



國立交通大學
National Chiao Tung University



國立臺灣大學
National Taiwan University

MAXEDA 至達科技

TAU2019 Timing Contest

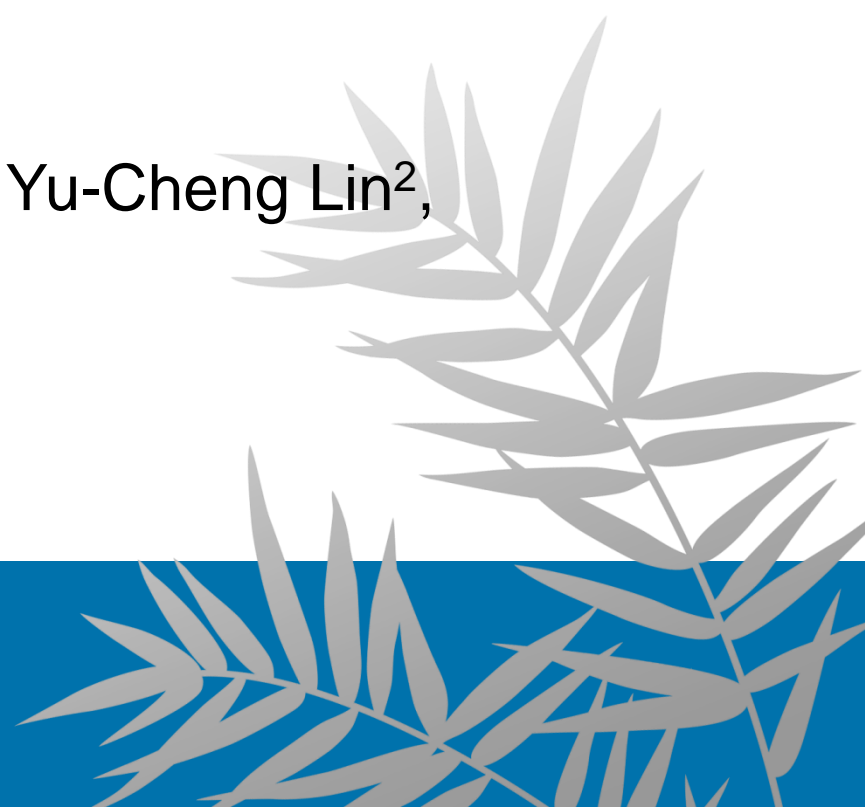
Team: iTimer

Hsien-Han Cheng¹, Tung-Wei Lin², Yu-Cheng Lin²,
Iris Hui-Ru Jiang², Pei-Yu Lee³

¹National Chiao Tung University

²National Taiwan University

³Maxeda Technology



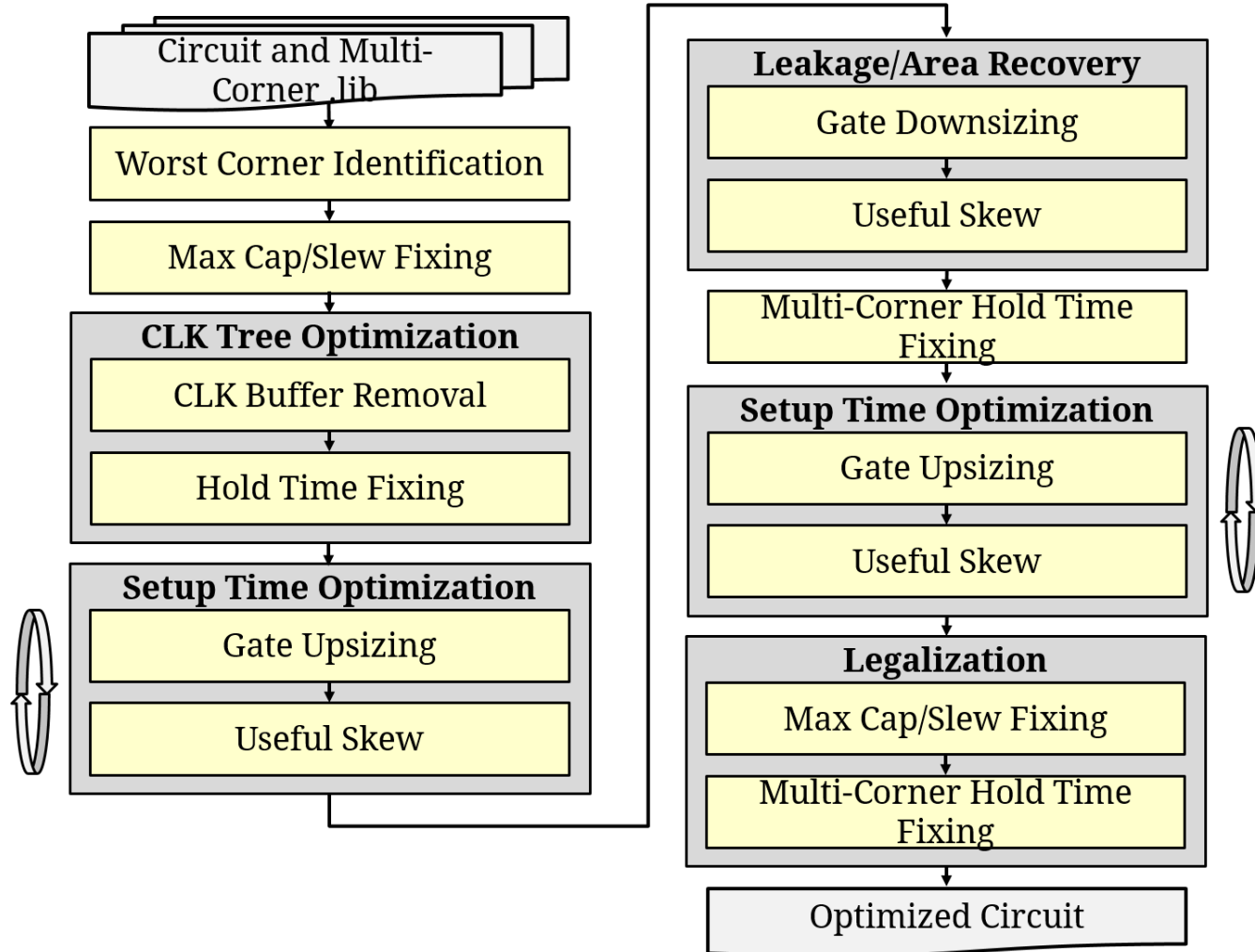
Problem Formulation

- The Design Optimization Problem
 - Given
 - Initial circuit netlist (.v)
 - RC parasitics (.spef)
 - Timing and design constraint file (.sdc)
 - Multiple corner liberties (.lib),
 - Constraints
 - No hold time violations across multiple corners
 - No slew or cap violations across multiple corners
 - Objectives
 - Maximize working frequency
 - Minimize leakage
 - Minimize area
 - Minimize runtime
 - Minimize memory

Challenges

- Gate sizing is NP-hard
- Multi-corner timing optimization is first considered
- Unbalanced clock tree complicates timing optimization

Algorithm Flow



Worst Corner Identification

- The corner which has the slowest cells bounds the highest operating frequency
- The corner with the most total negative slack (TNS) is worst corner
