Latency Constraint Guided Buffer Sizing and Layer Assignment for Clock Trees with Useful Skew

Necati Uysal*, Wen-Hao Liu+, Rickard Ewetz*

University of Central Florida*,

Cadence Design Systems+

Outline

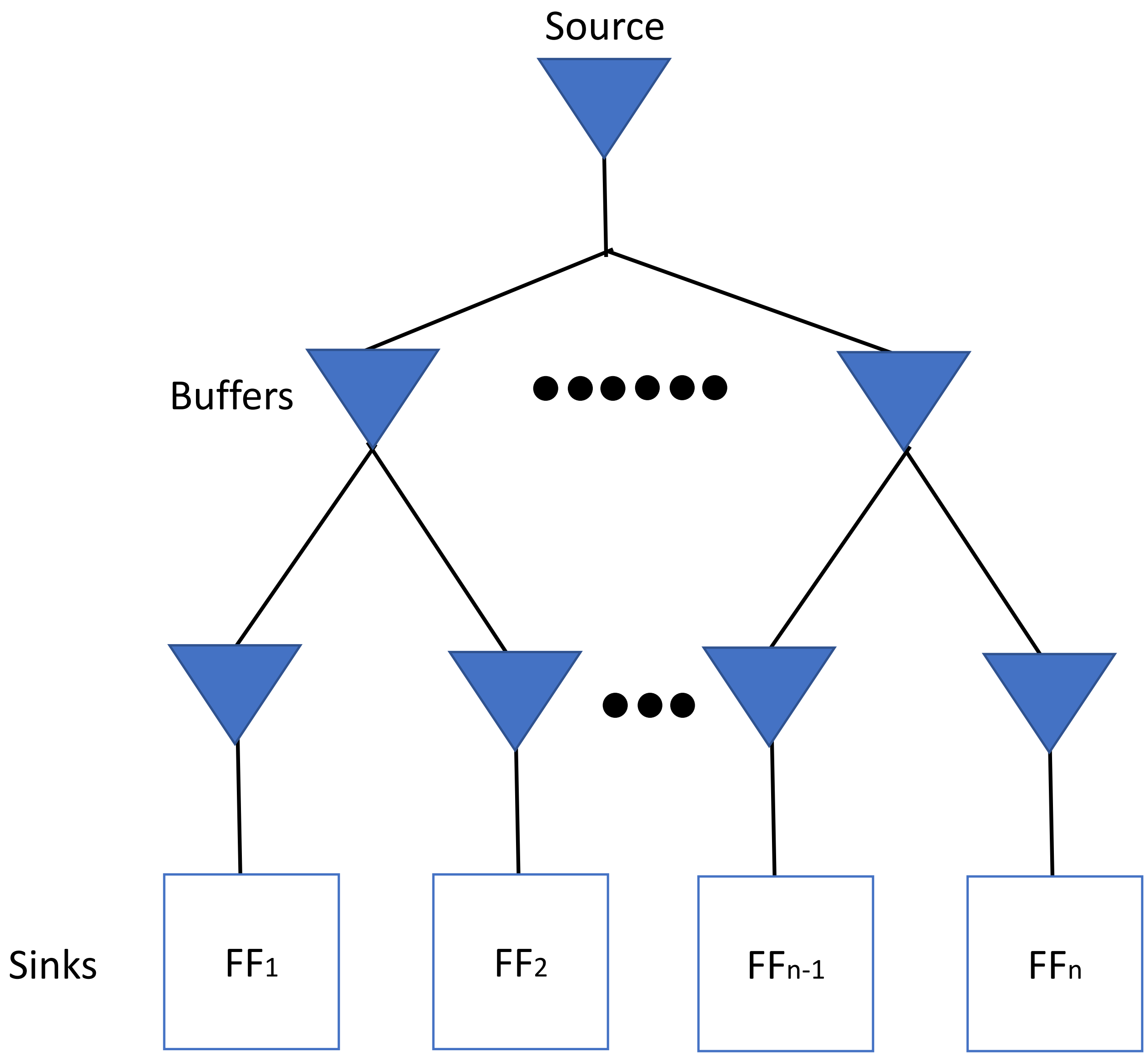
- Introduction
- Preliminaries
- Proposed Techniques
- Experimental Results

Introduction

Clock Tree

- Source
- Flip-flops
- Buffers
- Wires

- Timing constraints
  - Skew
  - Transition time
- Challenges
  - Power consumption
  - On-Chip Variations (OCV)



