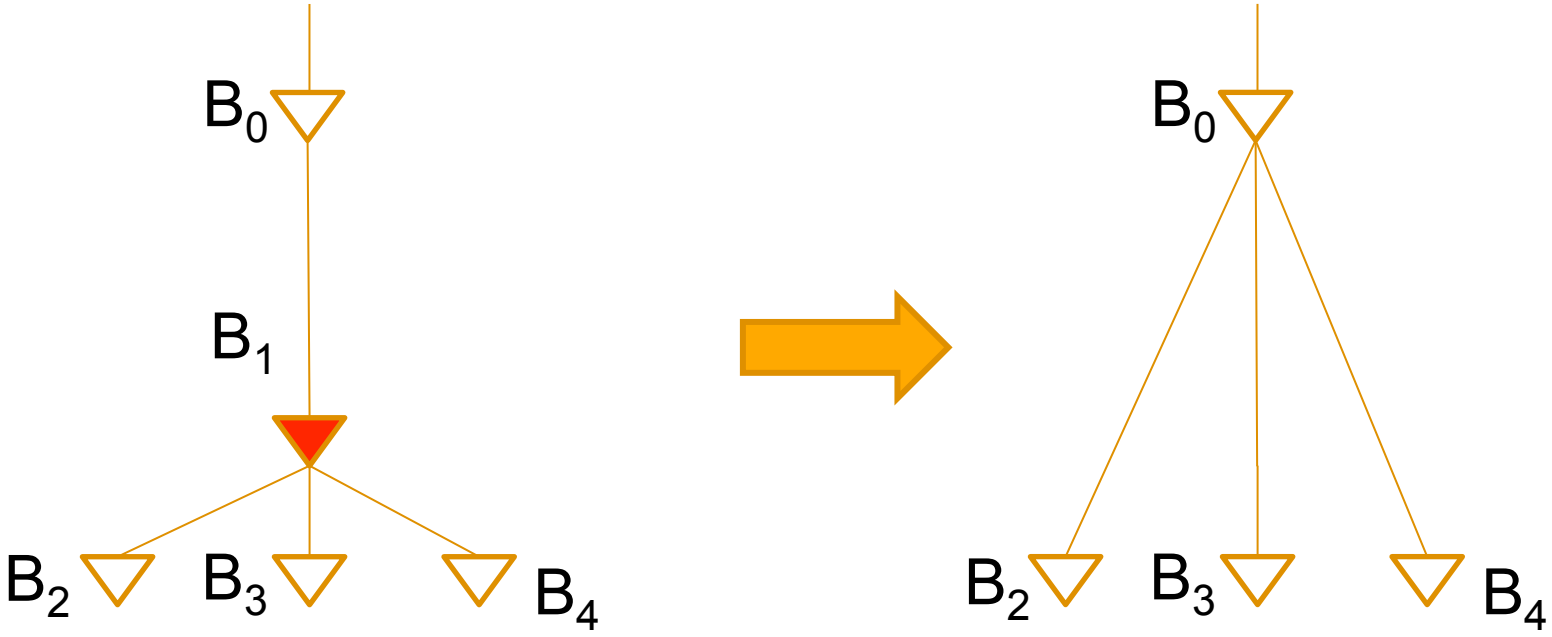
Negative Offset Realization Issues(1)



- Significant impact on timing profile
  - Impact on leaf cells at the TFO cone of old/new siblings of B_5
  - Difficult to guarantee the overall improvement of timing

Negative Offset Realization Issues(2)

- Speed-up by buffer removal may not be practically realizable



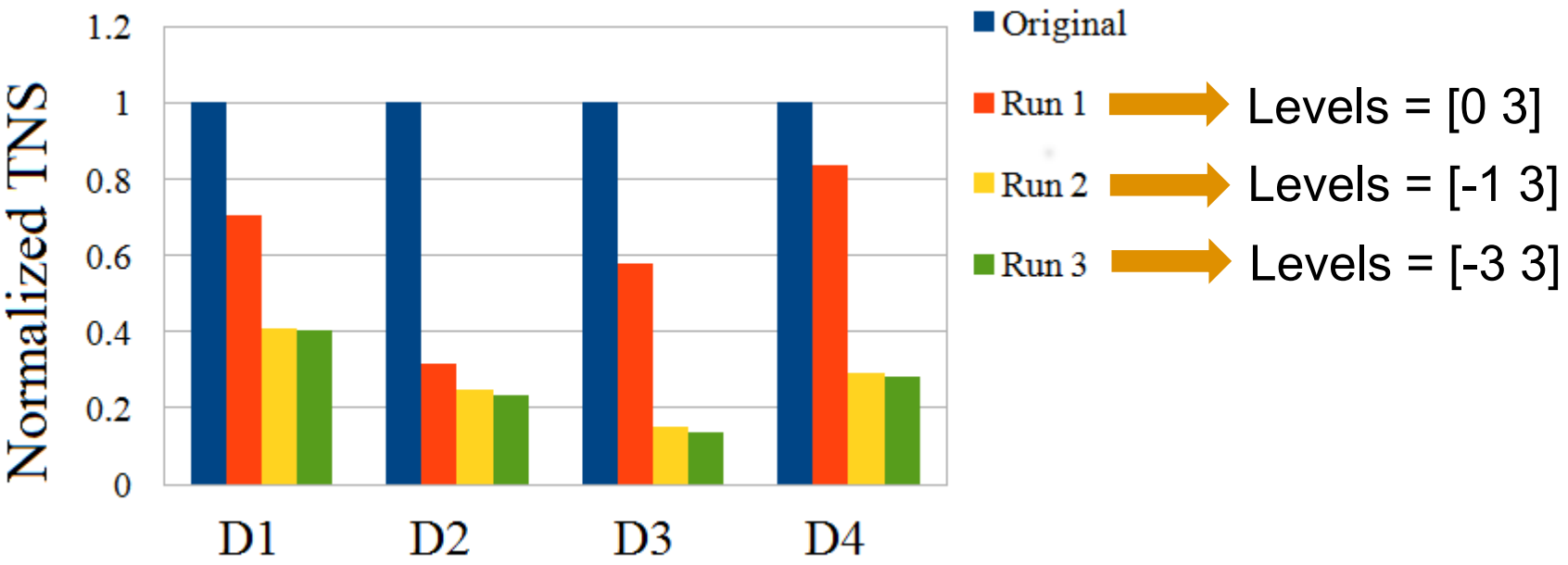
B_0 is driving more load (wire load + buffers) after buffer removal

Offset Bounded Clock Scheduling

- Implementing negative offset is difficult
- For a pin, more the negative offset
  - More the pin needs to be moved upwards tree
  - More FFs downwards the tree will be impacted
- Solution:
  - Calculation and realization of offsets should be tightly coupled
  - Need for offset-bounds

