Robust Verification for Complex Liberty IP

Santhosh K, Khushboo R, Pramod G (NXP),
Ajay Kumar, Ray Valencia (Siemens EDA)

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA IEEE Council on Electronic Design Automation, sigda acm

Motivation and Main Idea

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA® IEEE Council on Electronic Design Automation sigda acm

Motivation

Foundational IP design of standard cells, memory's, IOs and more are a crucial component for SoC design.

- The foundational IPs in question can have numerous views, including **Liberty (.lib).**

Liberty is the industry standard for representations of timing, power, noise compiled within a cell **library**

NXP provides technology solutions targeting global industries such as automotive, industrial, IoT, mobile, and communication infrastructure.

- The Logic library team works to deliver standard cell libraries for differing NXP design teams for these various targeted applications.

With such a heavy reliance on the standard cell libraries delivered to the greater team, any issues found in the libraries too late in the flow can result in high costs to alleviate.

- **QA (verification) early in NXPs flow is crucial to reduce cost, time, and engineering resources.**

Main Idea

Liberty as a data representation of cells is often a complex and difficult to interpret format.

- Alongside timing, power, and area, liberty also represent aspects of noise, statistical variation, and waveform data for the most advanced processes.

The work presented here elaborates on a solution embedded into NXPs QA flow for robust verification that is easy for all liberty users to use and enables full automated execution resulting in effective engineering and compute resource savings.

More specifically empowering **error detection with AI, analysis, and comparison** of liberty characteristics in advanced technology nodes (sub 10nm) such as:

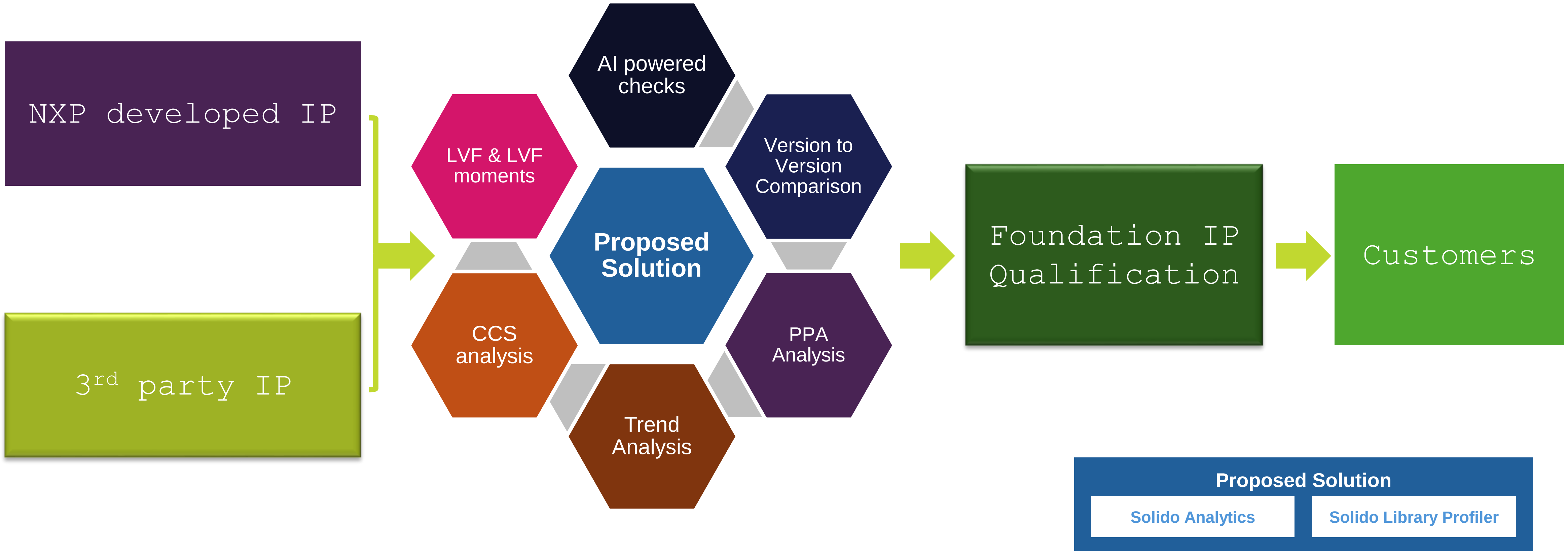
- Liberty Variation Format (LVF) for statistical data.
- Composite current source (CCS) timing, noise, and power waveform data.