Timing Constraints

Skew

$$t_{ij} = t_i - t_j$$

Slack

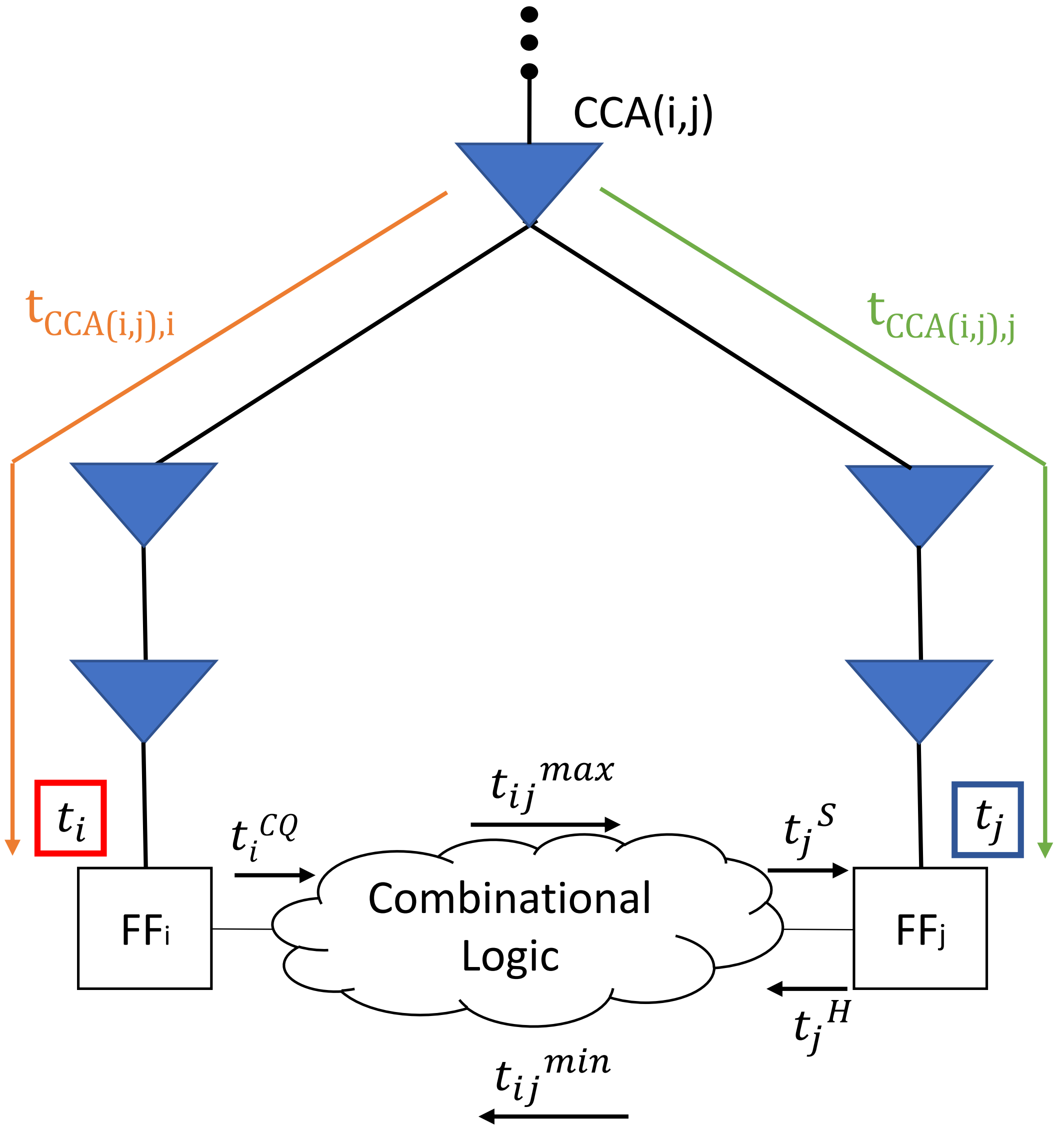
$$setup\ slack_{ij} = \underline{t_j} - \underline{t_i} + T - t_j{}^S - t^{CQ}{}_i - t_{ij}{}^{max} - \delta_i - \delta_j$$

$$hold\ slack_{ij} = \underline{t_i} - \underline{t_j} + t^{CQ}{}_i + t_{ij}{}^{min} - t_j{}^H - \delta_i - \delta_j$$

On-Chip Variations (OCV) [1]