Offset Bounded Clock Scheduling

Offset Bound Experiments

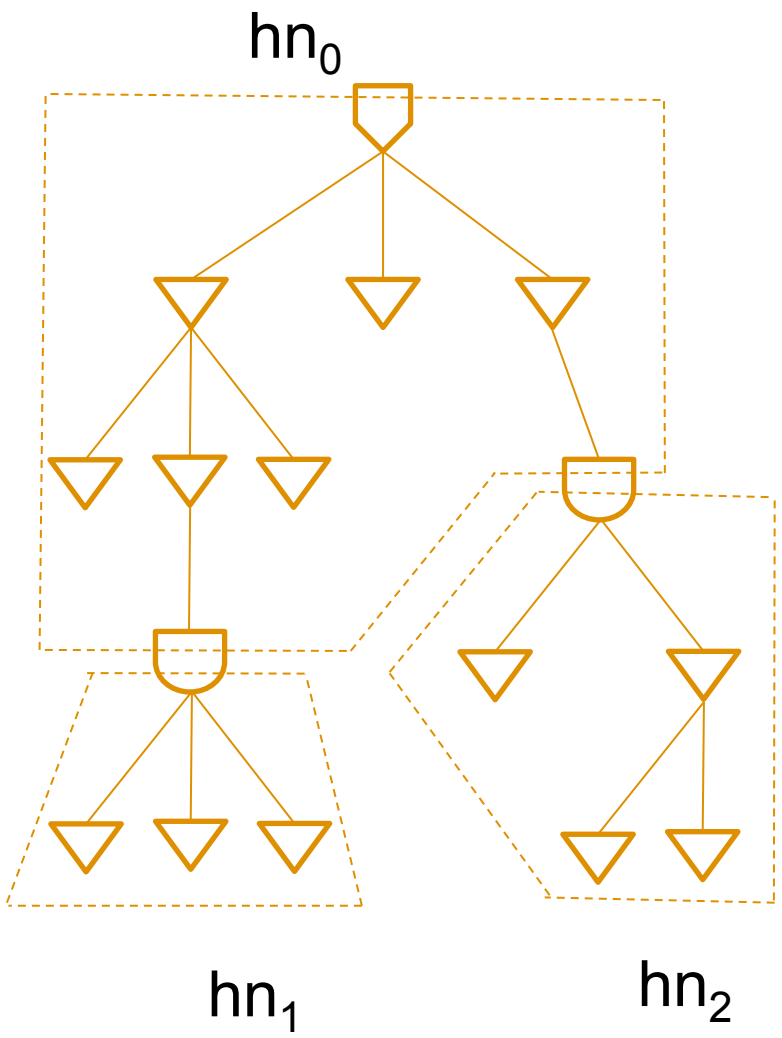


- Discrete offsets in steps of buffer delay (say 50ps)
  - if **Levels** = [-1 1], then possible offset values: -50ps and 50ps

Observation: Hardly any TNS improvement from Run 2 to Run 3

Conclusion: Realize the offsets for Run 2

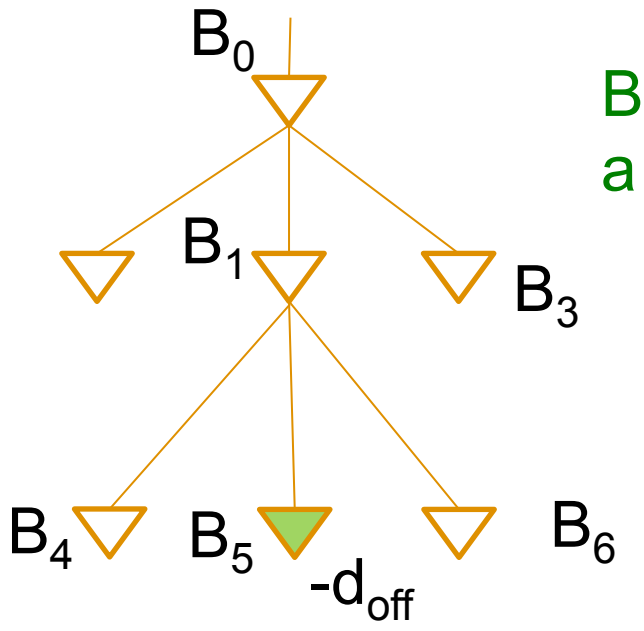
Robust Negative Offset Realization



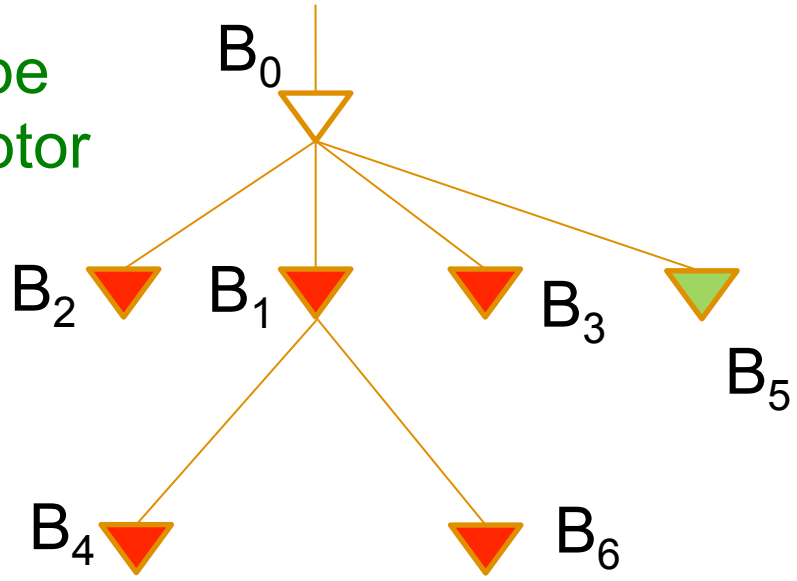
- Any Restructuring should be performed within the scope of hyper-net
  - Clock gating functionality preserved
- Hyper-net → set of nets in same physical partition
  - Nets are logically equivalent or opposite polarity
  - Separated by buffers/inverters
  - Connected in a tree-topology

Robust Negative Offset Realization

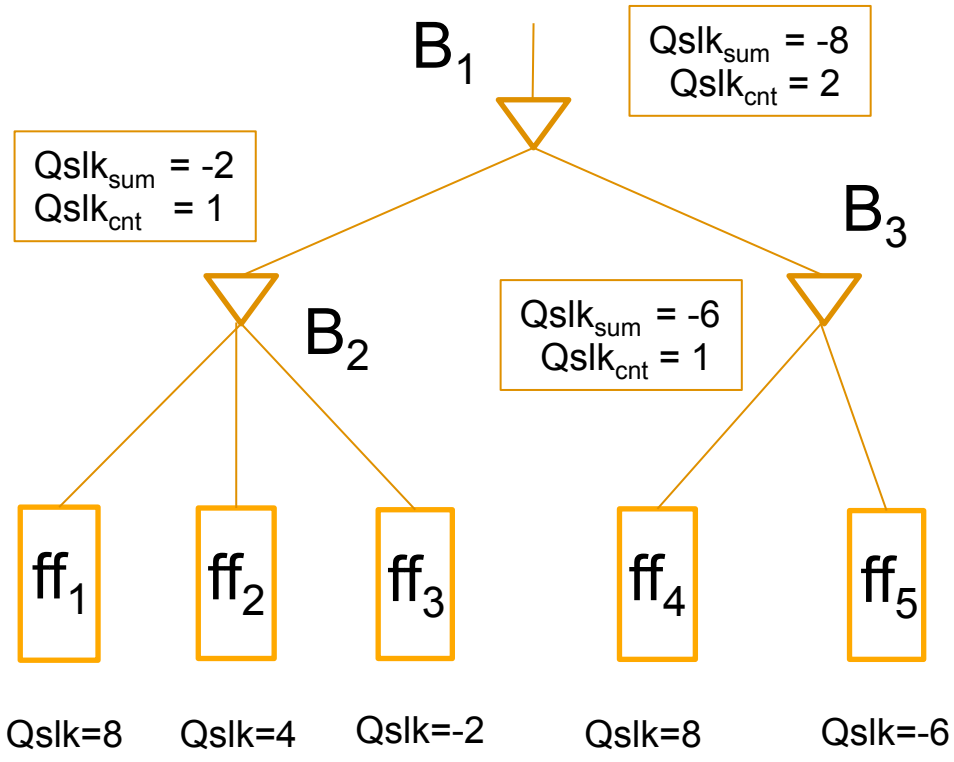
- Restructuring should guarantee no adverse impact on clock-tree under MCMM
- Need to identify potential acceptor pins
  - Sequential cells in TFO should have available positive slack