QA Flow

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY

NXP’s Comprehensive QA Flow for Foundational IP qualification



Outlier Detection using AI

What is an outlier?

Outliers .libs are data values that don't align with the **neighboring data**.

This neighboring data can be from within a table i.e. slew/load points or across different operating conditions like Process, Voltage, Temperature (PVT) etc..

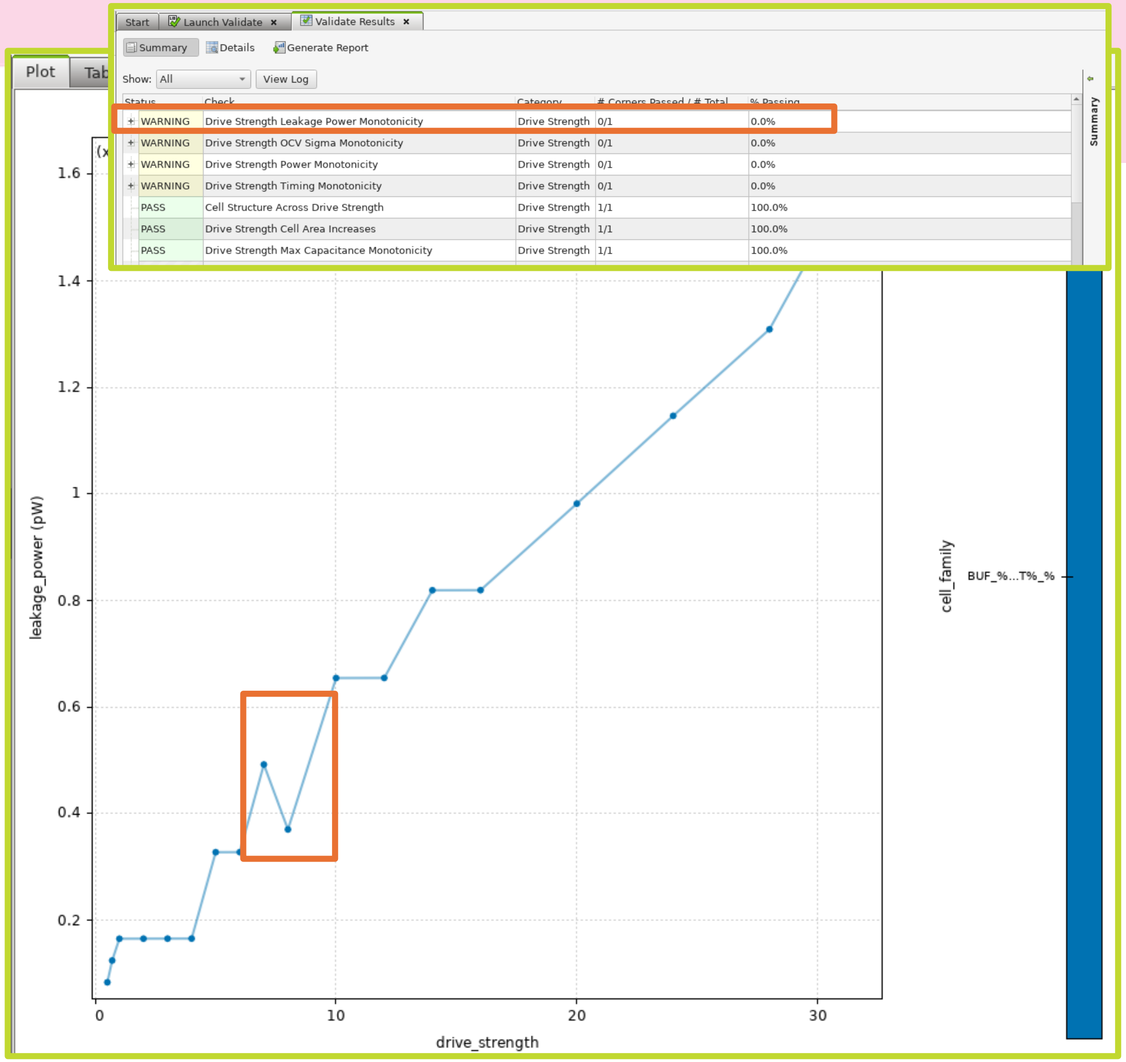
How is it checking?