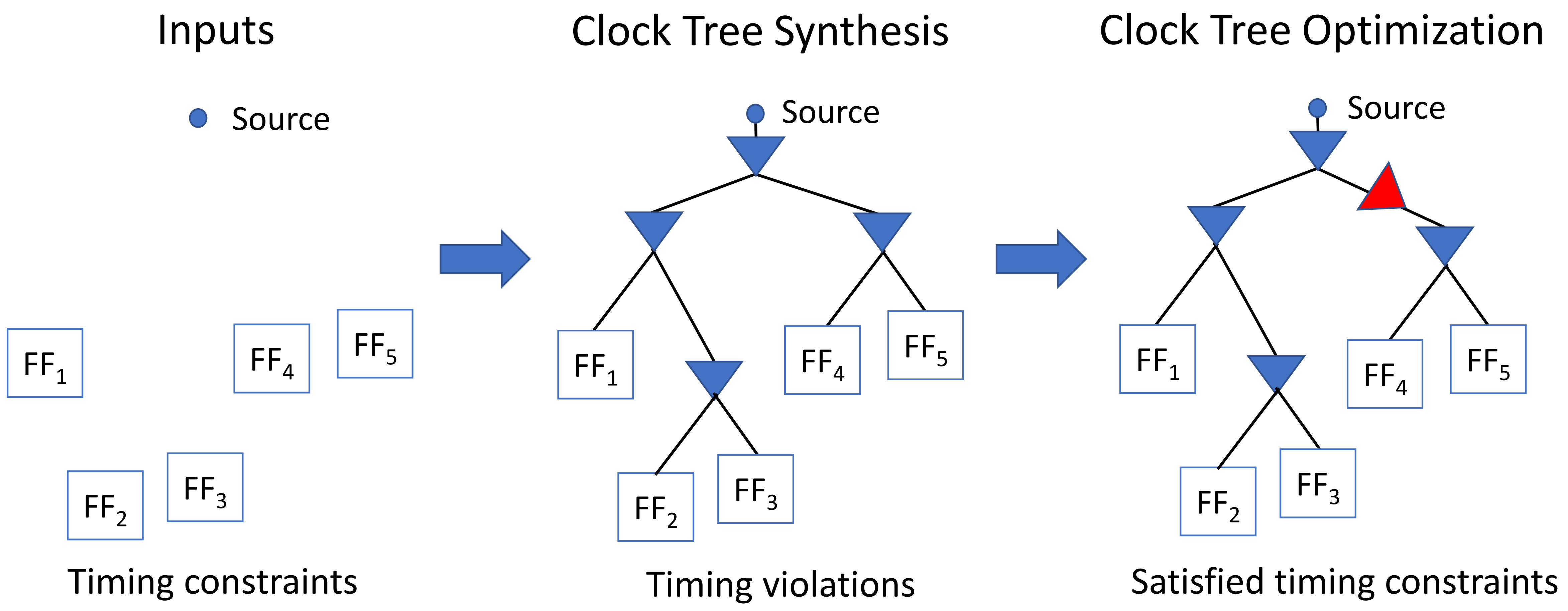
$$\delta_i = c_{ocv}\, t_{CCA(i,j),i}$$

$$\delta_j = c_{ocv}\, t_{CCA(i,j),j}$$



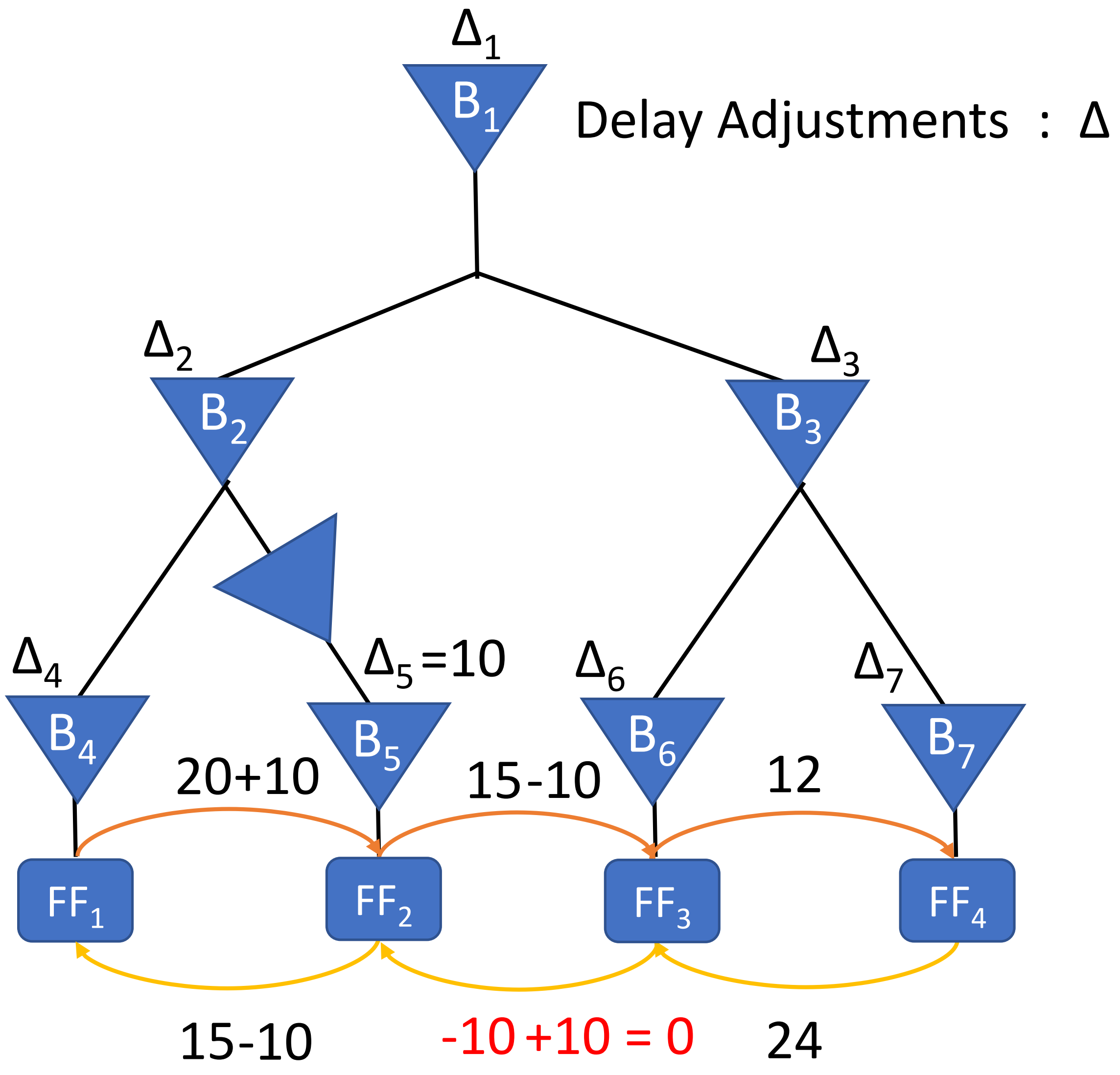
[1]Subhendu Roy et al. 2015. Clock Tree Resynthesis for Multi-Corner Multi-Mode Timing Closure. TCAD (2015), 589–602.

Clock Tree Synthesis Steps



Traditional Clock Tree Optimization

- Steps:
  1. Specify delay adjustments
  2. Realize delay adjustments



LP Formulation & Predicted Timing Quality [2,3]

$$\min \sum c_t \triangle_k + c_{WNS} P_{WNS} + c_{TNS} P_{TNS}$$

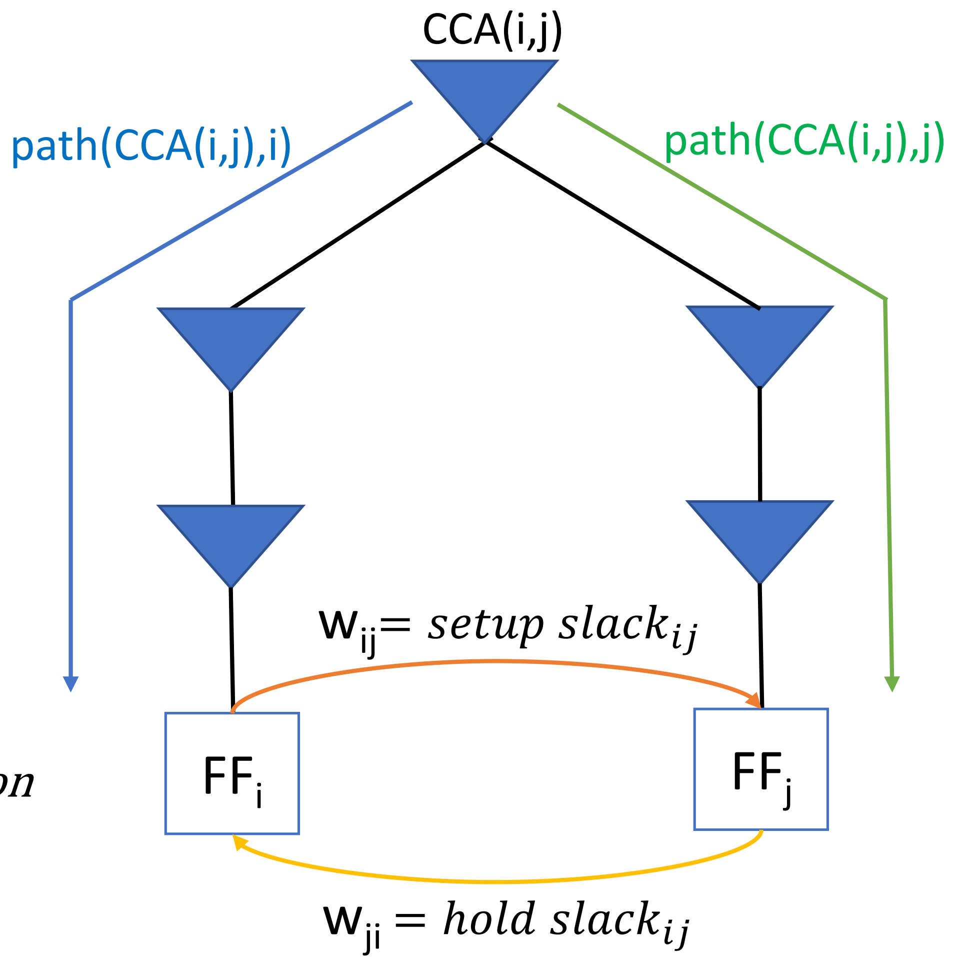
s.t.