B_0 needs to be a good acceptor

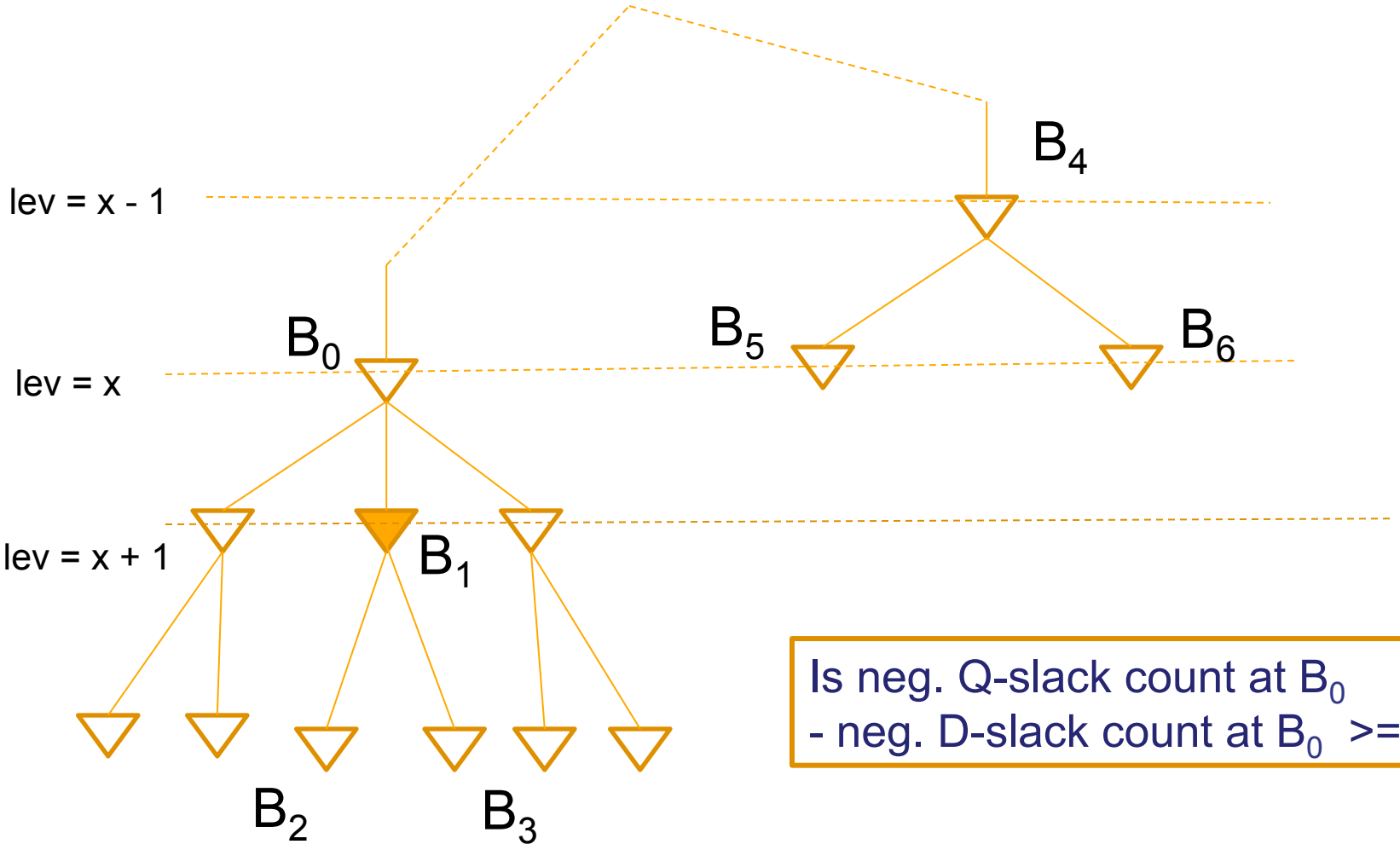


Slack Manager to Identify Acceptors



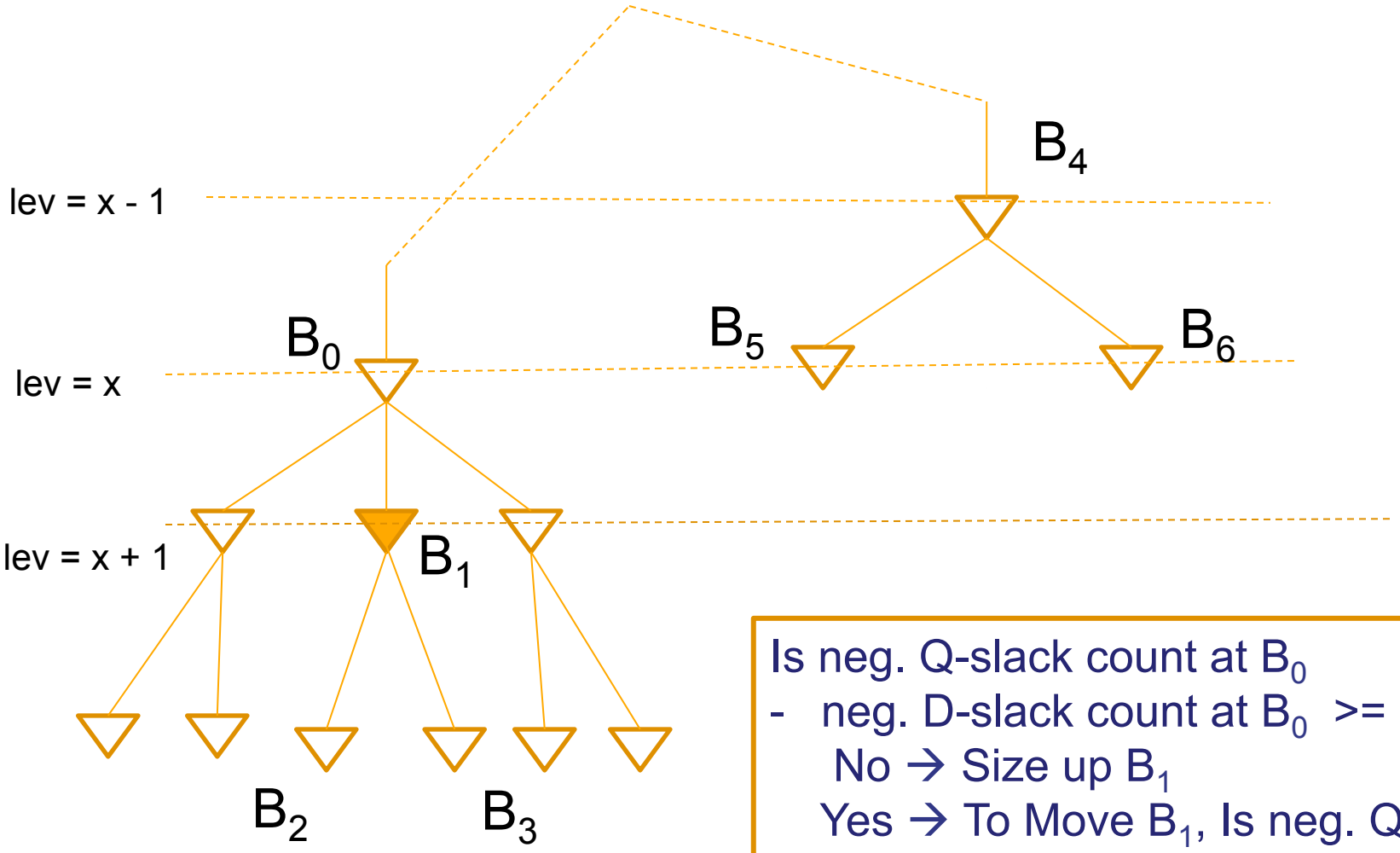
- Same info kept for D-slack parameters
- Slack parameters calculated
  - Per scenario (mode + corner combination)
  - Bottom-up fashion