- All subsequent optimization steps focus on timing from worst corner except hold time fixing

Max Cap/Slew Fixing

- Gate upsizing or buffer insertion can solve the violations
- Apply the following procedures sequentially unless the violation is fixed
 - Upsize C
 - Downsize the fanout cell of C
 - Insert buffer after C
 - Insert buffer before the fanout cell of C
- Perform cap/slew violation fixing in BFS order first and then reverse BFS order

Clock Tree Optimization

- CLK Buffer Removal
 - Remove clock buffers as many as possible in this stage
 - Can insert buffers later without inducing too much area overhead
- CLK Buffer Insertion for Hold Time Fixing
 - Fix hold time violations in three ways
 - Clock tree split point buffer insertion
 - Clock tree leaf point buffer insertion
 - Data path buffer insertion

Setup Time Optimization

- Gate Upsizing

- Sensitivities of gates on top k critical paths are recorded

$$sensitivity = \frac{|\sum_e \Delta delay|}{|\Delta area \times \Delta leakage|^{ratio}} \quad (1)$$

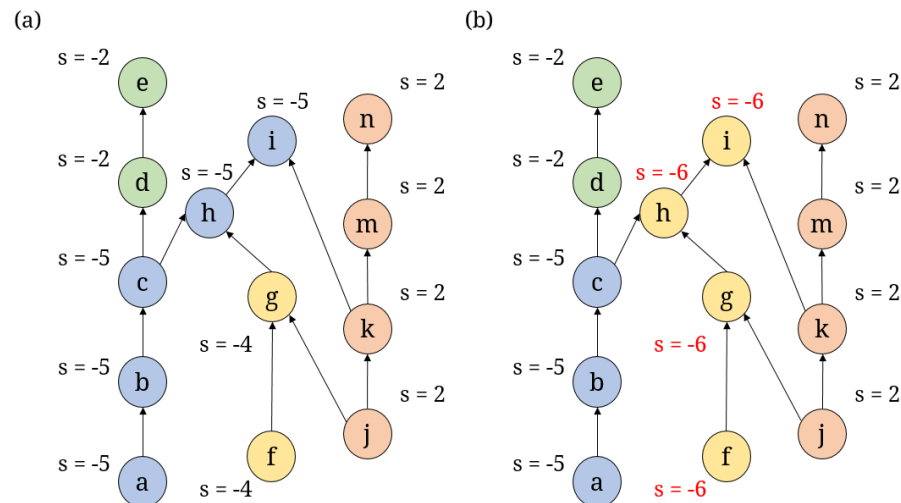
- The top n gates with the highest sensitivities (defined by Equation (1)) are upsized