$$\sum_{k \in path(CCA(i,j),i)} (1+c_{ocv})\triangle_k - \sum_{h \in path(CCA(i,j),j)} (1-c_{ocv})\triangle_h - s_{ij} \leq w_{ij}, \quad (i,j) \in E,$$

$$s_{ij} \leq P_{WNS}, \quad (i,j) \in E,$$

$$\sum_{(i,j) \in E} s_{ij} = P_{TNS},$$

$s_{ij} \geq 0 \rightarrow$ *timing violation*
predicted TNS $\rightarrow P_{TNS}$
predicted WNS $\rightarrow P_{WNS}$



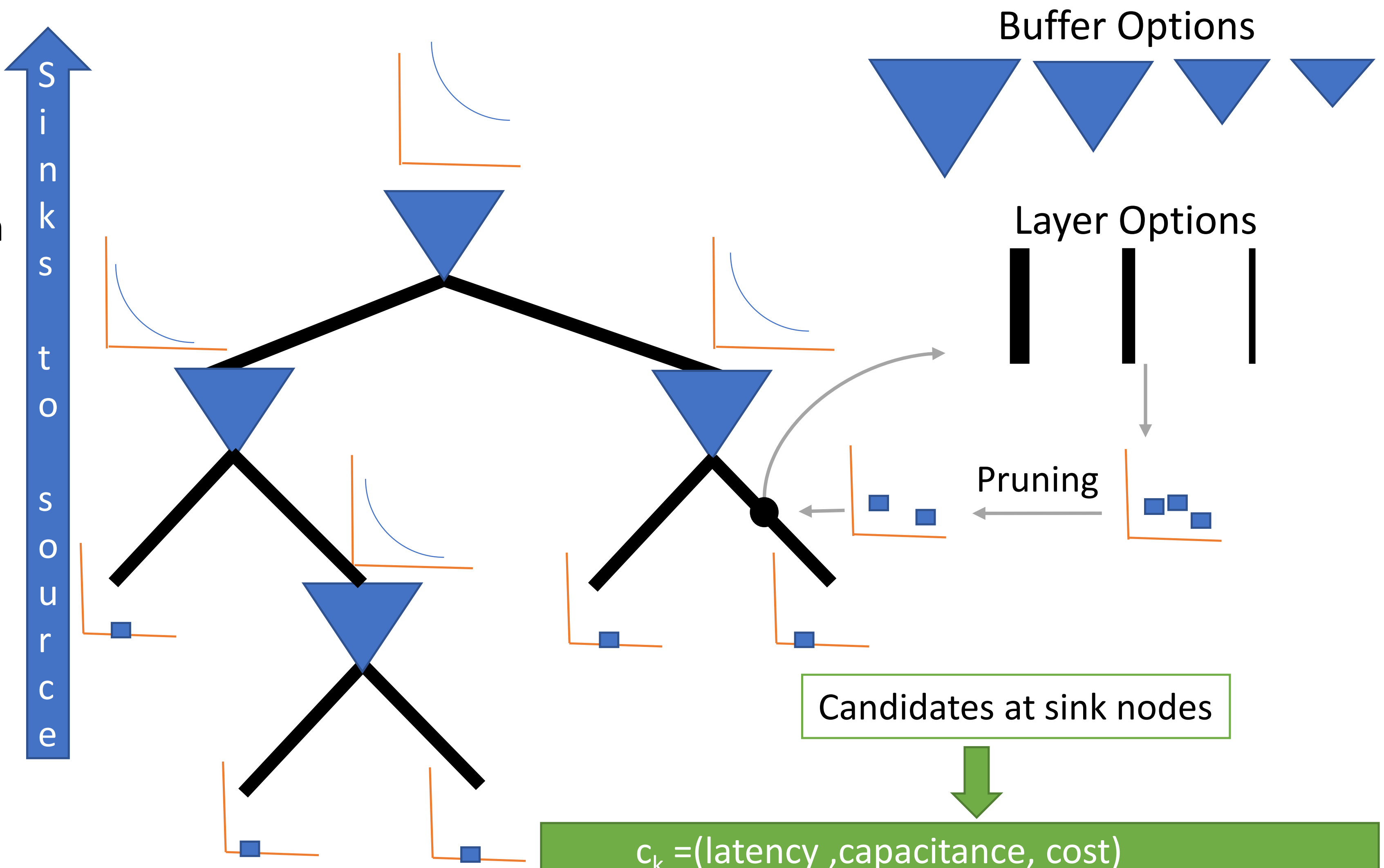
[2] Jianchao Lu and Baris Taskin, "Post-CTS Clock Skew Scheduling with Limited Delay Buffering", Proceedings of the IEEE International Conference on Midwest Circuits and Systems (MWSCAS), August 2009, pp. 224--227.

[3] Rickard Ewetz. 2017. A Clock Tree Optimization Framework with Predictable Timing Quality (DAC'17). 13–18.