Clock Tree Restructuring



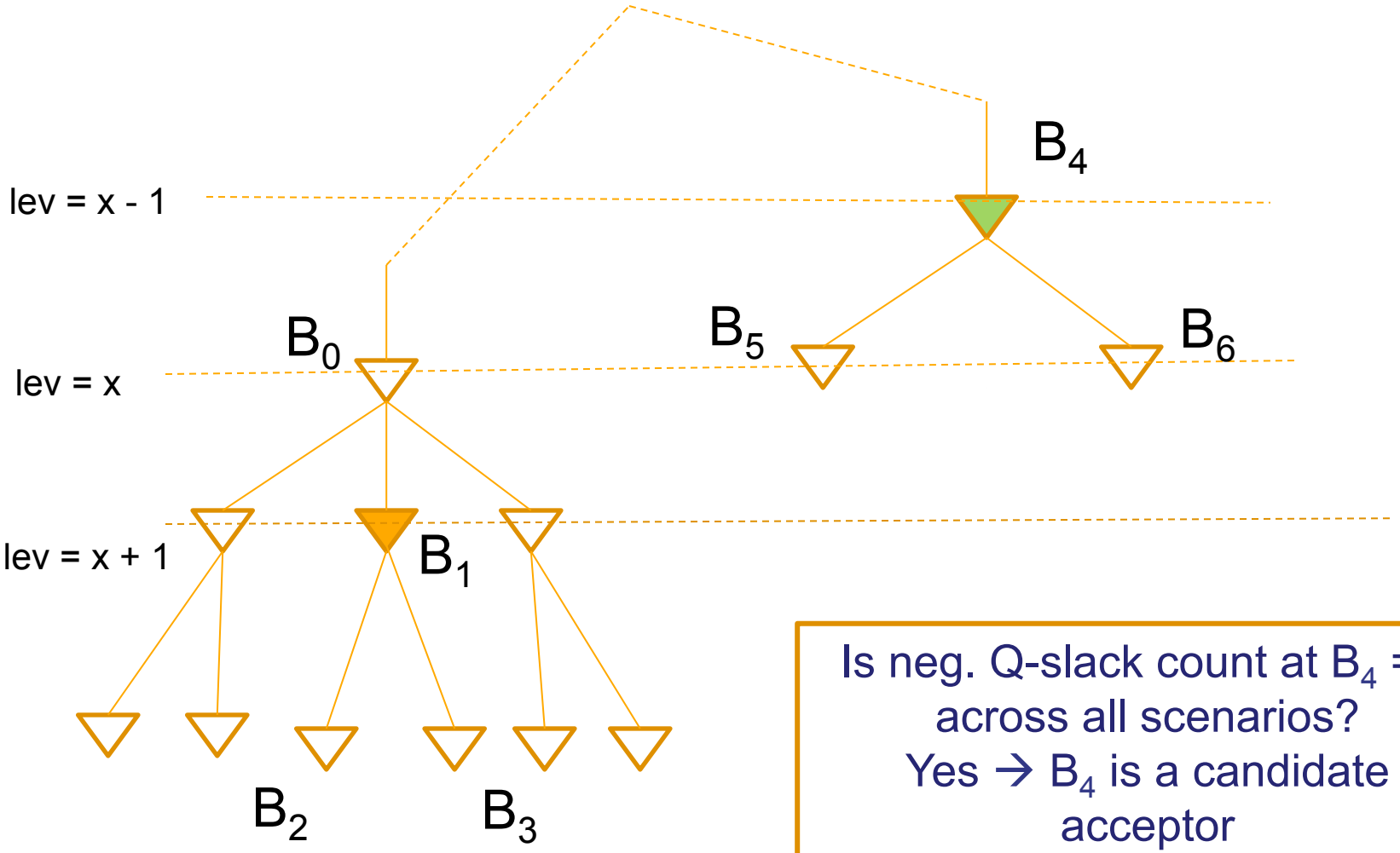
Is neg. Q-slack count at B_0
- neg. D-slack count at B_0 >= 0 ?

Clock Tree Restructuring



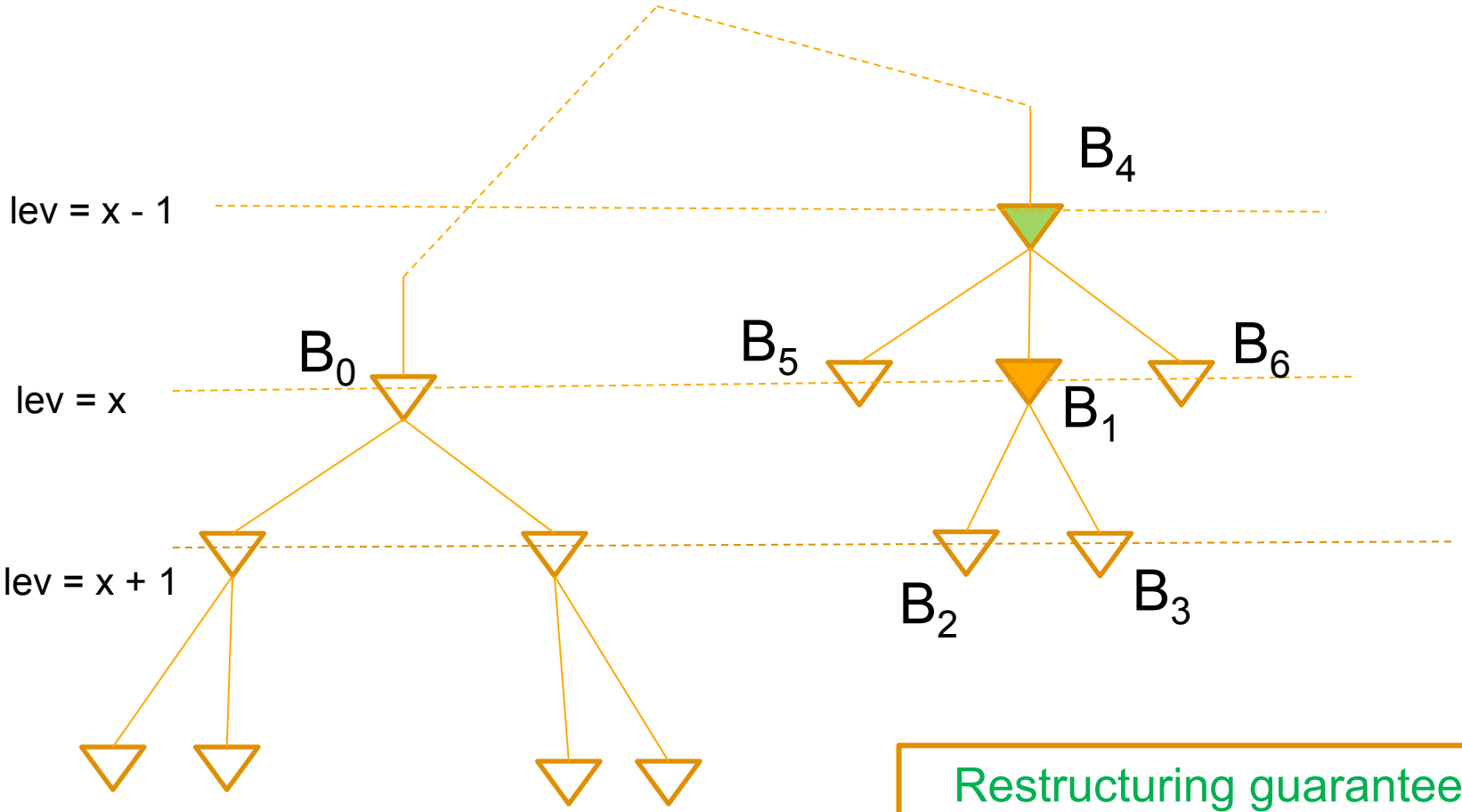
Is neg. Q-slack count at B_0
- neg. D-slack count at B_0 >= 0 ?
  No → Size up B_1
  Yes → To Move B_1, Is neg. Q-slack count at B_4 = 0 across all scenarios?