The methodology uses ML models with **Siemens Solido Analytics** that does a sweep across different Liberty dimensions to detect outliers.

The user can define a tolerance value that when exceeded, would trigger this outlier detection.

Example of dimension that are checked for outliers include:

- Transition/load within a single timing or power table.
- Constrained pin transition within a single constraint table.
- Temperature sweeps across PVTs.
- Voltage sweeps across PVTs.
- Custom numerical sweeps (e.g.: cell drive strength) across PVTs.



Version to Version Comparison

When would we need to do this?

Process Design Kit (PDK) revisions from providers which is very common.

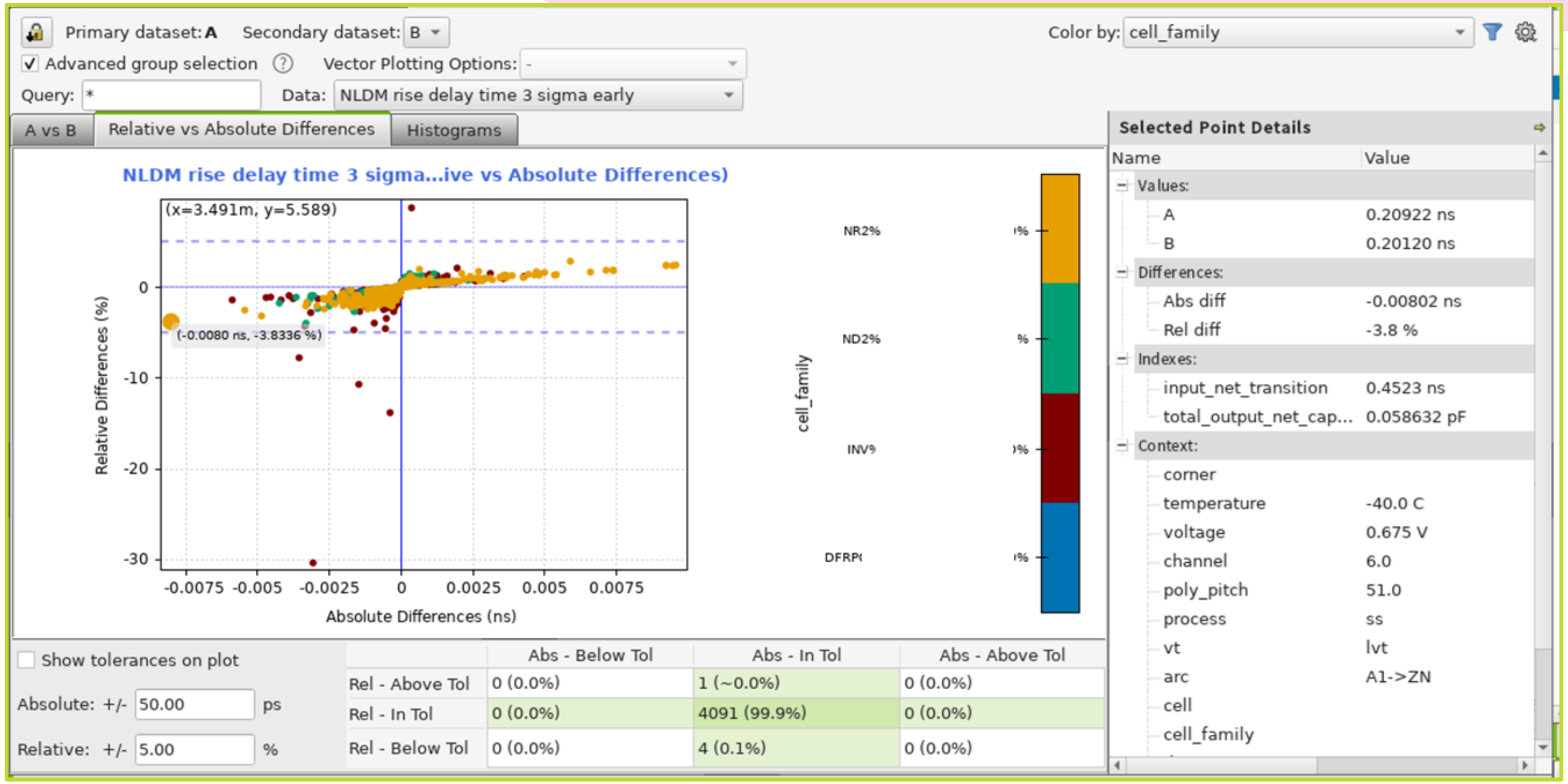
Changes in the design spec that would result in re-characterization of the cells.

