SYNOPSYS®

Clock Sensitivity Test

Lei Han
Test & Validation Engineer, Synopsys

CONFIDENTIAL INFORMATION

The information contained in this presentation is the confidential and proprietary information of Synopsys. You are not permitted to disseminate or use any of the information provided to you in this presentation outside of Synopsys without prior written authorization.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys' future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to update this presentation or develop the products with the features and functionality discussed in this presentation. Additionally, Synopsys' services and products may only be offered and purchased pursuant to an authorized quote and purchase order or a mutually agreed upon written contract with Synopsys.

EDA Specific Challenges

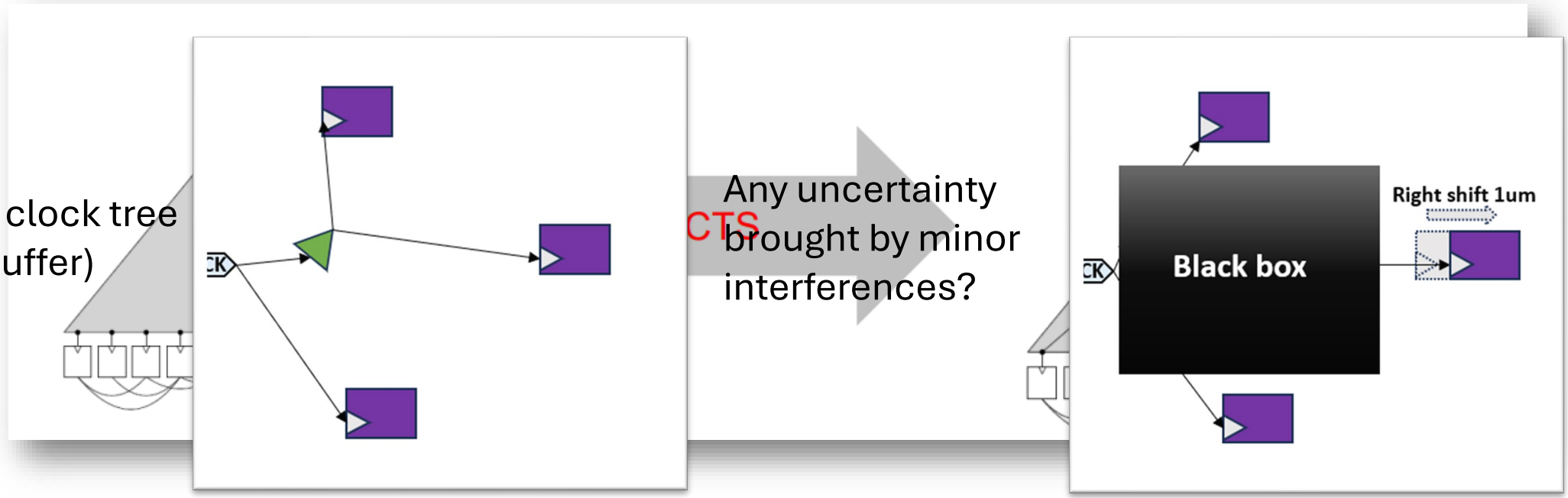
Predicting customer usage is essential, which cannot be accomplished by a single testing scenario.

Customers may encounter **fluctuations in QoR** due to **minor changes** in flow, setting, or design, thereby impacting the release-to-release migration.

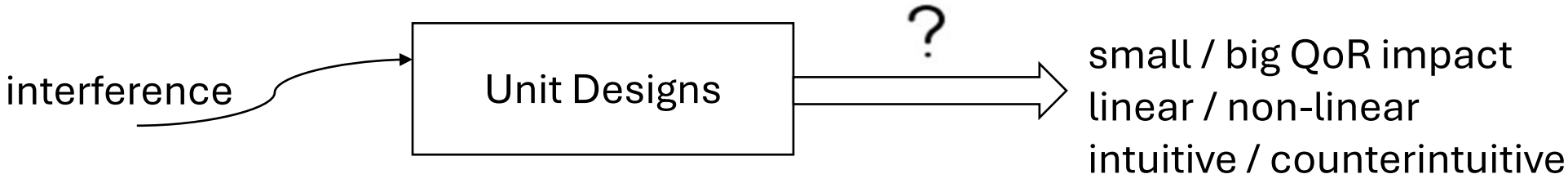
Explore the opportunities from **sensitivity tests** to improve tool quality.

About CTS

Clock tree synthesize + clock tree optimization in the backend flow takes the responsibility to deliver an optimized clock tree solution for **timing-satisfied**, **power/area minimum and latency/skew optimization**.

