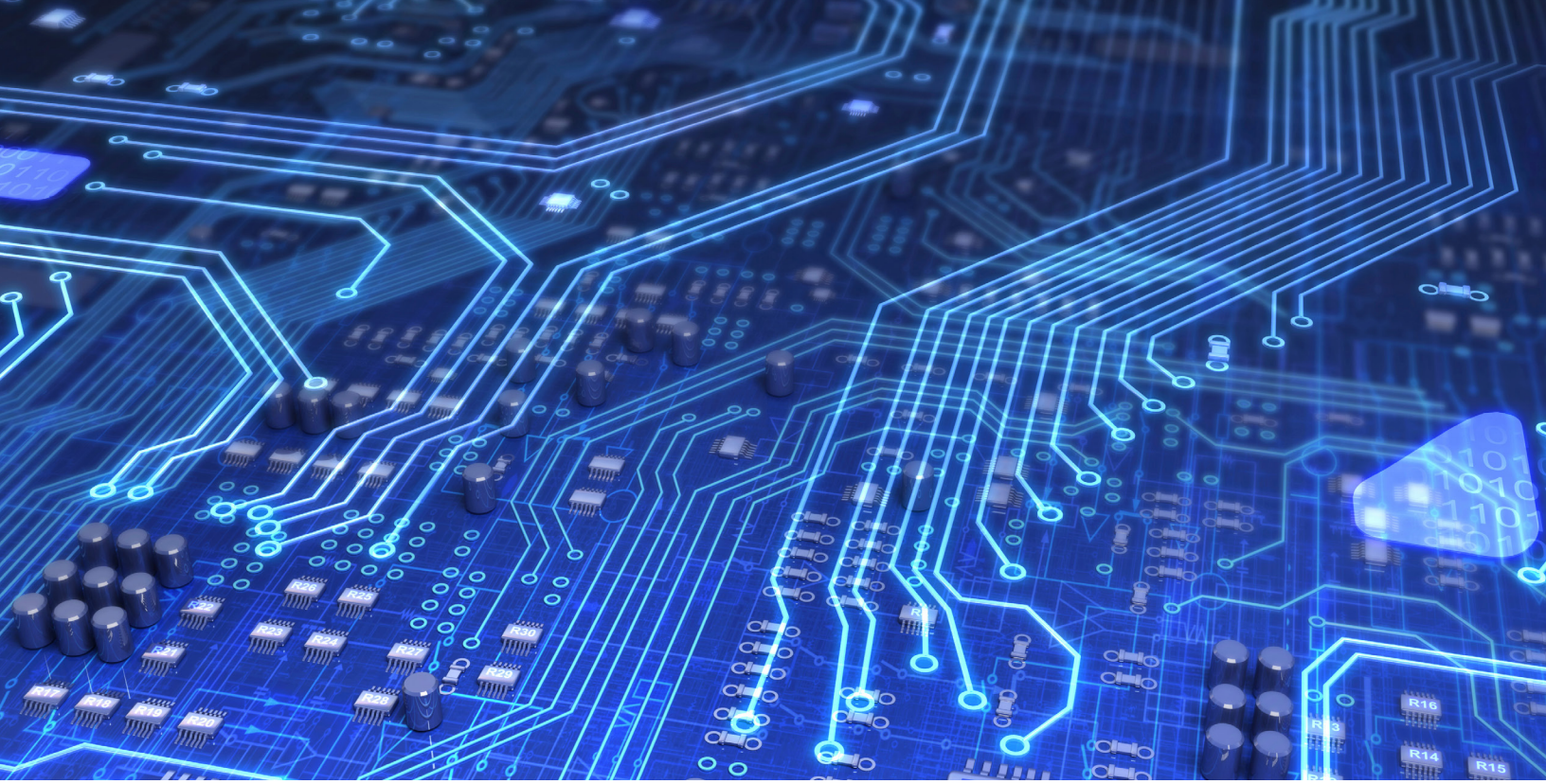


DIGITAL INDUSTRIES SOFTWARE

Addressing library characterization and verification challenges using ML

Executive summary
At advanced process nodes, Liberty or library (.lib) requirements are more demanding due to design complexities, increased number of corners required for timing signoff, and the need for statistical variation modeling. This results in an increase in size, complexity, and the number of .lib characterizations. Validation and verification of these complex and large .lib files is a challenging task and poses a significant threat to successful timing closure and even silicon failures if the .lib errors are not detected and fixed in time. This white paper describes the use of machine learning (ML) techniques in the Siemens EDA Solido™ Characterization Suite that accelerates production quality .lib characterization and verification at advanced technology nodes. These ML techniques address some of the fundamental challenges with the demanding .lib requirements of modern technology nodes and their validation.

Tina Durgia, Product Management, Siemens EDA

SIEMENS

Contents

Introduction

The traditional corner-based signoff approach is still commonly used for tape-outs today. At higher technology nodes such as 180 nanometers (nm), the process of design closure was less complex and mostly involved closing timing at the worst-case and best-case conditions. However, as technologies shrank variability increased with reduced supply voltages, and systematic variations related to location and patterns came to the forefront. Designers also started observing temperature inversion effects (where the delay increases at lower temperatures), which caused changes in the electrical characteristics of the devices, leading to variability in circuit performance.