Clock Tree Restructuring



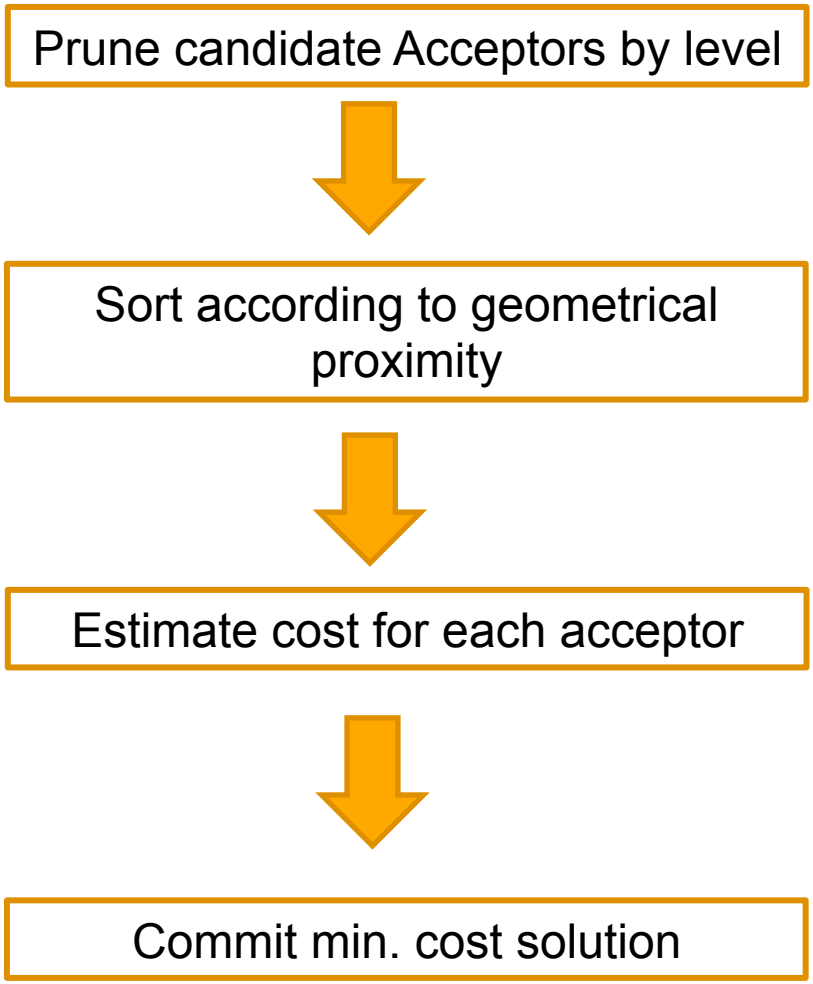
Is neg. Q-slack count at B_4 = 0 across all scenarios?
Yes → B_4 is a candidate acceptor

Clock Tree Restructuring



Restructuring guarantee no adverse impact on FFs at the TFO of B_5 and B_6

Neg. Offset Realization Algorithm (NORA)



Cost Function

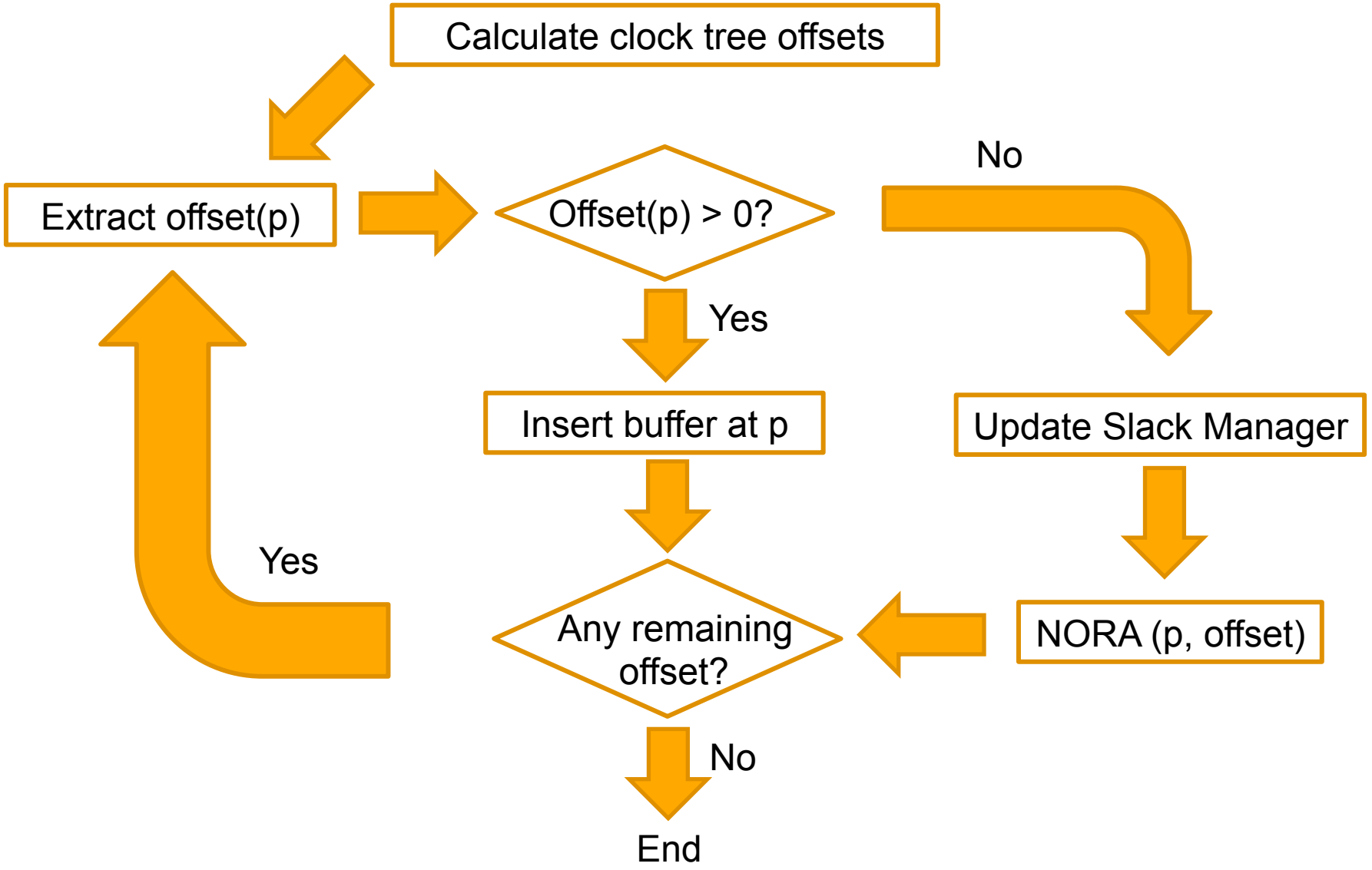
Cost = ∞, if DRC violation
β * (error), o.w.

where, error = inaccuracy in Offset implementation in constraint scenario

Neg. Offset Realization Algorithm (NORA)

- If lot of acceptors, first 10 acceptors considered
  - Saves run time
  - At the same time, area-efficient restructuring
- If no potential acceptor with available slack,
  - Choose the acceptor with max. $Qslack_{sum}$ across all scenarios

Clock Tree Resynthesis Algorithm



Experimental Setup

- Integrated to Industrial P&R tool
- Run on 256GB RAM, 16-core 3GHz CPU
- 7 industrial designs using 20-32nm technology node

Design	Cells (M)	Scenarios	TNS (ps)	WNS (ps)	FEP
A	0.35	5	-789723	-4433	1907
B	0.62	8	-1586320	-414	12850
C	0.62	8	-82529	-218	1262
D	0.7	8	-1129784	-6433	2408
E	0.85	1	-8032671	-1483	17491
F	1.17	5	-8968128	-6394	43938
G	2.03	6	-4289746	-15418	31946

Only Negative Offset Realization

Design	% TNS Imprv.	% WNS Imprv.	% FEP Imprv.	% Clock Tree Overhead	Run Time (min)
A	10.70	-0.13	5.61	2.56	43
B	11.67	0.24	3.61	7.33	175
C	13.35	0.92	9.75	2.56	178
D	32.80	2.64	25.46	1.11	125
E	2.24	2.83	2.20	1.36	98
F	5.91	0.75	7.31	0.17	161
G	34.30	0.08	27.54	0.04	410
Avg.	15.85	1.05	11.64	1.95	-

