DIGITAL INDUSTRIES SOFTWARE

Accelerating AMS verification with Mixed-Signal Visualizer

Executive summary

Debugging now consumes approximately 47 percent of integrated circuit (IC) verification time due to increasing semiconductor design complexity. Analog mixed-signal (AMS) verification presents unique challenges, as engineers must correlate continuous analog behavior with discrete digital logic across multiple abstraction levels, hierarchical configurations, and voltage domains.

Traditional waveform-based debug tools are insufficient for modern AMS verification, resulting in poor observability and inefficient workflows. Engineers require a unified solution capable of handling multiple modeling formats, complex boundary element (BE) configurations and industry-standard methodologies like the Unified Power Format (UPF) and the Universal Verification Methodology (UVM).

Siemens EDA's Symphony Pro with Mixed-Signal Visualizer (MS Visualizer) addresses these challenges through a comprehensive graphical debug environment featuring multi-window design visualization, automatic schematic generation, mixed-signal net analysis, boundary element inspection, and logic cone tracing.

Real-world case studies demonstrate MS Visualizer's effectiveness in diagnosing complex issues, such as multiple-driver scenarios and boundary element configuration errors. Symphony Pro with MS Visualizer reduces verification time, accelerates issue resolution, and helps teams deliver designs to market faster.

Authors: Luong Nguyen and Neel Natekar

Introduction

In a typical modern IC verification cycle, engineers spend a significant portion of their time debugging. A 2024 study conducted by Siemens EDA and Wilson Research Group discovered that IC/ASIC verification engineers spend approximately 47 percent of the time on debug activities, while the remaining 53 percent is allocated to test planning, testbench development, simulation creation and execution, and other verification tasks [1]. Several underlying factors are responsible for this trend. Semiconductor designs are becoming increasingly complex, incorporating features such as multiple power domains, aggressive clock and power gating schemes, innovative circuit topologies, etc., all of which require more time to design and verify.

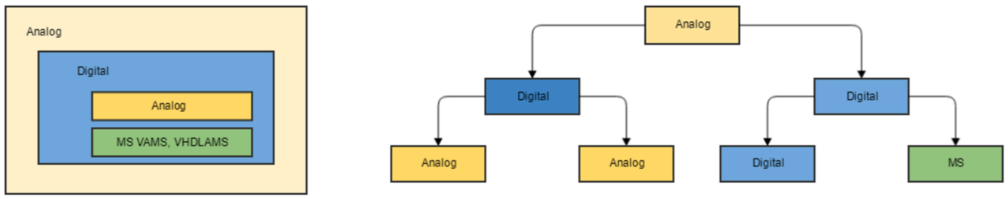

Figure 1: An example of a mixed-signal "sandwich" design

AMS verification compounds the verification challenges, as engineers must correlate continuous analog behavior with discrete digital logic while navigating diverse modeling abstractions. Designers are increasingly implementing complex hierarchical configurations such as “sandwich” structures, which involve interleaving analog and digital blocks, adding another layer of complexity.

This complexity is often amplified by tool fragmentation. In conventional mixed-signal verification flows, the debugging environment has very limited correlation with the underlying simulation engine, resulting in poor observability, missing simulation data and a subpar debug experience.

Consequently, verification engineers require a unified and intuitive debug solution that can deal with such complexities to diagnose issues accurately and efficiently.

Analog mixed-signal verification and debug requirements

1. Efficient AMS simulation requires handling multiple abstraction levels within a unified testbench. As designs scale, simulation performance becomes increasingly dominated by analog subcircuits, especially those with high-frequency behavior such as voltage-controlled oscillators (VCOs), radio frequency (RF) amplifiers, etc. A common strategy is to replace analog blocks with digital behavioral models when this substitution does not compromise functional verification requirements.

Typical modeling formats include:

- Analog circuits described in SPICE formats
- AMS behavioral models in Verilog-A, Verilog-AMS, or VHDL-AMS formats
- Digital logic described in Verilog or VHDL formats

The AMS simulation and verification environment must support all these languages.

2. Modern AMS designs involve extensive interactions between analog and digital blocks at various hierarchical levels. The AMS simulator needs to seamlessly coordinate these interactions. For example, a configuration may traverse analog and digital domains multiple times as shown in the previous "sandwich" structure case. Engineers need a clear overview of the mixed-signal configuration to ensure that all the blocks in the design are represented by their desired or intended views. For instance, failing to specify a digital view to represent a high-frequency oscillator can lead to a big impact on simulation runtime. Providing designers with visibility into which views are present in the design is therefore essential.