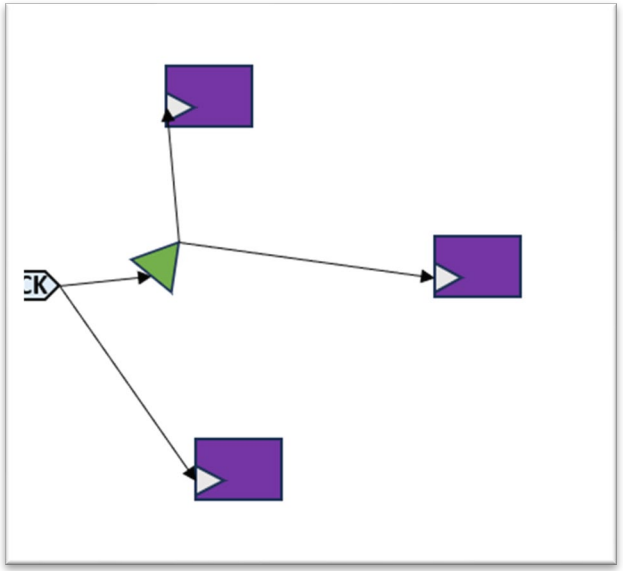


Methodology

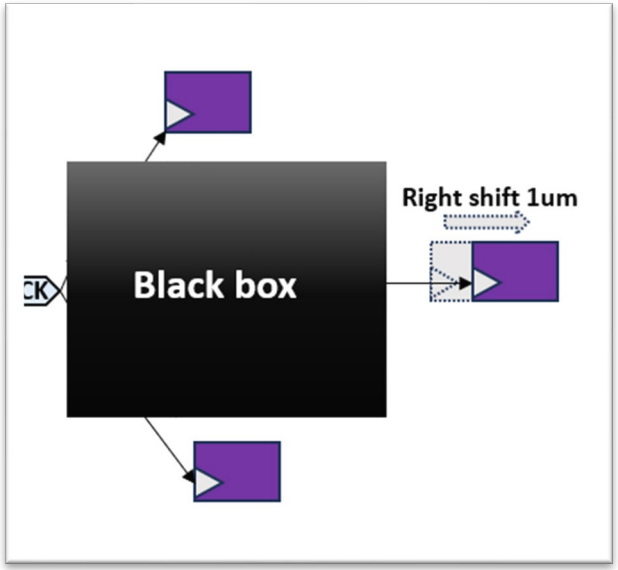


Interference type	Description	Step
Sink location	Shift FF location along x-axis	$[0, \pm1, \pm2, \pm3, \pm5, \pm7, \pm10]$ (um)
Blockage	Add placement blockages	utilization = $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]\%$
	...	

Original clock tree (green buffer)



Any uncertainty brought by minor interferences?



Implementation

Diversity
N7, N5, N3e, N2…
covering various **tech nodes and benchmarks**

Simplicity
Simple enough to guarantee runtime and debuggability

Sophistication
Complex enough to mimic customer clock scenarios

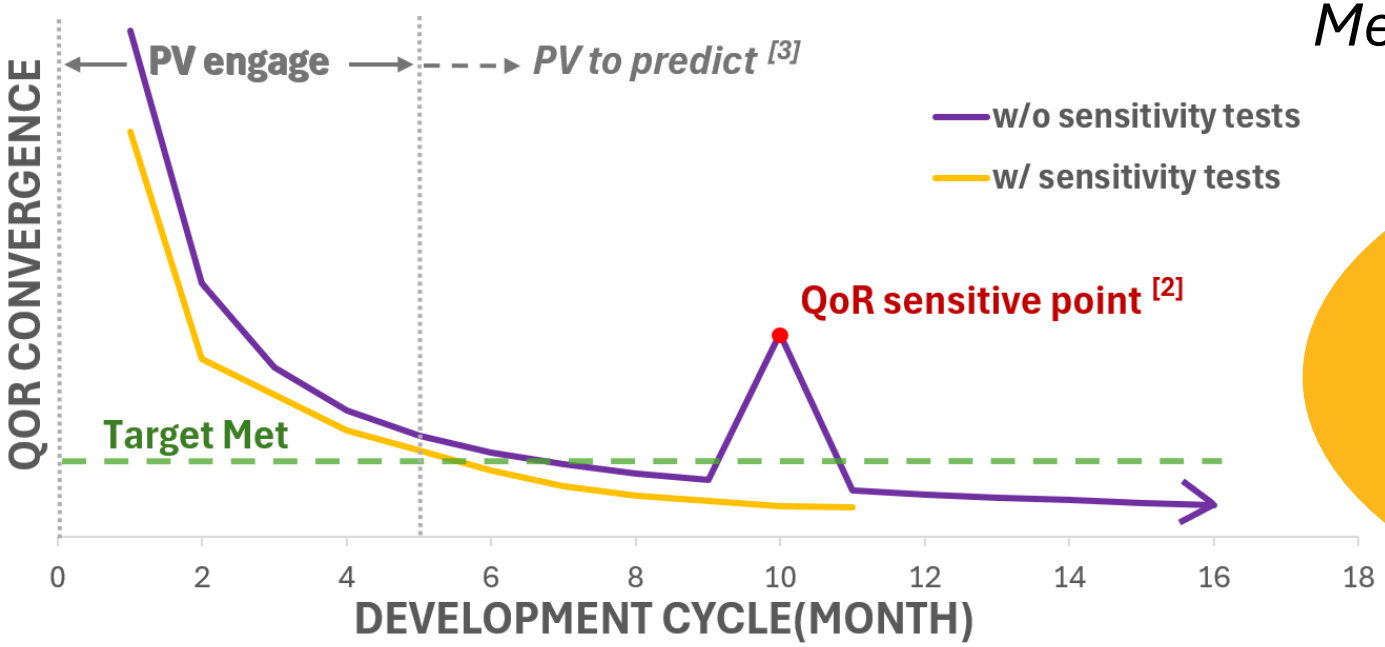
Suite	Design Count	Interferences	Skew Variation Average/Outlier		Latency Variation Average/Outlier	
			Before	After	Before	After
Suite1 (N3E library)	23	Shift FF location	-0.10%/-2.46%	**-0.00%/-0.09%**	-0.36%/-5.28%	**-0.11%/-0.16%**