Trying to recreate a legacy characterization setup and comparing the .libs for correlation.

Proposed flow:

Leverages plotting that make the comparison process faster

Capable of interpolating the tables between datasets for an apples-to-apples comparison



CCS vs NLDM

Why do we need this?

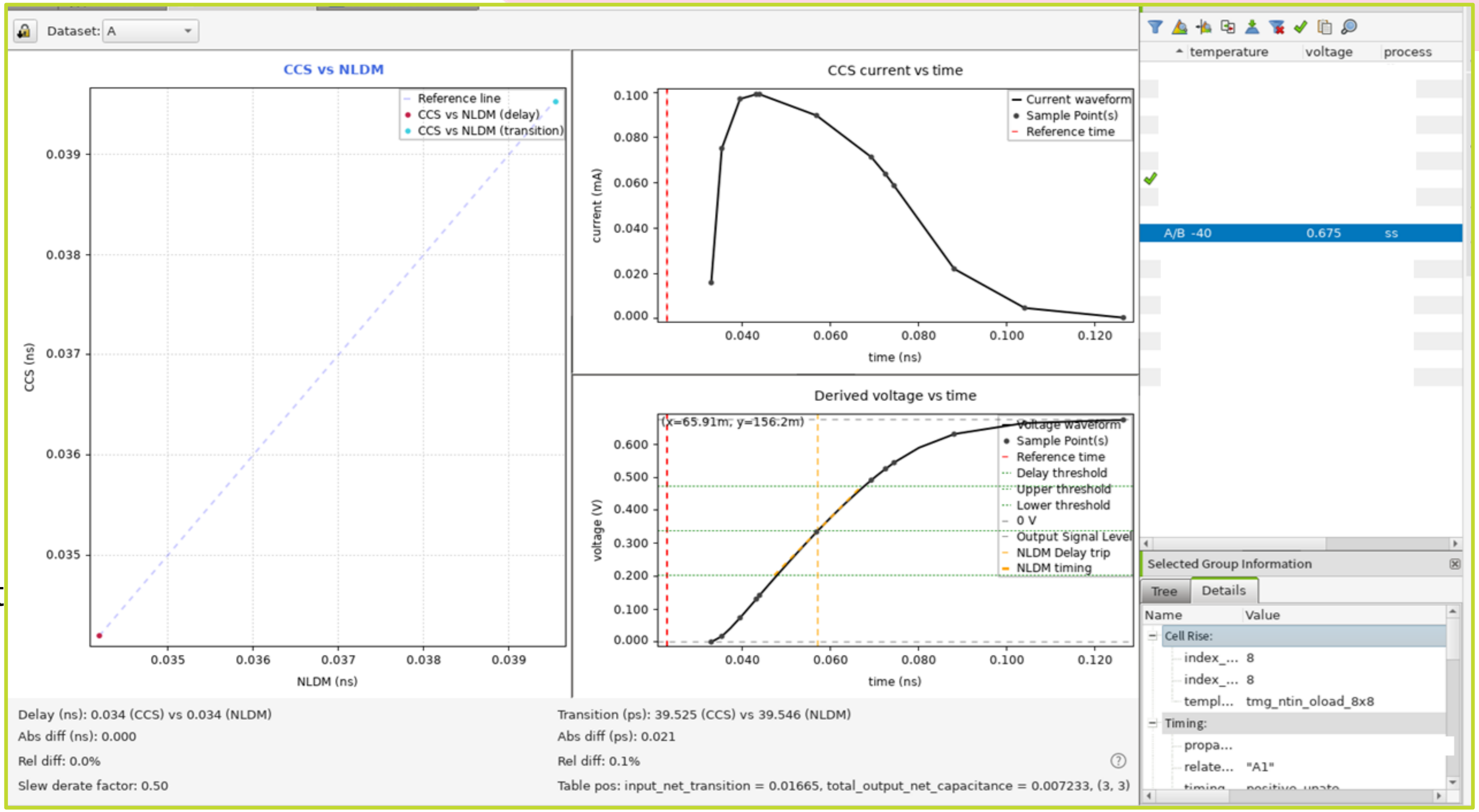
In Liberty format, the effective timing data captured by Composite Current Source Timing (CCST) should be the same as that of Non-Linear Delay Model (NLDM).