Libraries are the backbone of the digital design flow and are used at every step of the flow that involves static timing analysis (STA) (figure 1). In the early parts of the flow, namely logic synthesis, formal verification and implementation or place-and-route stages, library requirements are not as high as only a handful of critical corners are used. However, library requirements become more aggressive at the signoff timing stage, where the number of corners can range anywhere from 70 to 200 at smaller nodes. Foundries typically provide a few combinations of the PVT .libs (e.g., 25 to 30 corners for 7nm), but the rest need to be either characterized by library teams or requested from the foundry.

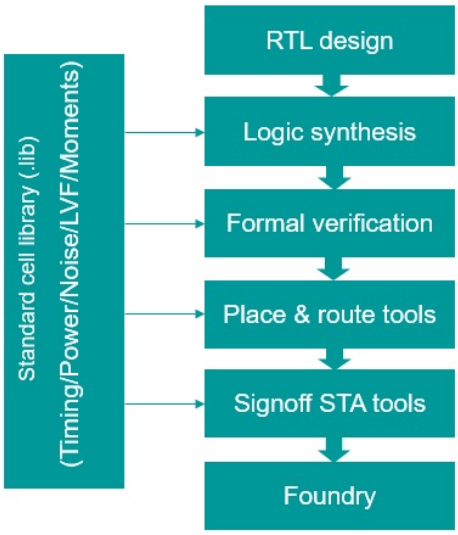


Figure 1: Standard cell library usage in digital design flow.

Traditional methods used to characterize and validate Liberty files have remained largely unchanged. However, with shrinking technology nodes, design complexities, and the need to model the advanced node effects to account for variations in circuit performance, the number of SPICE simulations required for characterization of these Liberty files has increased significantly, resulting in characterization runtimes exceeding production schedules. To reduce the characterization runtimes, sensitivity-based approximations and netlist reduction techniques are employed by some characterization tools. These methods can impact the accuracy of the Liberty models, necessitating a thorough library validation process.

Validation of complex .libs is an equally difficult undertaking. Traditional .lib validation tools that provide static, rule-based checks can verify the syntax of the .libs, but given the complexity and large volume of characterized data at the advanced technology nodes, these checks are no longer sufficient. There is a need for a sophisticated .lib verification system that not only performs syntactical checks but also detects any data outliers that could potentially cause failures in the later stages of the flow.

This white paper describes innovative and production-proven ML techniques utilized to significantly speed up library characterization and validation runtime to achieve better quality libraries, shorter and predictable production schedules and improved power, performance, area (PPA) and yield for silicon designs. These ML techniques can also bridge the gap that exists today between what design engineers require for signoff and what library teams can deliver.

Challenges with advanced nodes

As previously discussed, the .lib requirements at smaller process nodes, namely 16nm and smaller, are rigorous and pose severe characterization challenges as described below:

Drastic increase in the number of signoff corners

One of the biggest challenges of smaller process nodes is the increase in the number of corners required for timing signoff. Several factors including (a) process variations due to variations in the oxide thickness, dopant concentrations and diffusion depths, (b) temperature variations due to power dissipation in the devices, and (c) variations in voltage due to IR drop and voltage supply noise due to parasitic inductance result in a large increase in the number of signoff corners for STA. Figure 2 shows the steep rise in the number of PVT corners used for signoff at smaller process nodes. A higher number of signoff PVT corners used for timing analysis is generally required to prevent timing-related chip failures. However, more signoff PVT corners come with a cost. It generally means a greater number of libraries need to be generated for these corners, resulting in a significant amount of characterization runtime.

Most timing analysis tools support voltage and temperature scaling and use an existing set of PVT libraries characterized at different temperature and voltage conditions for interpolation. Although this technique has been used for many years at more mature technology nodes, it does not provide the required accuracy at smaller process nodes. Therefore, additional characterization of PVT .libs at each of these additional timing signoff corners is required for correct timing closure.

On-chip variation modeling

At smaller process nodes, local variations are a big challenge that chip designers must deal with. These are variations in electrical characteristics of the transistors. For more mature nodes, on-chip variation (OCV) effects are small compared to nominal timing, therefore flat or stage-based derates can be used. However, for advanced process nodes, OCV effects can exceed the nominal timing and therefore Liberty Variation Format (LVF) characterization is a requirement. LVF (without moments) is an extension to the Liberty format and adds statistical variation information in the form of mean and standard deviation to the nominal delay and constraint values that are sensitive to both input slew and output load, thus providing the most granularity and accuracy at smaller process nodes where the distribution is Gaussian or normal as shown in figure 3. During STA, LVF 3-sigma values are used by timing tools to add pessimism to the timing paths, because the 3-sigma interval from the mean or nominal delay covers roughly 99.7 percent of the variation and is adequate for modeling. With a brute-force approach to generate sigma values for every arc of every cell in the .lib, several thousand Monte Carlo simulations need to be run, which can cause characterization runtime to increase by approximately 1,000 times.