- Useful Skew

- Is applied on the most critical path
- With attention on positive hold time slacks

Leakage/Area Recovery

- Segment Dependency Graph (SDG) can estimate the propagation of setup slacks after downsizing
- With the global view provided by SDG, we can identify the segments that are less critical and downsize them without harming worst setup slacks



Legalization

- Apply Max Cap/Slew Fixing and Multi-corner Hold Time Fixing
- Multi-corner Hold Time Fixing
 - Iterate all corners
 - Insert buffers only on data path

Experiment Results (1/2)

- Platform: Intel Xeon 2.6GHz Linux Workstation with 197GB memory and 32 CPUs

w.r.t. zero clock period
 $|WNS (Setup)| = \text{longest path delay} = 1/\text{frequency}$

Benchmark			Fast.lib		Typical.lib				
Name	#Cells		WNS (Setup)	WNS (Hold)	WNS (Setup)	WNS (Hold)	Leakage	Area	Runtime
s1196	0.64K	Original	-340.17	0	-561.48	0	3.00×10^4	1.01×10^3	2s
		Optimized	-187.32	0	-305.74	0	1.31×10^4	0.60×10^3	
		Ratio	55.07%	-	54.45%	-	43.67%	59.40%	
systemcdes	3.44K	Original	-1113.61	-248.88	-1598.20	-346.88	1.81×10^5	6.44×10^3	42s
		Optimized	-491.31	0	-727.32	0	0.90×10^5	3.90×10^3	
		Ratio	44.12%	0%	45.51%	0%	49.72%	60.56%	
usb_funct	15.74K	Original	-1288.57	-709.95	-1879.94	-990.72	7.88×10^5	3.02×10^4	106s
		Optimized	-625.98	0	-934.62	0	4.56×10^5	2.09×10^4	
		Ratio	48.58%	0%	49.72%	0%	57.87%	69.21%	
vga_lcd	139.53K	Original	-1943.56	-645.73	-3847.01	-958.87	6.02×10^6	2.48×10^5	36s
		Optimized	-508.38	0	-849.17	0	3.09×10^6	1.46×10^5	
		Ratio	26.16%	0%	22.06%	0%	51.32%	58.87%	
leon3mp_iccad	1247.73K	Original	-2413.56	-1778.74	-3715.46	-2593.74	4.24×10^7	1.83×10^6	423s
		Optimized	-1020.11	0	-1705.49	0	2.37×10^7	1.02×10^6	
		Ratio	42.27%	0%	45.90%	0%	55.90%	55.73%	
Average	281.42K	Ratio	43.24%	0%	43.53%	0%	51.70%	60.75%	121.8s

Experiment Results (2/2)

- usb_function: enormous clock skew
- Clk Tree Opt reduces |WNS (setup)| by 30%
 - Origin goal is to solve hold time violation
 - The harm of an imbalanced clock tree

	WNS (setup)	TNS (setup)	WNS (hold)	TNS (hold)	Leakage	Area	Runtime
Initial	-5.25×10^3	-7.99×10^6	-2.50×10^3	-1.08×10^6	3.02×10^4	7.88×10^6	
After optimization	-2.71×10^3	-6.84×10^4	0	0	2.09×10^4	4.56×10^6	116s
Read libs							4.31%
Max cap/slew fixing	23.24%	15.16%	-0.02%	-8.87%	-23.41%	-35.01%	0.00%
Clk tree opt.	-30.58%	-12.10%	-99.98%	-91.13%	-10.75%	-9.74%	32.76%
Setup time opt.	-40.01%	-16.18%	7.11%	0.03%	1.07%	1.23%	28.45%
Leakage opt.	4.39%	2.14%	0.14%	-0.01%	-0.22%	-0.42%	2.58%
Multi corner hold	3.35%	0.99%	-7.25%	-0.03%	1.67%	0.90%	3.45%
Setup time opt.	-8.74%	-2.93%	0.00%	0.00%	0.75%	0.85%	27.59%
Legalization	0.00%	-1.47%	0.00%	0.00%	0.04%	0.10%	0.86%
Total	-48.35%	-14.39%	-100.00%	-100.00%	-30.85%	-42.09%	100.00%

Conclusion and Future Work

- On average, our flow can decrease worst setup slack by around 56%, leakage by 48% and area by 39%.
- Experiment results show that our proposed algorithm is imperative and can gain notable slack improvement in each stage
- Our future work includes further shortening the runtime and improving the solution quality.

Thank you!