Buffer Sizing and Layer Assignment: Van Ginneken's Algorithm [4]

Objective:

- Satisfy maximum latency constraint with minimum cost



c_k =(latency ,capacitance, cost)

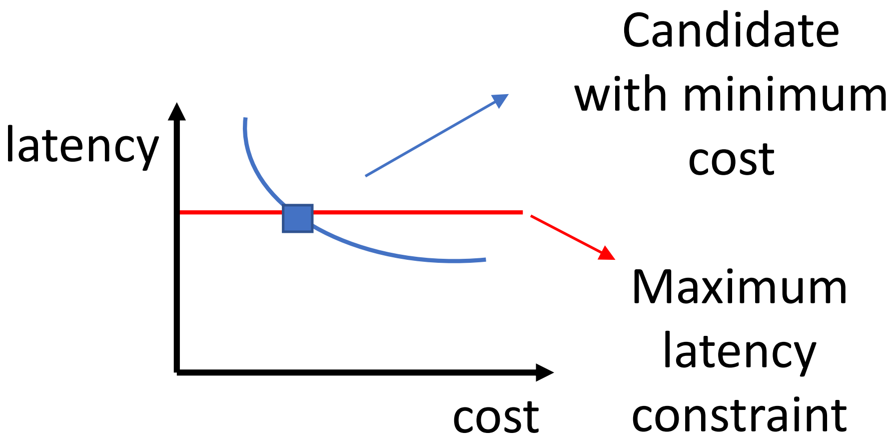
c_{sink} =(0,load capacitance, load capacitance)

[4] L. P. P. P. van Ginneken. 1990. Buffer placement in distributed RC-tree network for minimal Elmore delay. In Proc. IEEE Int. Symp. Circuits Syst. 865–868.

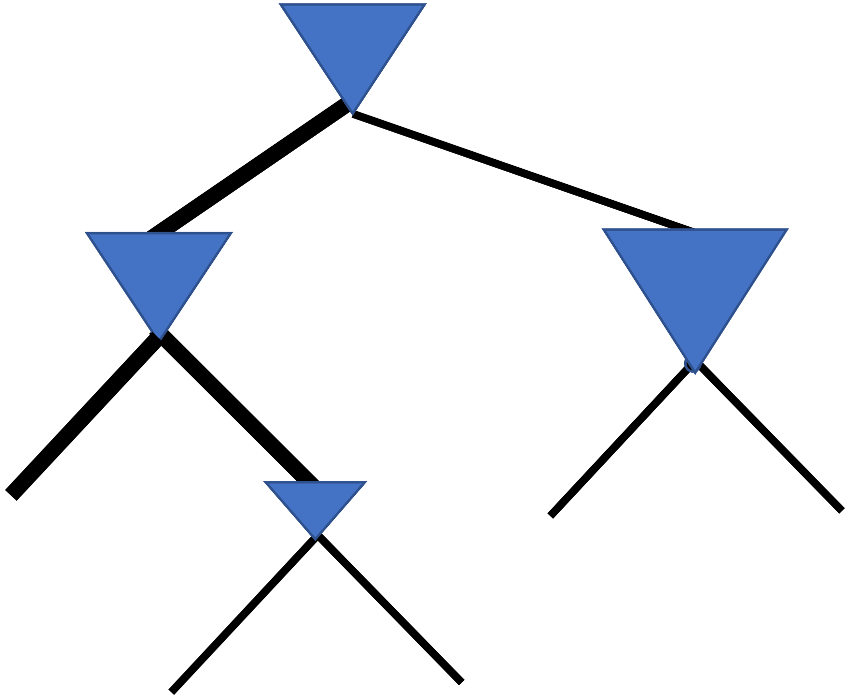
[5] J. Lillis, Chung-Kuan Cheng, and T. T. Y. Lin. 1996. Optimal wire sizing and buffer insertion for low power and a generalized delay model. IEEE Journal of Solid-State Circuits 31, 3 (1996).

Van Ginneken's Algorithm cont.

Candidates at root



Clock Tree after buffer sizing and layer assignment



Previous Works

Reference	Discrete/Continuous Buffer Sizes	Skew Type
[6]	Continuous	Zero
[7]	**Discrete**	Zero
[8]	**Discrete**	Bounded
[9]	Continuous	**Useful**
[10]	Continuous	**Useful**
This work	**Discrete**	**Useful**

[6] Krit Athikulwongse, Xin Zhao, and Sung Kyu Lim. 2010. Buffered Clock Tree Sizing for Skew Minimization Under Power and Thermal Budgets (ASP-DAC'10).
[7] Jeng-Liang Tsai, Tsung-Hao Chen, and C. C. P. Chen. 2004. Zero skew clock-tree optimization with buffer insertion/sizing and wire sizing. TCAD 23, 4 (2004),
[8]Logan Rakai et al. 2013. Buffer Sizing for Clock Networks Using Robust Geometric Programming Considering Variations in Buffer Sizes (ISPD'13).
[9]Matthew R. Guthaus, Dennis Sylvester, and Richard B. Brown. 2006. Clock Buffer and Wire Sizing Using Sequential Programming (DAC'06).
[10]Kai Wang and Malgorzata Marek-Sadowska. 2004. Buffer Sizing for Clock Power Minimization Subject to General Skew Constraints (DAC'04).

Problem Statement

- Clock Tree Optimization
  - Objective : Minimize **timing violations**, amount of delay insertion and **total capacitance**
- Inputs
  - A constructed clock tree
  - Discrete buffers and layers
- Constraints:
  - Skew (timing) constraints
  - Slew