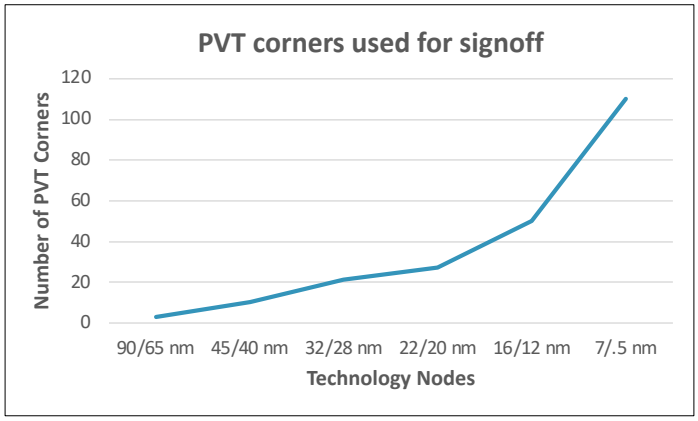


Figure 2: Sharp increase in the number of PVT corners with technology nodes.

At ultra-low voltages typically used for process nodes below 10 nm, timing variation is no longer Gaussian, and the spread is skewed either in the positive or negative direction as shown in figure 4. To account for these shifts from the mean values, LVF syntax now supports moments. LVF moments contain additional statistical parameters including standard deviation, mean shift (from nominal values), and skewness, which enable more accurate modeling of the statistical distribution for STA. Moments-based LVF exacerbates the library characterization runtime further due to additional statistical distribution information that needs to be characterized, something traditional SPICE characterization tools can't easily scale to.

At smaller process nodes, LVF characterization runtime dominates the overall library characterization turnaround time. To speed-up the characterization process and achieve faster time-to-market, it is imperative that the flow is streamlined and expedited.

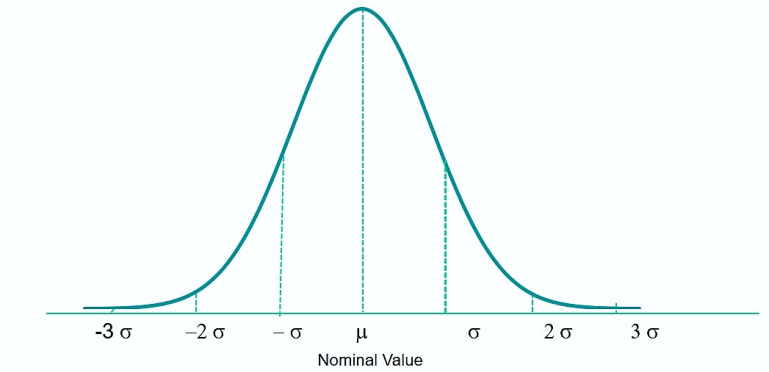


Figure 3: A .lib with LVF information includes mean and standard deviation data for Gaussian distributions.

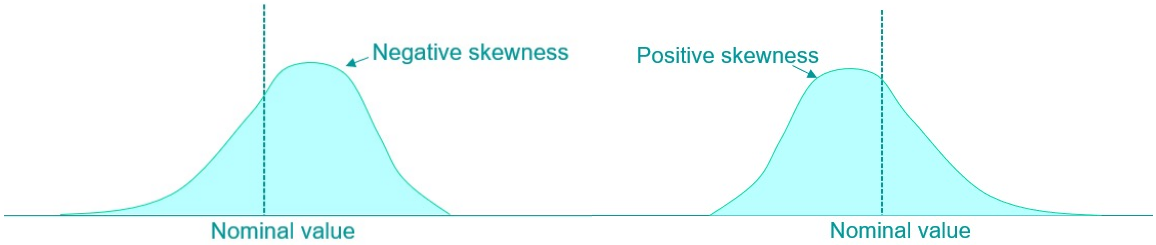


Figure 4: LVF .libs with moments include skewness for every timing arc.

Power management techniques and cell complexity

Power management methods in silicon designs are relatively new compared to the overall silicon design process. Today, the miniaturization of transistors in modern technology nodes has helped reduce the form factor of SoC/IC designs. However, the combination of having a small form factor with many active devices on the chip results in an increased risk of overheating. Therefore, several power management techniques are employed today to reduce power including power gating, static voltage scaling, dynamic voltage and frequency scaling (DVFS) and adaptive voltage scaling (AVS). All these techniques have a common approach to tackle the issue: partition the design into voltage regions or power domains, and have each region operate at its own power supply.

In addition, these multi-supply voltage designs need level-shifters for signals that cross voltage domains on the chip. Although these techniques help overcome

power issues, they also result in challenges for library characterization and signoff teams.

For a chip with a single supply voltage, libraries only need to be characterized for that operating voltage.

However, when the design is partitioned into multiple blocks (each one operating at a different voltage) libraries need to be characterized at all of these operating voltages.