Any mismatch of values between CCST and NLDM reflects incorrect characterization settings.

How do we check this?

CCST is capturing the timing data in current domain whereas NLDM captures it in voltage domain.

We need to convert the CCST data from voltage to current domain to compare it with the corresponding NLDM data.



LVF and LVF Moments

Why do we need this?

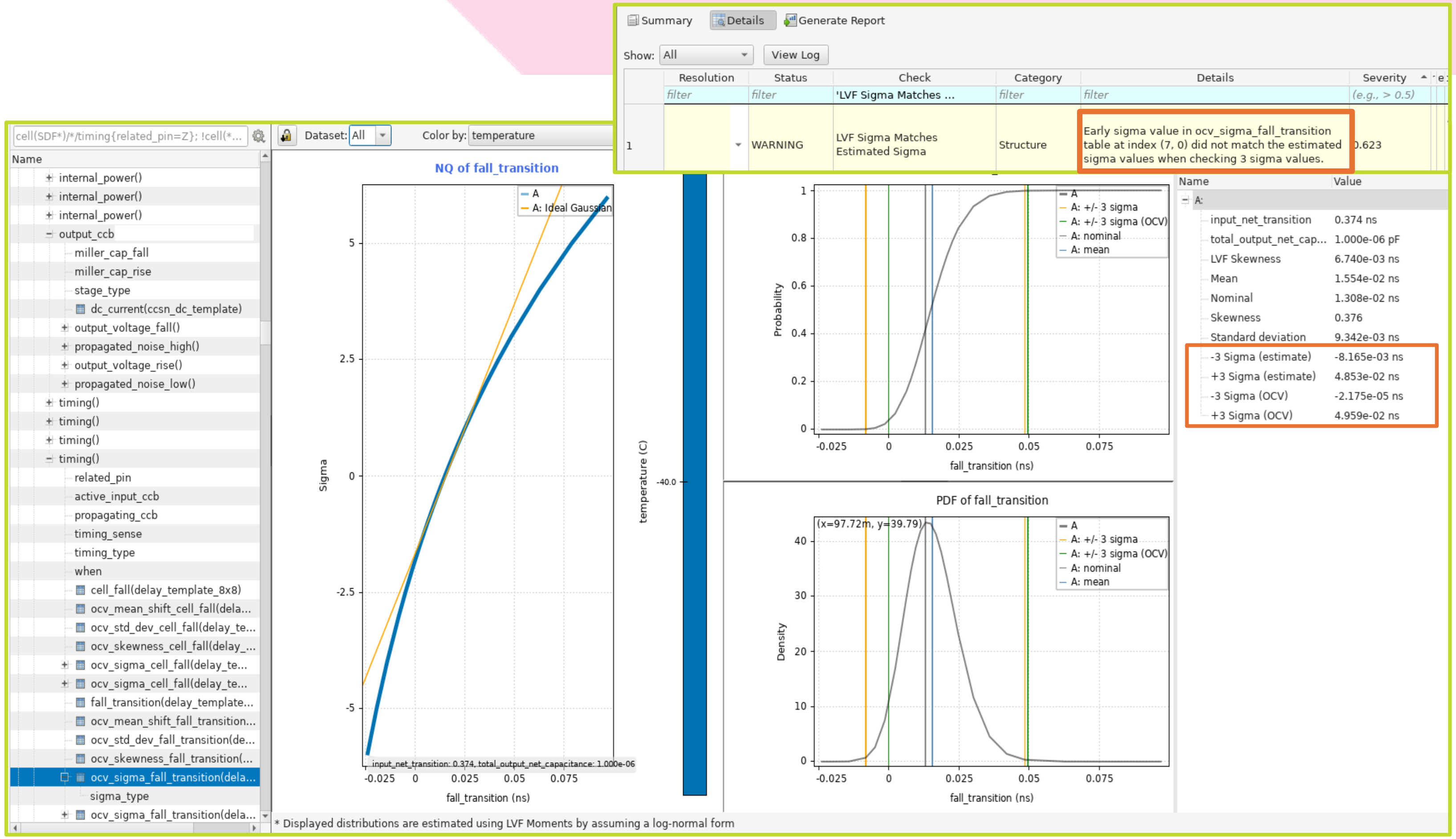
On-chip variation (captured by LVF) is a significant factor in timing sign-off for advanced nodes (<=20nm).