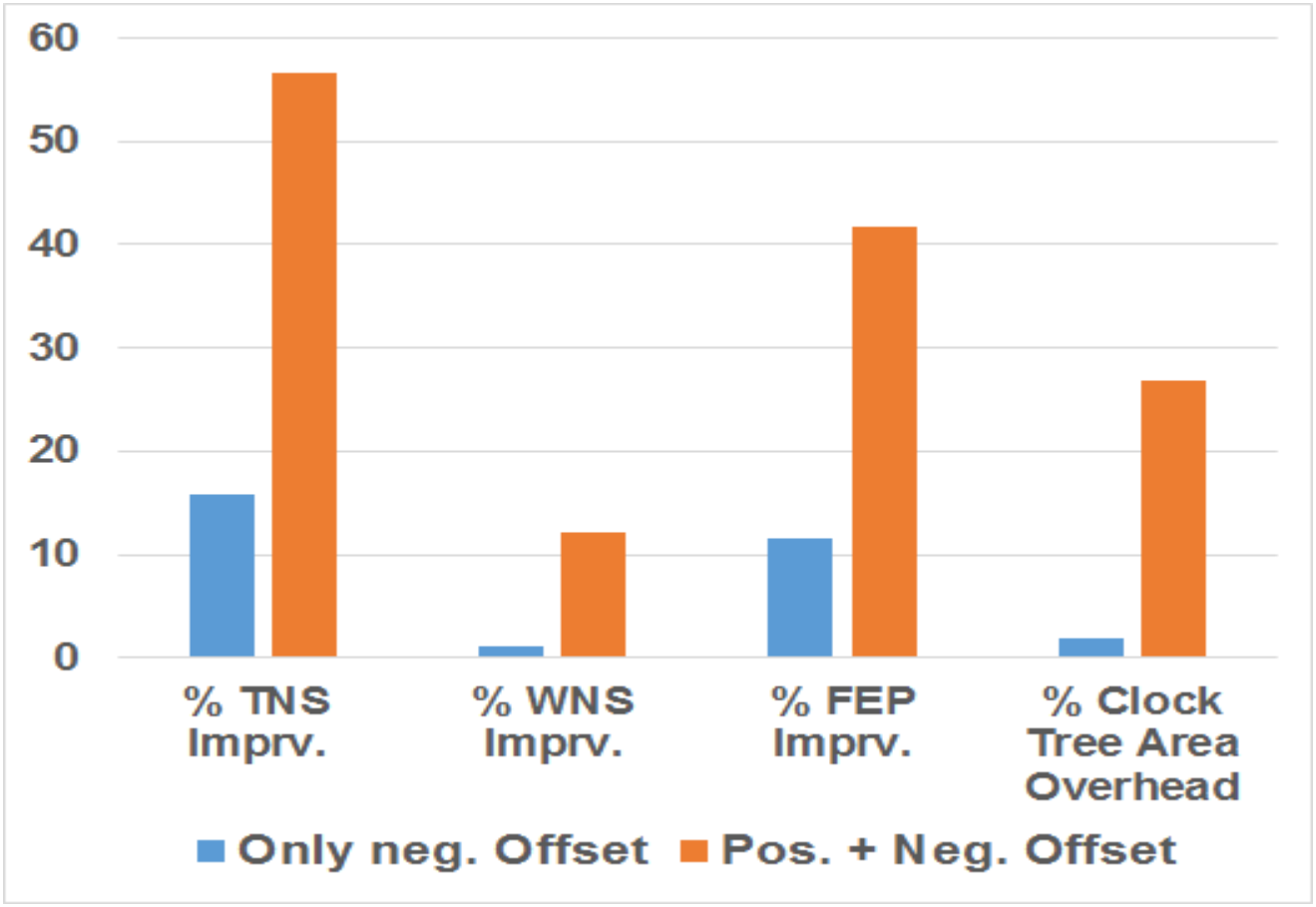
- Restructuring is area-efficient
- Avg. 15.85% improvement in TNS

Pos. and Neg. Offset Realization

Design	% TNS Imprv.	% WNS Imprv.	% FEP Imprv.	% Clock Tree Overhead	Run Time (min)
A	77.65	1.20	39.54	20.10	46
B	56.25	0.97	47.32	47.09	189
C	76.62	49.08	57.84	8.63	140
D	31.58	18.51	17.57	11.51	129
E	69.79	10.05	44.43	54.98	306
F	22.80	0.72	35.69	29.78	250
G	62.09	3.80	50.33	11.12	368
Avg.	56.68	12.04	41.82	26.87	-

- Timing improves more at the cost of clock-tree area
- Avg. 56.68% improvement in TNS

The Overall Comparison



Conclusion and Future Work

- First work to consider offsets at output pins of clock tree cells instead of estimating clock schedule at registers
- A novel clock tree resynthesis methodology presented
- Integrated to Industrial P&R tool
  - Avg. 57% TNS improvement with avg. 26% clock tree area overhead in large-scale MCMM industrial designs

Future Work:

- Concurrent offset realization
- Introduce OCV-impact into the cost function

THANK YOU

Questions?

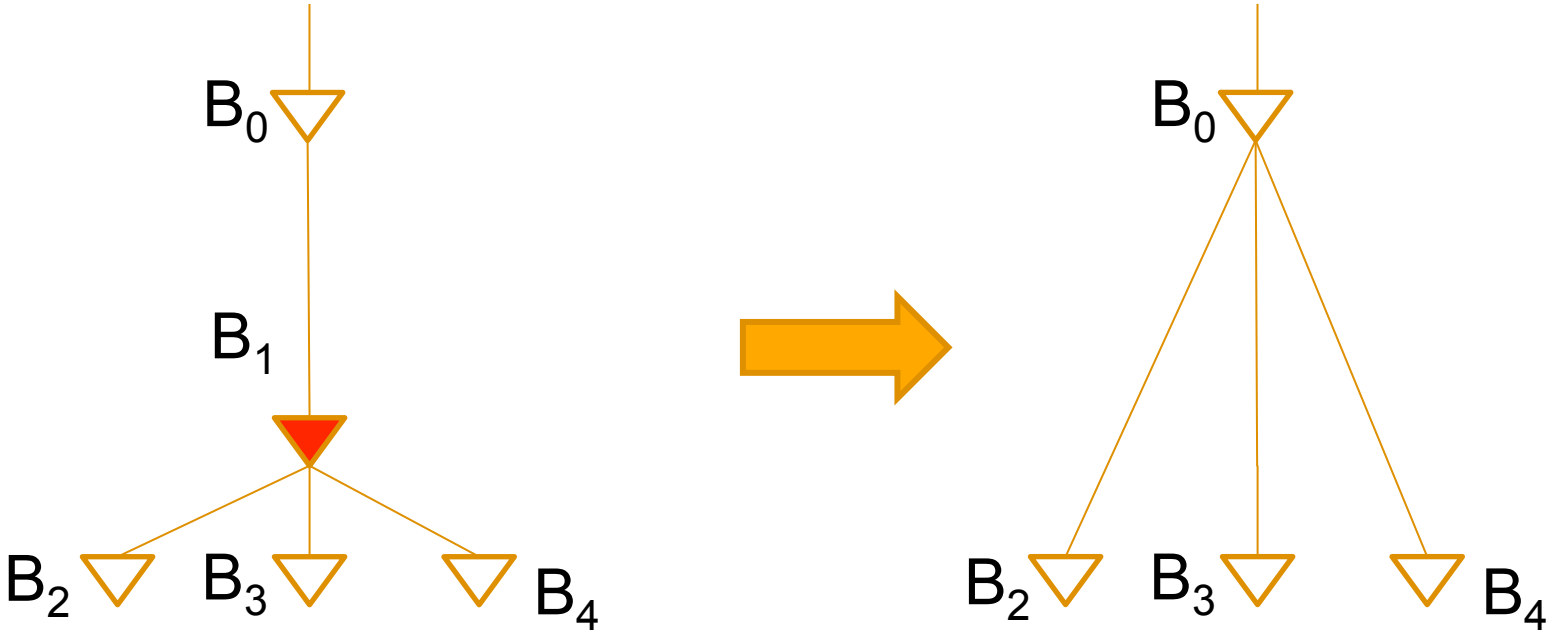
Back-up Slides

Future Work

- Concurrent offset realization
- Clock-tree area overhead is mainly due to pos. offset realization
  - Modify cost function in neg. offset realization
- Introduce the OCV-impact into the cost function
  - Inserting buffer might have adverse effect
  - Restructuring might improve/degrade OCV due to CPPR