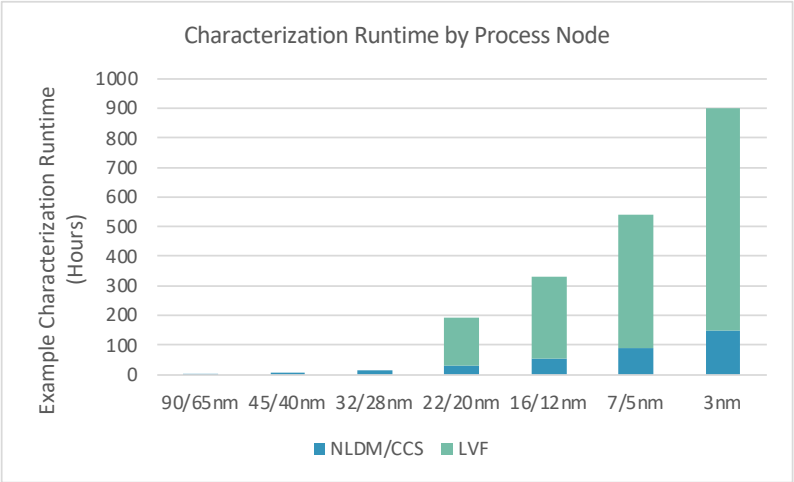


Figure 5: Characterization runtime is dominated by LVF data at smaller nodes.

Complex characterization environment setup

Advanced process nodes introduce a new set of challenges for characterization teams. One of the most common problems is outliers in the data tables that typically occur due to mismatches in characterization settings or differences in the environment used to characterize the PVT corner. Outliers often go undetected with the traditional validation flows that include static rule checks.

Figure 6 shows a rise constraint for a cell's timing arc with respect to voltage, and the highlighted section shows a problem. The rise constraint at 0.66v consistently has a systemic offset compared to the rise constraint at other voltages. Problems like these are hard to detect when analyzing a library in isolation but become apparent when multiple libraries are visually compared at the same time.

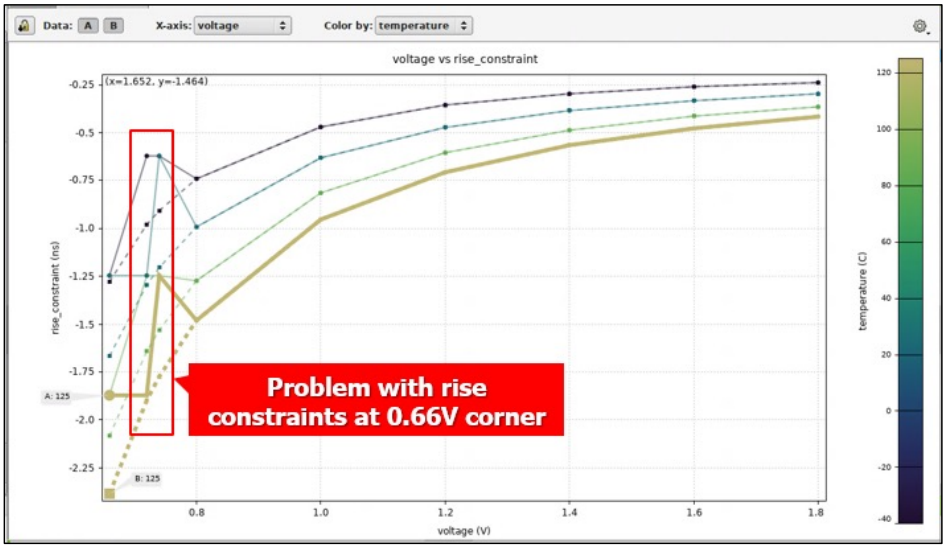


Figure 6: Example of outliers in .libs that are hard to detect with static-rule checks.

Validation of complex .lib data structures and cell types

Liberty syntax has inflated considerably over the years due to continuous introduction of new data structures to handle new STA issues like signal integrity, noise, OCV, and other factors. Because of this, detecting outliers and other issues in .libs has become more challenging. For example, characterization process issues like incorrect partial table data for specific slew-load indices as shown in figure 7 are hard to detect using scripts or rules and may lead to incorrect STA results.

Furthermore, today's IP libraries include a large set of specialized and complex cells, in addition to the baseline set of standard cells and I/O cells. These cells are represented both by nonlinear delay models (NLDM) and the more widely used and accurate composite current source (CCS) models. LVF also is a critical component of .libs today, making thorough verification and repair of any inaccuracies in the Liberty models vital and not to be ignored; otherwise, the risk of tape-out delays or chip failure increases significantly. In many validation flows today, human intervention and custom work is required to detect LVF errors (figure 8) and validate the accuracy of .libs. Unfortunately, this is time-consuming and error-prone.

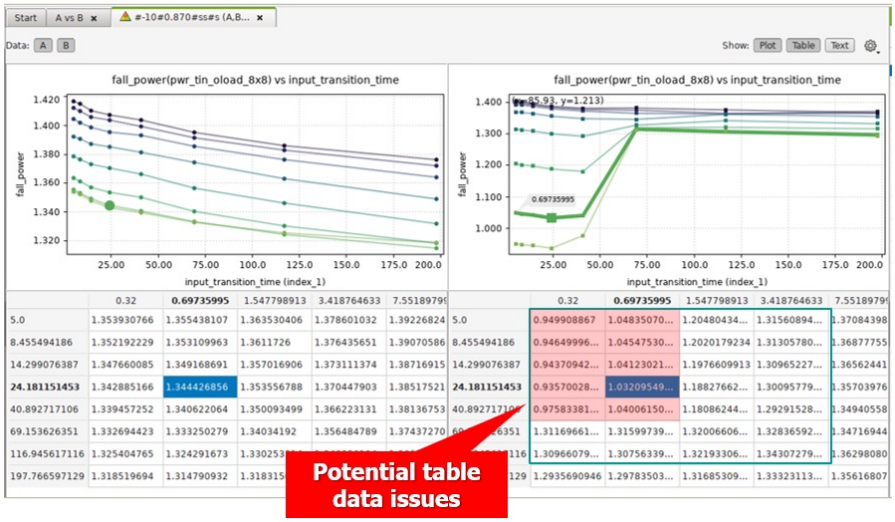


Figure 7: Partially incorrect table data due to characterization problems.