Inaccurate LVF data can lead to deviations in timing by as much as 50-100%.

How do we check this?

Automated checks in our flow that can find outliers in the LVF and LVF Moments groups.

Using LVF Moments, we can recreate the LVF sigma tables. These should match ideally as any mismatch would indicate incorrect characterization settings.



PPA Analysis

Why do we need this?

Currently there are many library providers in the market varying across different technology nodes.

A typical liberty (.lib) can have thousands of unique cells.

Formatting of cells, pins, and data in .libs differ significantly between technologies, sources and variants.

Traditional analysis methods involve multiple iterations of Synthesis, Static Timing Analysis, and Place & Route

Incorrect .lib selection can introduce bottlenecks later in the chip design cycle.

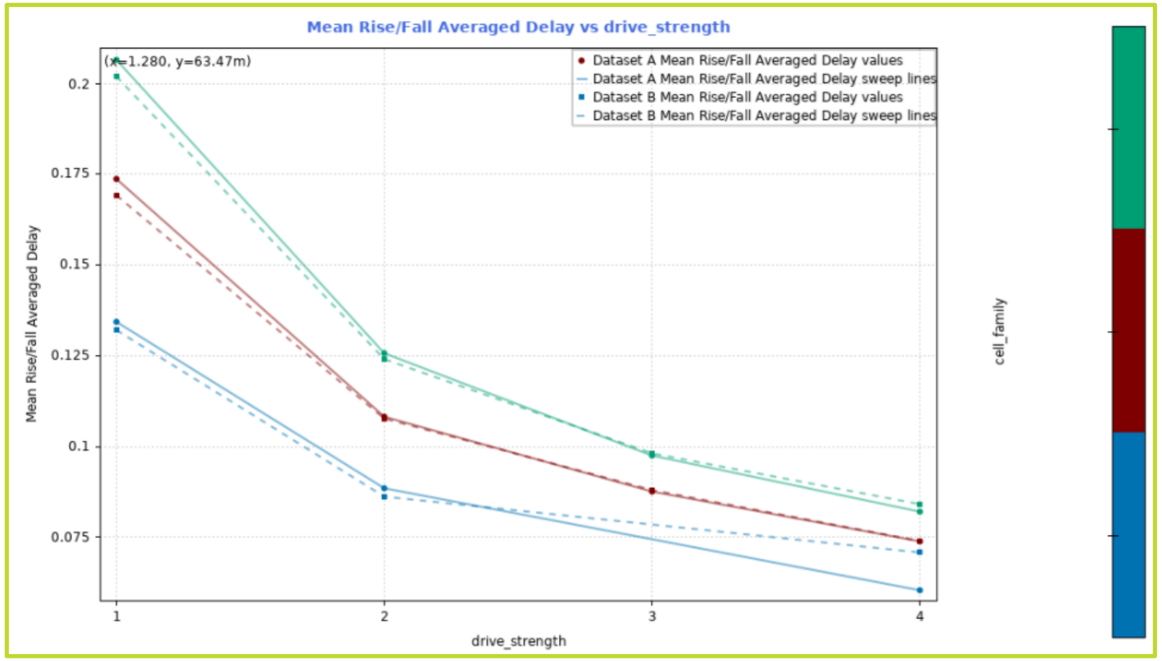
How are we doing the analysis?

Liberty files are available much early in the design cycle and have the required timing and power information for analysis.