Local Transformation

- Speed-up by buffer removal may not be practically realizable



B_0 is driving more load (wire load + buffers) after buffer removal

Our Approach

- Estimate offset (positive/negative) at the clock tree driver pins
  - Performed by an LP solver [Rama12]
  - MCMM scenarios are considered
- Realize the positive/negative offsets incrementally
  - On already synthesized and routed clock tree
  - To ensure rest of the clock tree remains intact

[Rama12] Functional Skew Aware Clock Tree Synthesis by V. Ramachandran, ISPD 2012

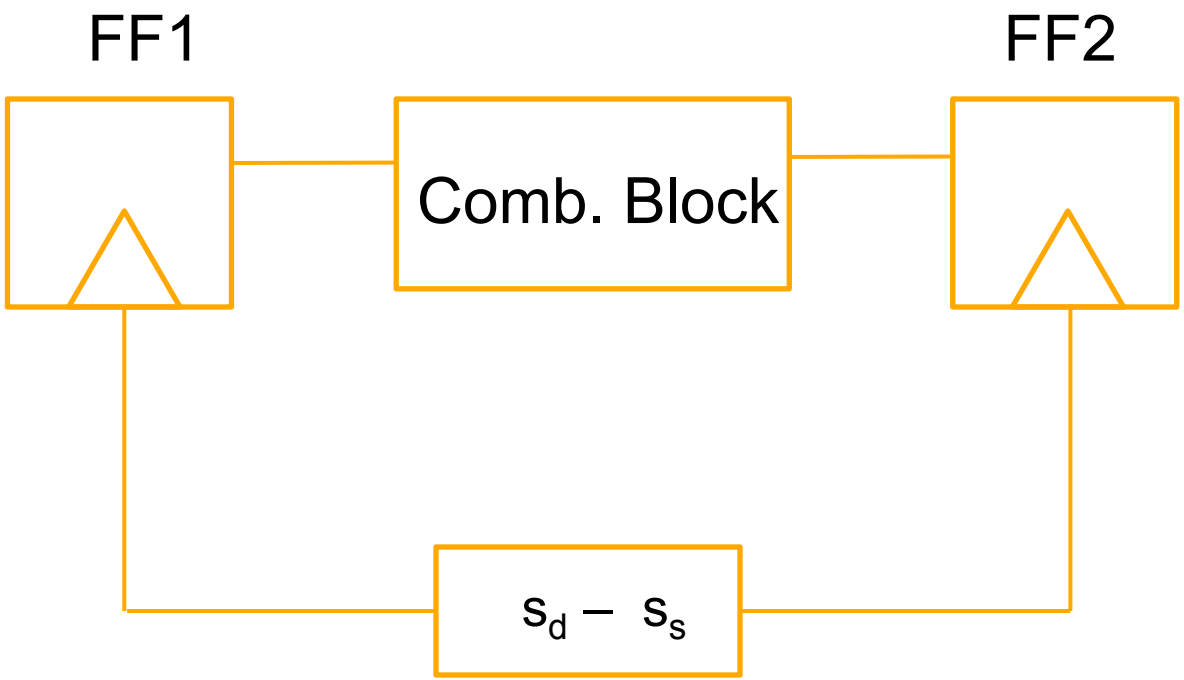
Motivation

- [Kour99],[Naw06] - data path aware clock scheduling
  - Calculate clock skew in pre-CTS stage
  - Actual implementation difficult to achieve
  - Unaware of MCMM scenarios

- [Lu09] – post-CTS bounded delay buffering at leaves
  - Buffering at leaves – high area/power cost
  - Does not tackle MCMM scenarios
  - Only delaying clock arrival – limited scope for optimization

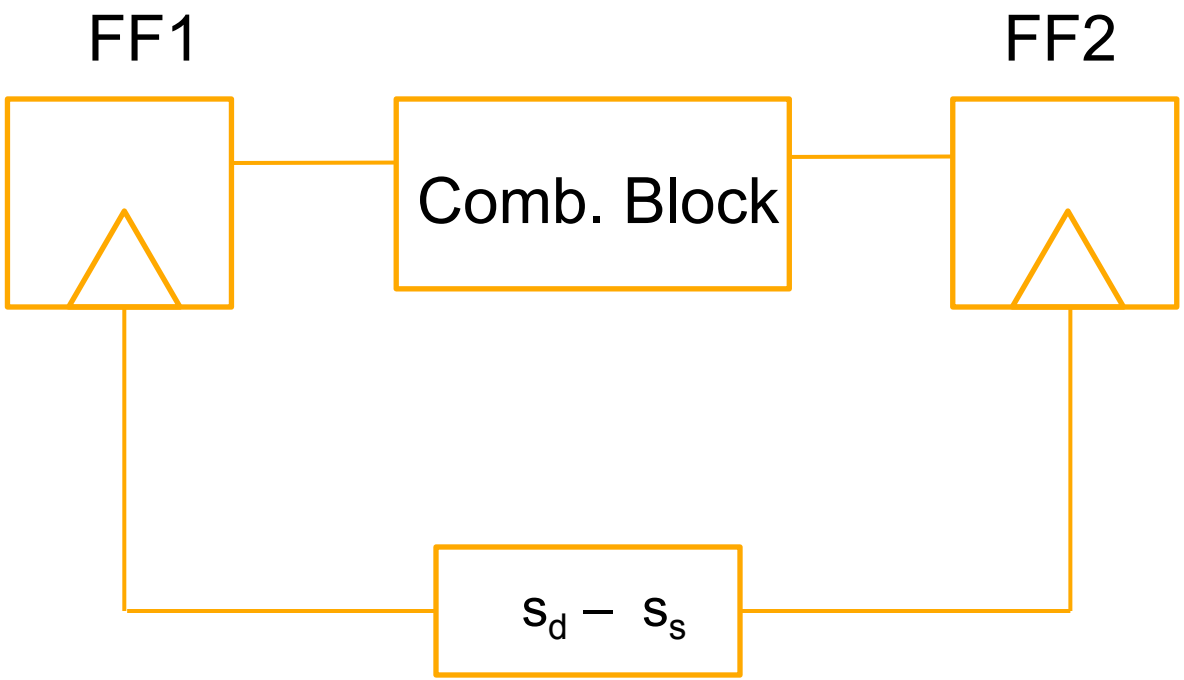
[Kour99] Clock Skew Scheduling for Improved Reliability via Quadratic Programming by Kourtav *et al.*, ICCAD 99
[Naw06] Optimal Useful Clock Skew Scheduling in the Presence of Variations Using Robust ILP Formulations by Nawale *et al.*, ICCAD 2006
[Lu09] Post-CTS Clock Skew Scheduling with Limited Delay Buffering by Lu *et al.*, IMSCS 2009

Preliminaries



$s_d - s_s > 0$ ➔ positive skew
$s_d - s_s < 0$ ➔ negative skew

Preliminaries



Set Up Constraint : $T + (s_d - s_s) > t_{pd,reg} + t_{pd,comb} + T_{su}$

Hold Constraint : $t_{cd,reg} + t_{cd,comb} > (s_d - s_s) + T_h$

Motivation

- Earlier Approach: Clock Skew Minimization
  - Fish90, Tsay91, Kahng92, Chen04

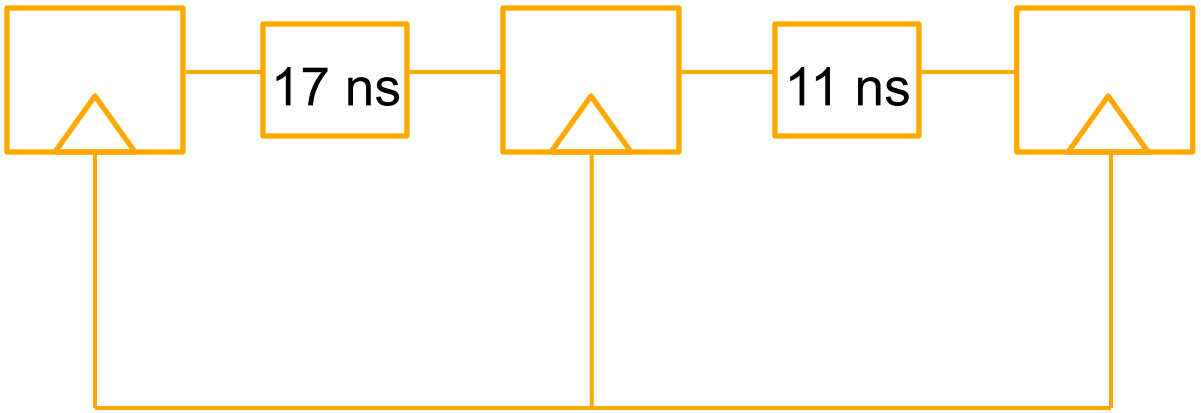
- Issues
  - Maximum operating frequency limited
  - Sacrifice in area/power