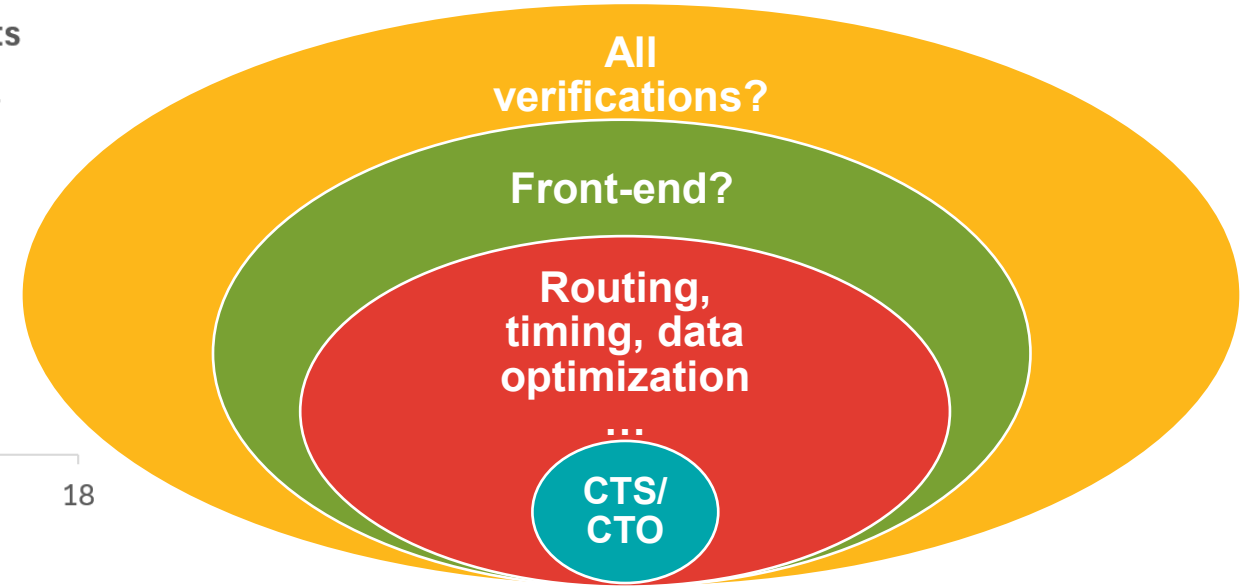
Benefits in Coverage, UX, Quality, Cost

From CTS ... ... to ALL

Scalable Methodology

Methodology can be migrated to any engines





	w/o sensitivity test	w/ sensitivity test
Testing Coverage	Traditional test coverage is **discrete**	Predict **diverse** customer scenarios
User Experience	Minor changes could lead to **large variation**	QoR stays **consistent** facing interferences
Quality	Possibly delivered to customers with **bugs**	Bugs resolved with **priority**
Cost	Debugging is **complicated**	Development cycle is **left-shifted**

Key Takeaways

- Sensitivity tests allows us to explore opportunities to improve tool robustness
  - Reduce variation in engine response to small changes in input stimulus
  - A revolution to break the current “GOLDEN” testing methods

- We start with the CTS engine, and has expanded to other engines

- Welcome your feedback 

SYNOPSYS®

Thank You