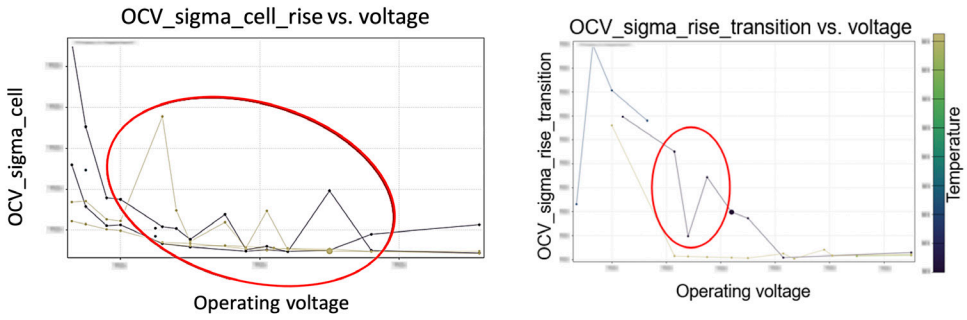


Figure 8: Some common issues encountered with LVF characterized data.

ML-enabled .lib production and verification with Solido Generator and Solido Analytics

The Solido Characterization Suite uses production-proven ML techniques to accelerate library characterization and verification of standard cells, memory and custom blocks. The two main components of the suite are Solido Generator and Solido Analytics.

Solido Generator uses ML methods to accelerate the overall library characterization process by instantly generating libraries for additional PVT corners after the initial characterization. Solido Generator uses existing SPICE-characterized libraries as anchor data to build ML models of the libraries and produce new PVT libraries.

Prior to generating the additional PVTs, Solido Generator analyzes the anchor corner set to determine the optimized set of libraries needed for additional PVT generation. Since the tool uses a set of pre-characterized .libs, it eliminates the dependency on SPICE netlists or subcircuits and the need to replicate characterization settings to match that of the library vendor. Solido Generator runs about 100 times faster than traditional SPICE.

The ML-enabled methods in Solido Generator give users the "best of both worlds" by generating production-accurate LVF .libs for additional PVT corners in a fraction of runtime compared to brute-force Monte Carlo or approximated Monte Carlo methods, while retaining accuracy equivalent to its input anchor .libs.

Solido Analytics is an advanced library validation, analysis, and debugging solution that includes not only fast, parallelized, and comprehensive static rule-based checks, but also employs an ML outlier detection tool that "learns" the expected characterized values in a library and automatically detects errors like outliers or non-monotonic behaviors in the characterized data that typically go undetected with other tools.

Custom plots in Solido Analytics

Solido Analytics comes with a library visualization graphical interface that helps users visualize the problems and explore and debug .libs quickly and efficiently, saving engineering time spent on a library-to-design timing closure debug that would otherwise be needed.

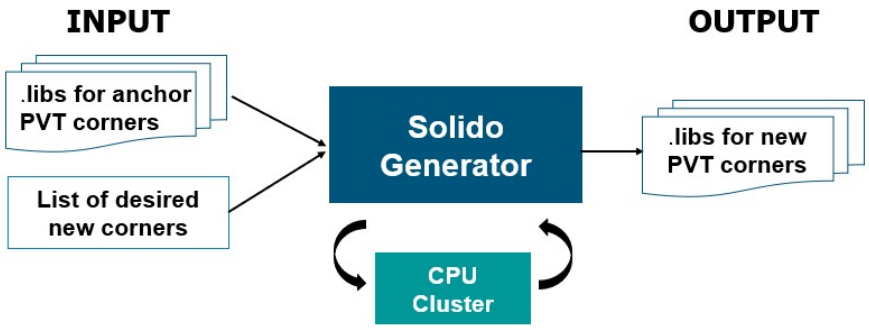
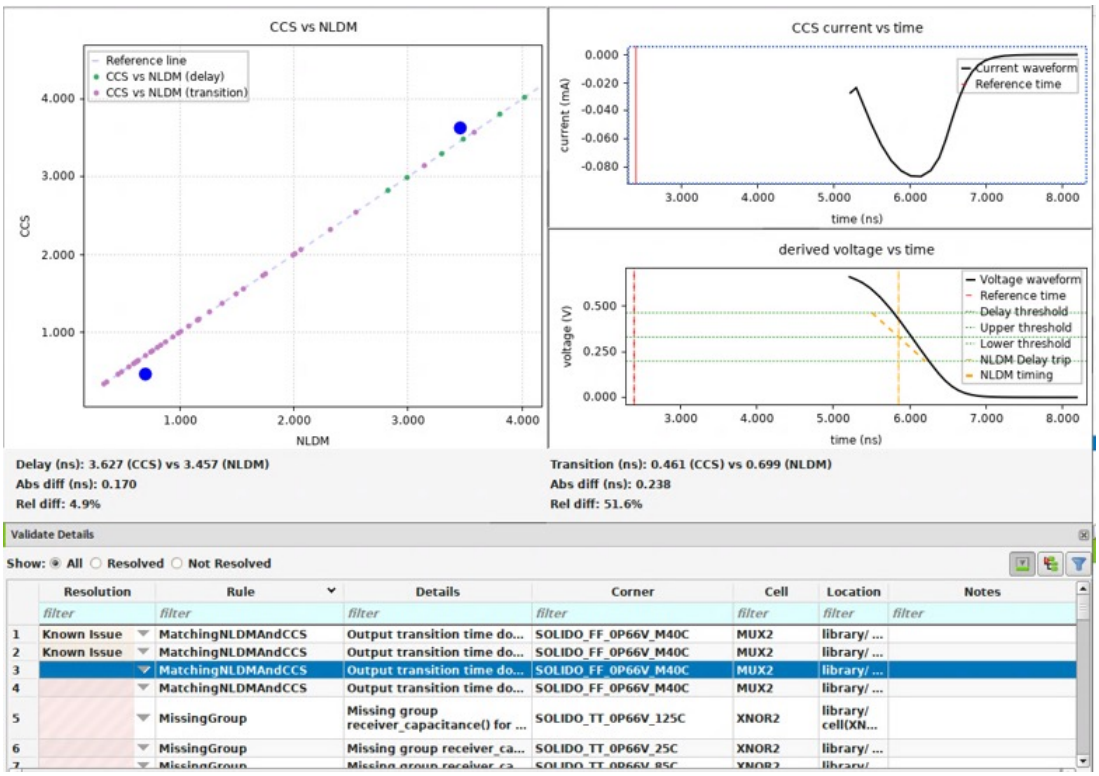