[Fish90] Clock Skew Optimization by J P Fishburn, Trans. On Computers 90
[Tsay91] Exact Zero Skew Clock-routing Algorithm by Tsay, ICCAD 91
[Kahng92] Zero Skew Clock Routing Trees with Min. Wirelength by Kahn *et al*., Int. Conf. on ASIC 92
[Chen04] Zero Skew Clock Tree Optimization with Buffer Insertion/Sizing and Wire Sizing by Chen *etal*., IEEE Trans. On CAD 2004

Motivation

$t_{pd,reg}$ = 2 ns
T_{su} = 1 ns

$$T + (s_d - s_s) > t_{pd,reg} + t_{pd,comb} + T_{su}$$

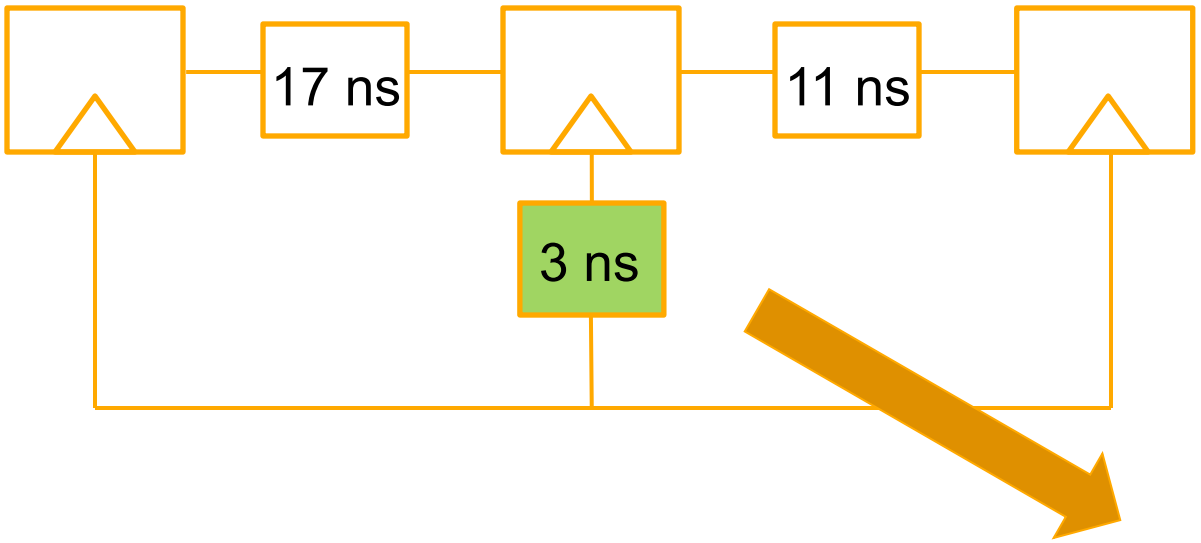


$$T_{clock,min} = 20 \text{ ns}$$

Motivation

$t_{pd,reg}$ = 2 ns
T_{su} = 1 ns

$$T + (s_d - s_s) > t_{pd,reg} + t_{pd,comb} + T_{su}$$



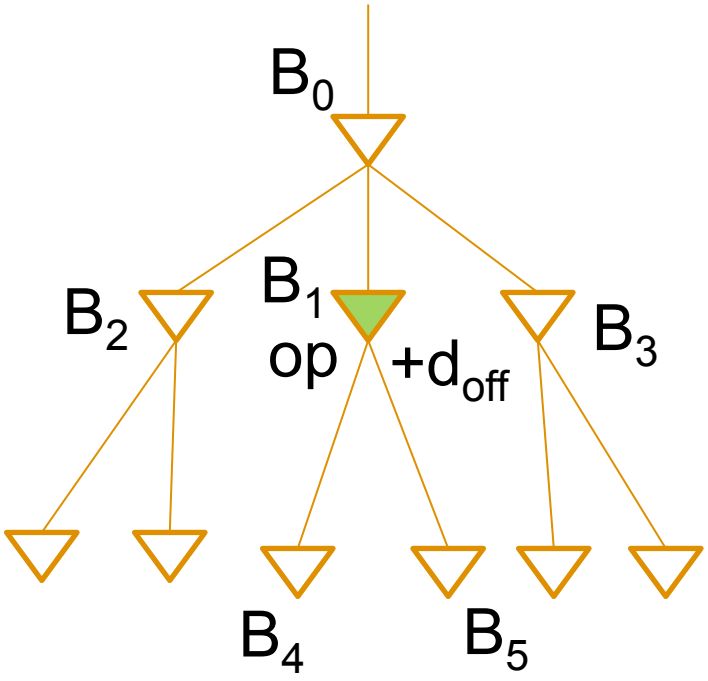
Useful Skew

$$T_{clock,min} = 17 \text{ ns}$$

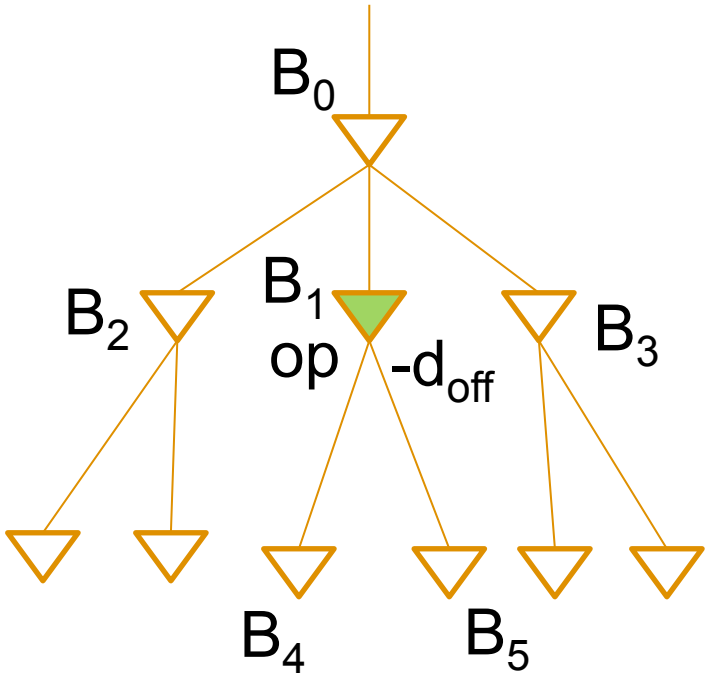
Outline

- Preliminaries
- Motivation
- Our Approach
- Feasibility Aware Clock Scheduling (FACS)
- Clock Tree Resynthesis
- Experimental Results
- Future Work and Conclusion

What is Offset?



Clock-arrival at op to be delayed by d_{off}



Clock-arrival at op to be expedited by d_{off}

Experimental Results

Discussion:

- In design E, clock-tree overhead (54.98%) seems high !
  - But increase in total area is < 1%
- Run time depends on
  - Size of the clock-tree
  - Number of offsets to be realized
- THS optimization with neg. and (pos. + neg.) offset
  - Design B: 14.5%, 88%
  - Design D: 13%, 15%
- Biggest benchmark: 2.03M cells, 6 scenarios
  - 62% improvement in TNS
  - 11% overhead in clock-tree area

Offset Extraction in MCMM

- MCMM Handling
  - Scaling factors calculated for each corner
  - Functional timing paths across all active modes analyzed

- Discrete offsets in steps of buffer delay
  - if **Level** = [-2 3] and D_{buf} = 50 ps, then possible offset values:

    -100 ps, -50 ps, 50 ps, 100 ps and 150 ps