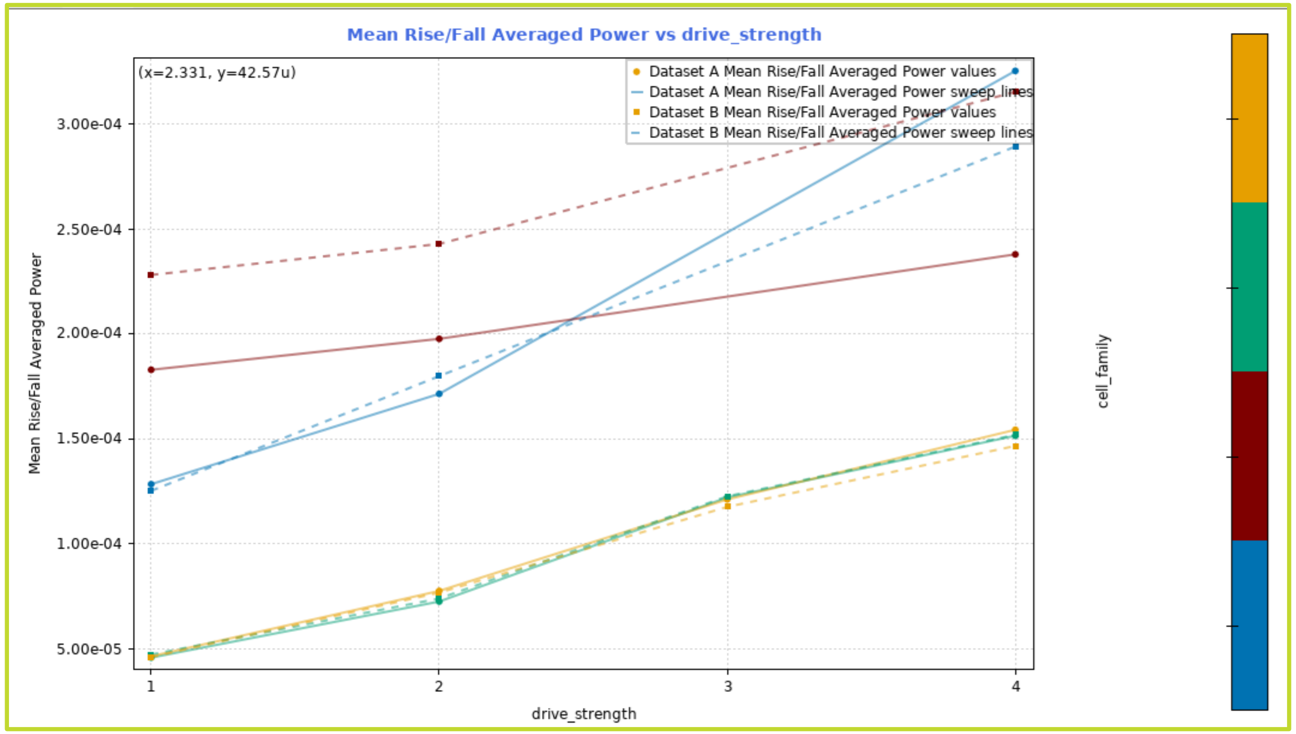
The proposed solution can align the data between different technologies using:

- Cell and pin names, Cell functionalities, Interpolating timing and power tables indices, and more

Examples of PPA analysis



At high drive_strength >3, Dataset A BUFF is performing better than B



Across drive_strength, Dataset A DFF is better in terms of power consumption than B

Summary

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA IEEE Council on Electronic Design Automation sigda acm

Summary

✓ **Through this work, NXP was able to improve efficiency of their QA flow at a rate of 2x**

Sr. No	Use case	No. of cells per .lib	Total no. of PVTs	Runtime (hrs)		Speedup
				Traditional WoW	Proposed Solution	
1	Trends analysis & Validate checks	1000	25	10 units	5 units	**2x**
2	Impact analysis	1000	6	5 units	3 units	**1.6x**
3	PPA Comparison	1000	4	5 units	3 units	**1.6x**

✓ **The embedded solution within NXPs flow enabled the following**

Impact analysis across PDK revisions

Automated execution with minimal setup

Easy validation of LVF moments data

Early detection of outliers within LVF liberty data

Full coverage for QA of liberty checks

Deep understanding of CCS data with respect to NLDM