Figure 9: Additional PVT generation with Solido Generator.

Since .libs are essential building blocks of all SoC/ASIC designs, a critical step to achieving optimal PPA for a design is to ensure these building blocks are robust and reliable. For example, one of the commonly encountered issues during .lib verification are differences between NLDM and CCS timing values, causing correlation issues between pre-route and signoff timing. Solido Analytics helps detect these issues and provides plots to help users visualize the problem and trace back to the source of the issue, as shown in figure 10 below.



	Resolution	Rule	Details	Corner	Cell	Location	Notes
	filter	filter	filter	filter	filter	filter	filter
1	Known Issue	MatchingNLDMAndCCS	Output transition time do...	SOLIDO_FF_0P66V_M40C	MUX2	library/ ...	
2	Known Issue	MatchingNLDMAndCCS	Output transition time do...	SOLIDO_FF_0P66V_M40C	MUX2	library/ ...	
3		MatchingNLDMAndCCS	Output transition time do...	SOLIDO_FF_0P66V_M40C	MUX2	library/ ...	
4		MatchingNLDMAndCCS	Output transition time do...	SOLIDO_FF_0P66V_M40C	MUX2	library/ ...	
5		MissingGroup	Missing group receiver_capacitance() for ...	SOLIDO_TT_0P66V_125C	XNOR2	library/ cell(XN...	
6		MissingGroup	Missing group receiver_ca...	SOLIDO_TT_0P66V_25C	XNOR2	library/ ...	

Figure 10: The Information Visualization GUI in Solido Analytics helps visualize errors and simplify debug.

Data visualization makes it easier to condense large amounts of information in the .libs to identify patterns and relationships and gain new insights about the .libs models. Therefore, in addition to the many built-in plots available, Solido Analytics also provides the flexibility to generate custom plots with user-defined functions, combining variables in .libs as dimensions for graphs to help analyze .libs from different viewpoints. For example, figure 11 shows the delay of a cell across a combination of slew and load values colored by voltage.

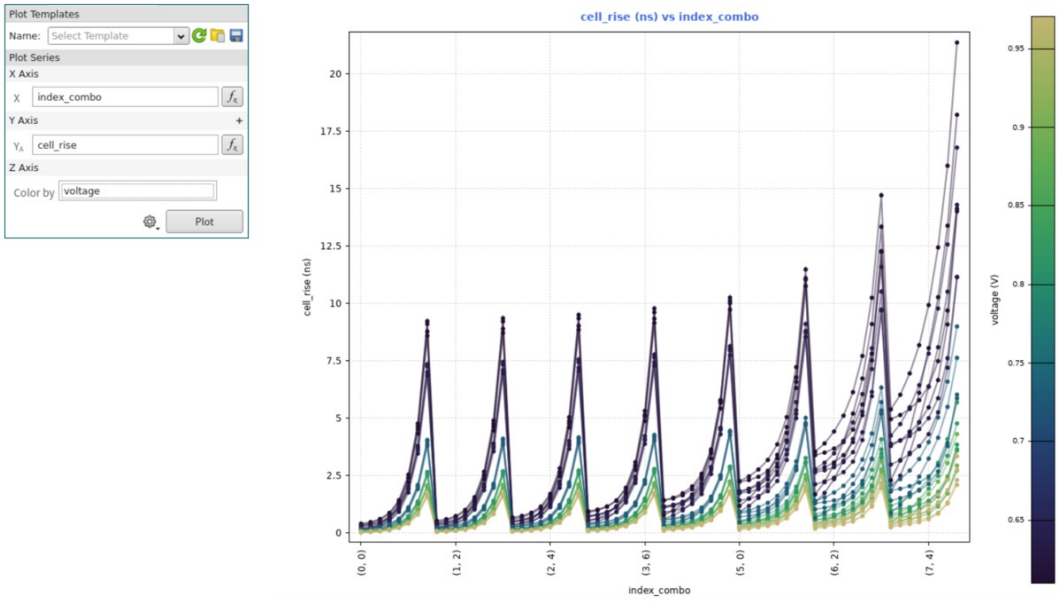


Figure 11: Custom plot for rise delay of a cell across a combination of slew/load values.