Motivation

Traditional CTO

Specify delay adjustments + Buffer insertion and detour wires =

- Reduces timing violations
- Increases capacitance

Proposed CTO

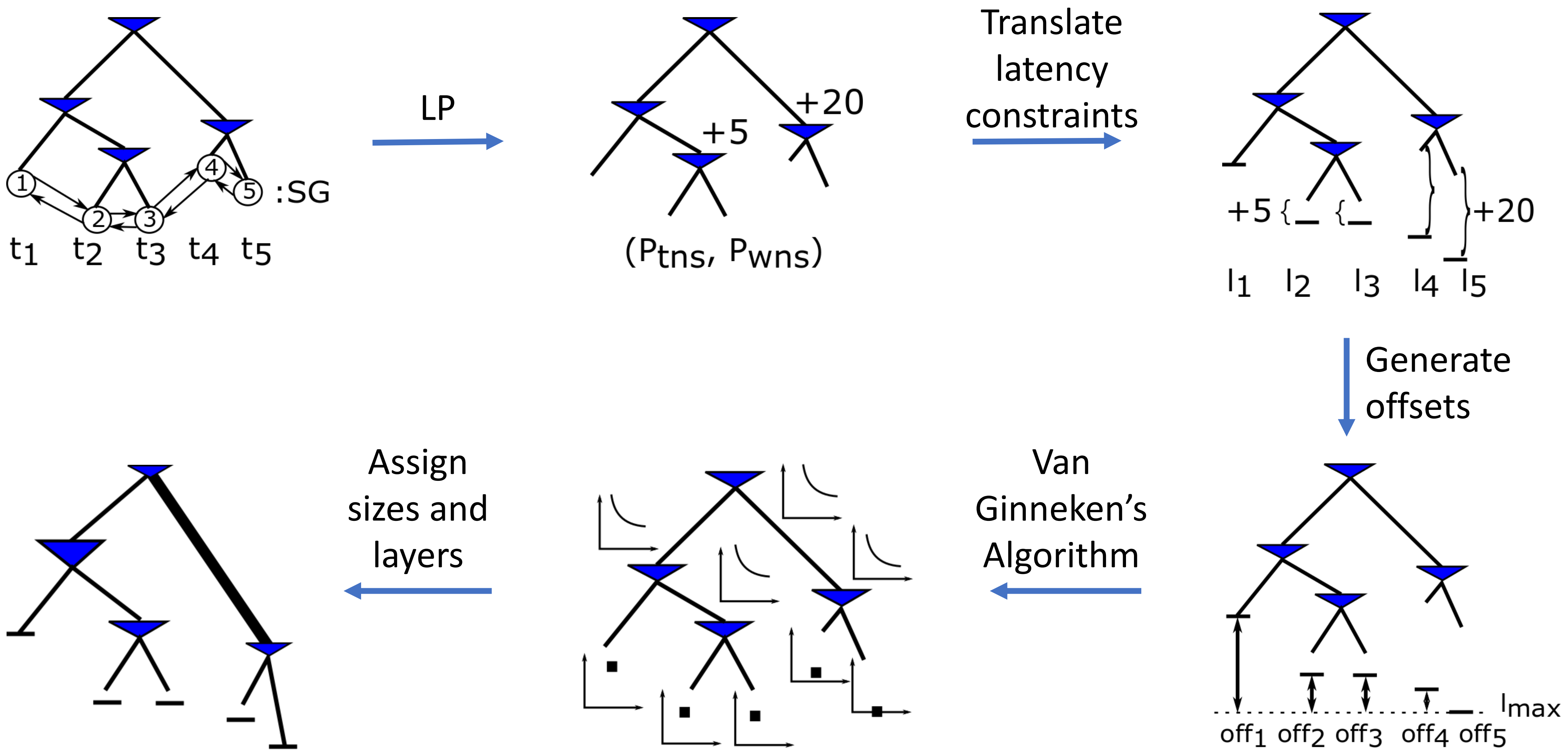
Specify delay adjustments + Van Ginneken's Algorithm + (If necessary) Buffer insertion and detour wires =

- Reduces timing violations by satisfying latency constraints
- Reduces capacitance

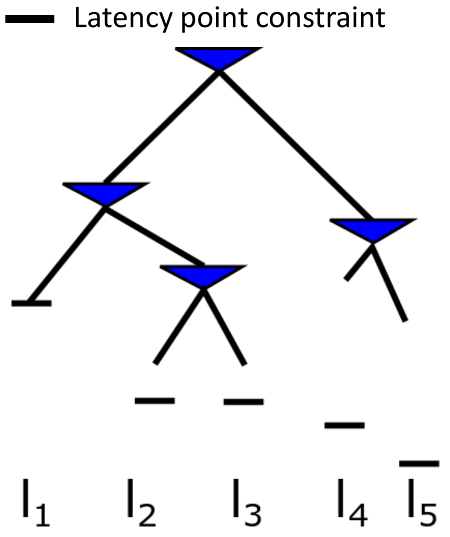
Proposed Solution : BLU Framework

- Method: Delay adjustments are translated into latency constraints such that van ginneken's algorithm can realize delay adjustments.
- Feature I : To further reduce total capacitance
- Feature II: To realize negative delay adjustments

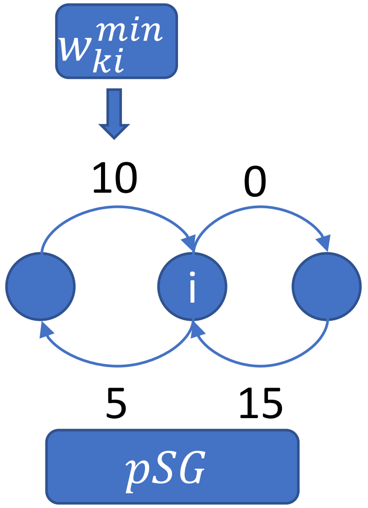
Baseline of the BLU Framework



Feature I: Relaxing point constraints (l_i) into range constraints (l^r_i)



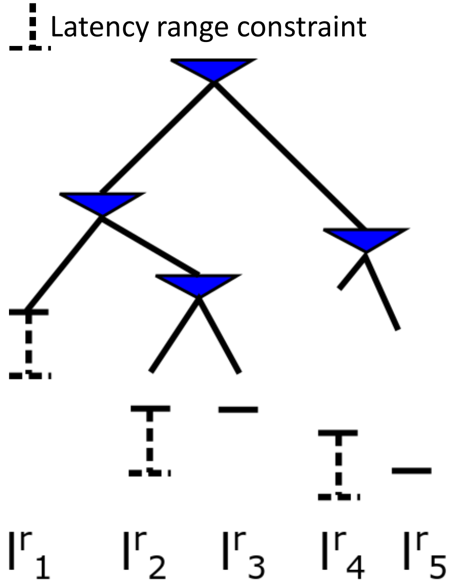
Predicted Slack Graph is formed.