Another useful application of custom plots is to divide large quantities of data into smaller visual plots to make it easier to understand. Custom plots such as delay versus drive strength, or power versus drive strength of a family of cells, can be easily generated as shown in figure 12.

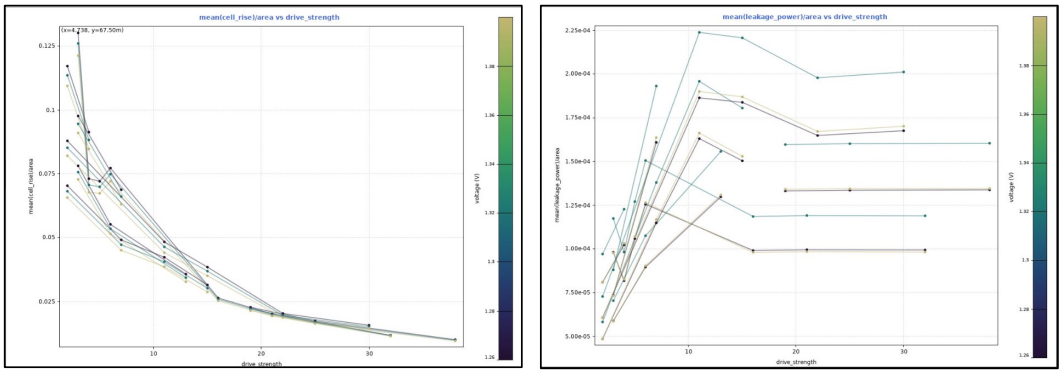


Figure 12: Custom plots for performance and power with respect to drive strength of the cells.

The custom plots feature also supports multiple functions in the same plot. For example, the power and performance relationship of a logic cell is inversely proportional to the supply voltage. As the supply voltage increases, the delay of a cell decreases because the drain current flowing through the device is higher which will charge the load cap faster, resulting in an increase in power consumption. As shown in figure 13, overlaying multiple functions on the same plot is the quickest and most effective method of visualizing this relationship.

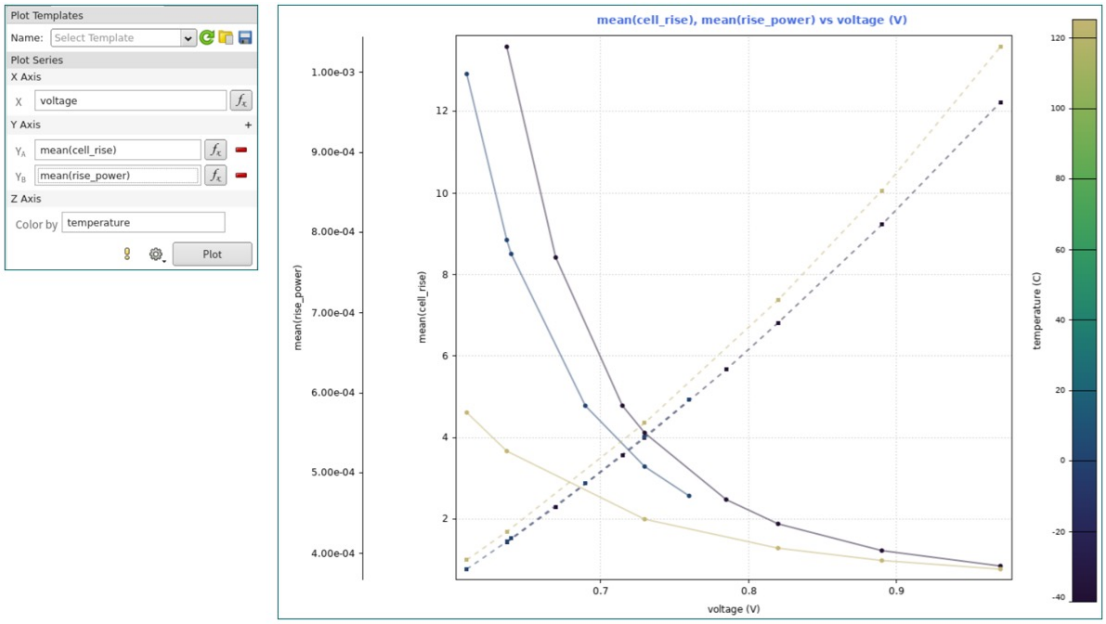


Figure 13: An example of custom plots generated using the custom plot feature in Solido Analytics.

The examples provided here show how custom plots help users analyze and understand .lib data in the fastest, most effective way.