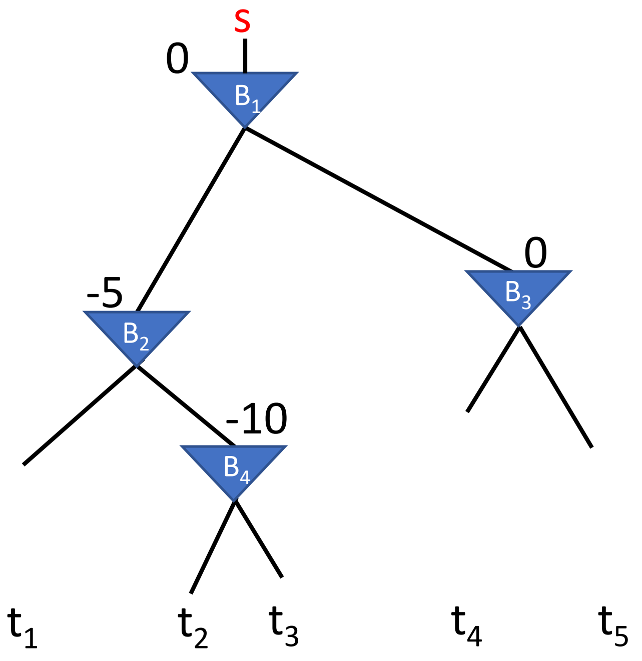
Range constraints at sinks are computed considering OCV variations.

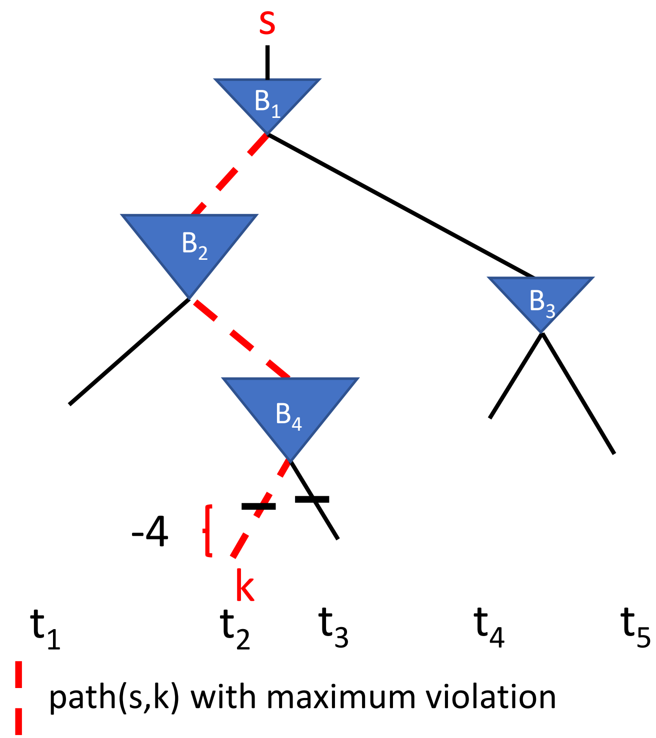
$$l_{ri} = l_i + \frac{w_{ki}^{min}}{(1+ c_{ocv})}$$

Non-critical paths are depromoted to reduce cost.



Feature II: Improving predicted timing quality





Negative delay adjustments can improve P_{TNS} and P_{WNS}.

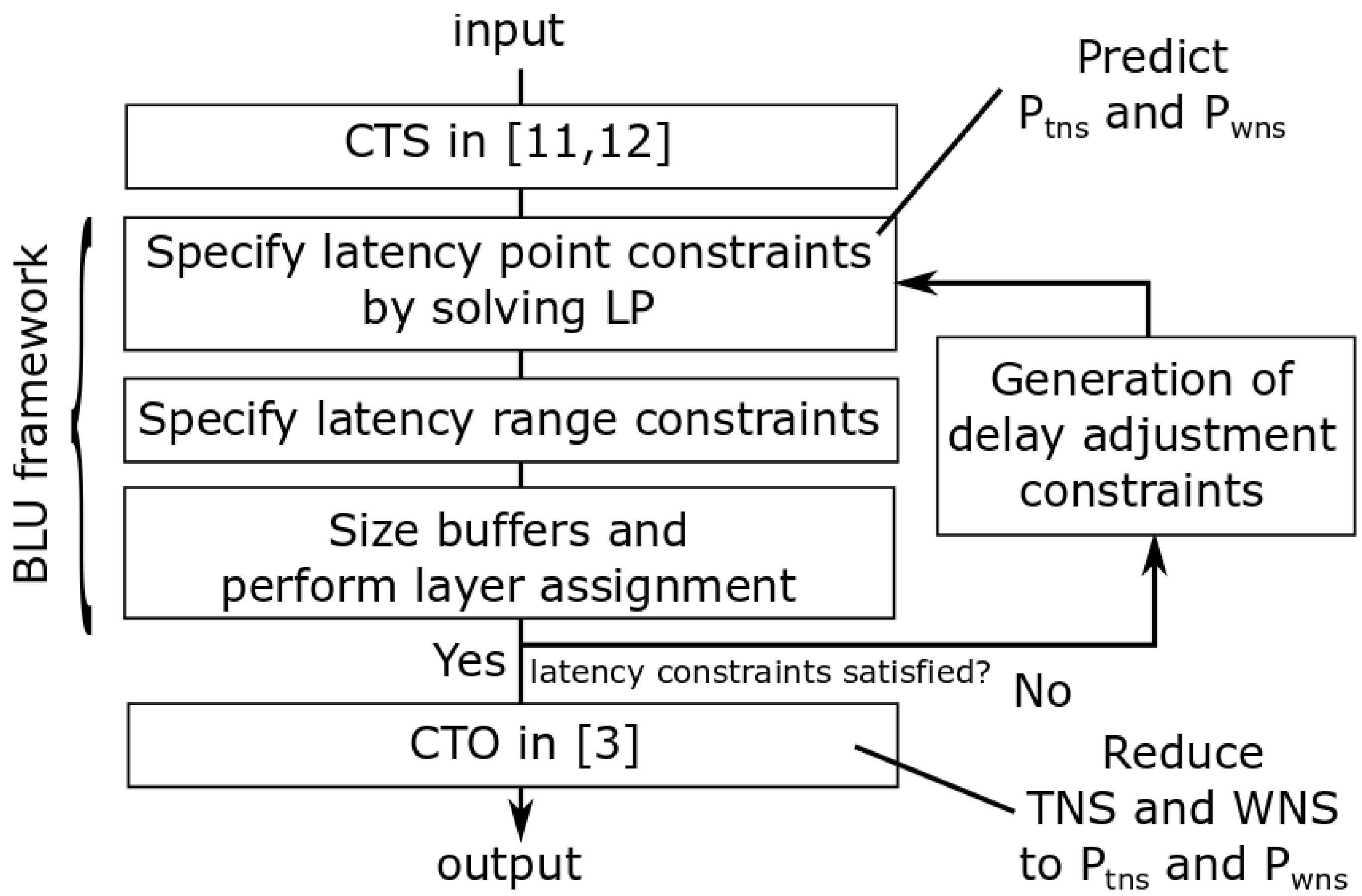
$$\Delta_1 + \Delta_2 + \Delta_4 \geq 0 - 5 - 10 + 4$$

$$\sum_{h \; in \; path(s,k)} \Delta_h \geq L_{sk} + L_{vio}$$

<u>Only realizable negative delay adjustments</u> are specified.

BLU framework introduces additional delay adjustment constraints into LP formulation.

Methodology



[11] Rickard Ewetz and Cheng-Kok Koh. 2018. Scalable Construction of Clock Trees with Useful Skew and High Timing Quality. TCAD (2018).

[12] S. Held et al. 2003. Clock scheduling and clock tree construction for high performance ASICs (ICCAD'03). 232–239.

Experimental Setup

- Benchmarks are synthesized by Synopsis DC & ICC
- CTS Engine in [11] is used to synthesize initial USTs.
- Buffer & Wire Library
- On-chip variations
- Evaluations in timing
  - NGSPICE simulations
- Evaluations in capacitance
  - Total capacitance

Circuit	#Sinks	#Skew Constraints
scaled s1423	74	78
scaled s5378	179	175
scaled s15850	597	318
msp	683	44990
fpu	715	16263
usbf	1765	33438
dma	2092	132834
pci bridge32	3578	141074
ecg	7674	63440
des peft	8808	17152
eht	10544	450762
aes	13216	53382

[11] Rickard Ewetz and Cheng-Kok Koh. 2018. Scalable Construction of Clock Trees with Useful Skew and High Timing Quality. TCAD (2018).

Evaluated Tree Structures

• UST	After CTS (Useful Skew Tree Sythesis)
• UST-CTO	Tree structure after CTO is applied to UST
• UST-P	Tree structure after BLU Framework with point constraints is applied to UST
• UST-P-CTO	Tree structure after CTO is applied to UST-P
• UST-R	Tree structure after BLU Framework with range constraints is applied to UST
• UST-R-CTO	Tree structure after CTO is applied to UST-R
• UST-RT	Tree structure after applying UST-R structure combined with realizing negative delay adjustments.
• UST-RT-CTO	Tree structure after CTO is applied to UST-RT

Results (only non-negative delay adjustments)

Circuit	Cap (pF)						Run-Time (pF)					
(name)	UST	UST-CTO	UST-P	UST-P-CTO	UST-R	UST-R-CTO	UST	UST-CTO	UST-P	UST-P-CTO	UST-R	UST-R-CTO
msp	1.41	1.41	1.35	1.35	1.20	1.20	0.0	0.0	0.5	0.0	3.7	0.1
fpu	1.60	1.60	1.52	1.52	1.35	1.35	0.0	0.2	0.7	0.0	1.2	0.0
usbf	4.55	4.55	4.14	4.14	4.07	4.07	1.0	0.2	0.4	0.2	2.4	0.2
dma	5.06	5.17	4.49	4.65	4.44	4.56	1.0	2.1	1.1	2.5	10.3	2.1
ecg	23.44	23.66	20.39	20.96	20.54	20.84	8.0	11.3	1.7	15.5	5.0	12.8
s15850	18.09	18.85	15.84	16.86	15.77	16.62	0.2	2.3	0.3	14.7	0.2	2.0
Norm.	**0.99**	**1.00**	**0.89**	**0.90**	**0.87**	**0.87**	**0.30**	**1.00**	**0.60**	**1.10**	**1.20**	**1.70**

UST: Useful Skew Tree. UST-P: BLU structure with point constraints. UST-R: BLU structure with range constraints.

Results (with strict timing constraints)

Circuit (name)	Structure (name)	TNS (ps)	WNS (ps)	P_{tns} (ps)	P_{wns} (ps)	Cap (pF)	Run-time (min)
aes	UST	16041	32	7095	14	112.1	24.9
	UST-CTO	8448	18	7950	15	121.6	139.0
	UST-R	36367	47	4478	13	97.8	9.2
	UST-R-CTO	5697	19	5186	15	111.8	73.5
	UST-RT	15685	36	2636	11	102.2	56.5
	UST-RT-CTO	3569	16	3330	14	113.4	189.2
Norm.	UST	3.56	3.04	**1.00**	**1.00**	0.91	0.13
	UST-CTO	**1.00**	**1.00**	4.39	1.12	**1.00**	1.00
	UST-R	6.33	3.84	**0.77**	**0.88**	0.79	0.26
	UST-R-CTO	**0.71**	**1.13**	1.17	1.25	**0.90**	0.95
	UST-RT	4.24	3.19	**0.44**	**0.63**	0.85	0.60
	UST-RT-CTO	**0.42**	**0.80**	0.53	0.73	**0.95**	1.21

UST: Useful Skew Tree. UST-R: BLU structure with range constraints. UST-RT: UST-R & negative delay adjustments.

Summary

- BLU Framework
  - handling discrete buffer sizes
  - layer assignments
  - utilizing useful skew
  - reducing capacitive cost

Questions ?

- Thank you!