Machine learning outlier detection in Solido Analytics

The ML engine in Solido Analytics also allows it to detect and analyze outliers automatically, providing coverage for a new class of potential .lib issues that are undetectable by other tools. It starts out by building ML models of the entire .lib dataset. Then, it compares all datapoints in the .lib against expected values of the ML models and automatically flags outliers or noisiness in the data. For example, figure 14 below highlights the tool detecting an outlier across temperature of a nominal fall_constraint table.

LVF is critical for a successful tape-out at advanced process nodes. Noisy data or inaccuracies in LVF .libs can lead to schedule delays or even chip failures and moments data is susceptible to characterized inaccuracies. Used together with the visualization GUI, the ML outlier detection capability in Solido Analytics makes it easy to identify and debug errors in LVF modeling. The example plot in figure 15 shows fall_constraint skewness as a function of voltage and how the tool uses trends to identify outliers. The interactive dashboard allows the user to click on any data point on the plots to bring up relevant information about it.

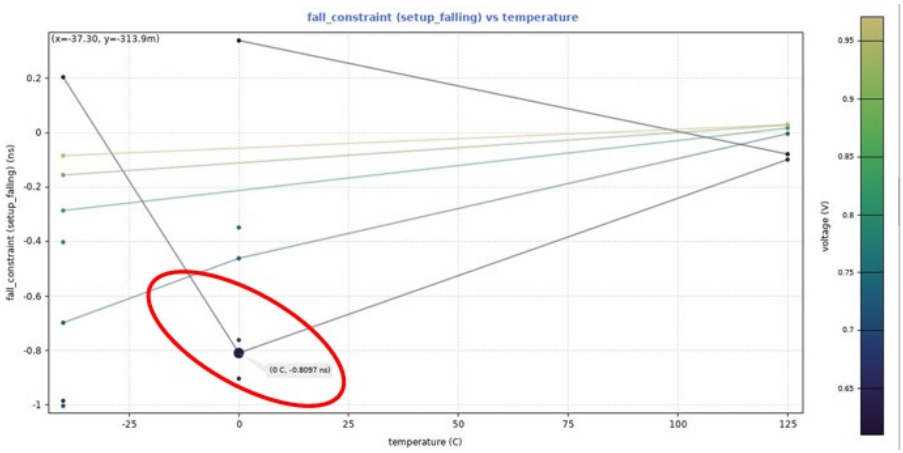


Figure 14: The ML outlier detection check in Solido Analytics can detect any spikes/noise in the .lib data.

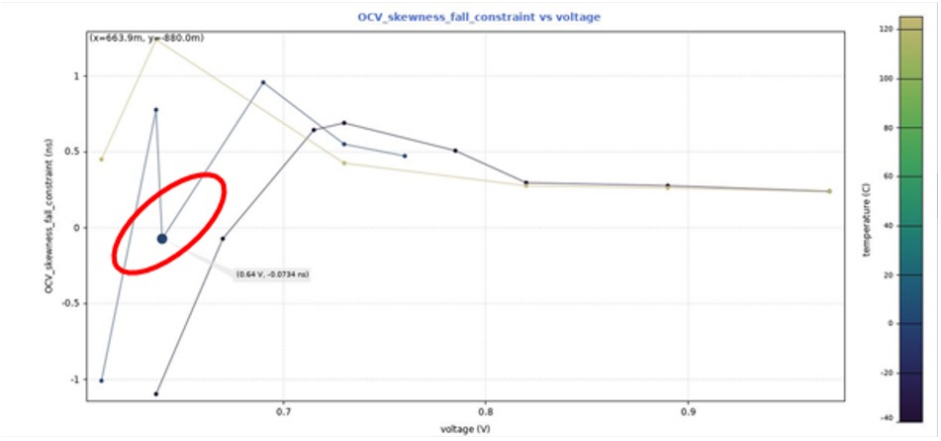


Figure 15: Moments Outlier detected in fall_constraint skewness LVF moments data.

Conclusion

In recent years, the number of SPICE simulations required to characterize modern libraries has increased to hundreds of millions of simulations per library set, resulting in weeks to months of characterization runtime with traditional characterization tools. Also, the complexity and large amounts of data in the Liberty files makes verification and analysis increasingly difficult. While traditional methods of Liberty file characterization and verification are still in use, they may no longer be sufficient for nanometer .lib requirements.

The Solido Characterization Suite uses production-proven machine learning techniques to address the challenges of .lib characterization and verification. The two main components of the Solido Characterization Suite are:

- Solido Generator, which uses ML techniques to accelerate library characterization by two to four times and works with all characterization tools and flows equally well
- Solido Analytics, Siemens EDA's next-generation library validation and verification solution that provides static-rule checks and ML outlier detection capability along with a visualization GUI for easy debugging

The ML methods utilized by the Solido Characterization Suite help characterization teams generate "instantly" production-accurate Liberty files for additional PVT corners and provide tools to extensively verify and analyze the .libs quickly and effectively.

Siemens Digital Industries Software

Americas: 1 800 498 5351

EMEA: 00 800 70002222

Asia-Pacific: 001 800 03061910

For additional numbers, click here.

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Xcelerator, the comprehensive and integrated portfolio of software and services from Siemens Digital Industries Software, helps companies of all sizes create and leverage a comprehensive digital twin that provides organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

84364-D5 1/22 A