3. Low-power designs often include multi-voltage domain architectures. When critical signals cross analog and digital boundaries, engineers must verify that the analog-to-digital (A2D) or digital-to-analog (D2A) conversion is correct. The conversion takes place through boundary elements that act as an interface and translator between nets, signals, elements, and domains. The ability to view various parameters associated with these BEs, such as signal threshold, drive strengths, and reference voltages, is important for debug, as an incorrect setting can compromise the functionality of the design.

4. Recently, real number modeling (RNM) in SystemVerilog or Verilog-AMS is becoming increasingly popular for modeling analog blocks, particularly in power management applications. It uses real numbers (64-bit floating point) or "wreal" (wire real) to carry analog information like voltage or current in and out of the blocks. These real-valued signals may interact directly with analog circuits through real-to-voltage conversions. Their values often depend on upstream multiplexing logic. Therefore, when an analog supply behaves unexpectedly, engineers need visibility into the actual digital driver to diagnose the issue (Figure 2).

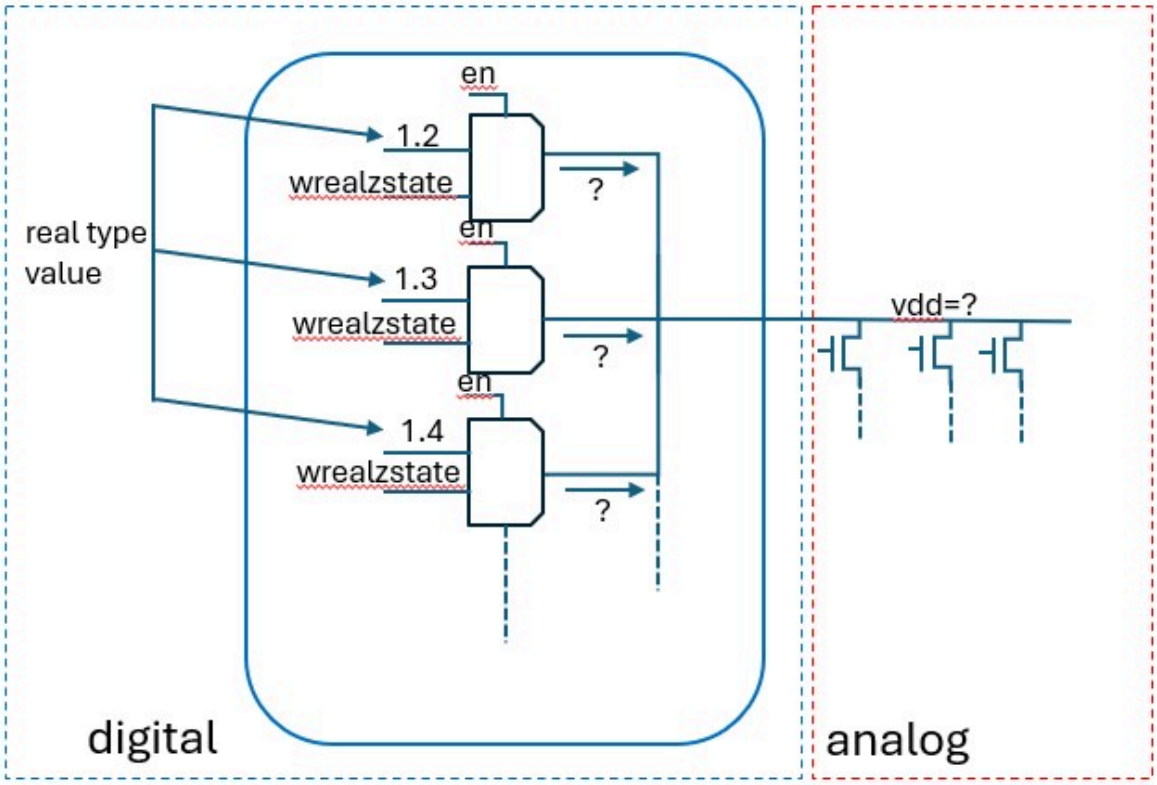

Figure 2: Real Number Modelling (RNM) technique to pass a real value as a power supply of an analog block

5. To effectively and efficiently verify modern designs, verification engineers resort to techniques such as leveraging UPF, which describes the power intent of a design, and UVM, which is a standardized framework for verification. It is equally important for both the mixed-signal simulator engine as well as the debug environment to support these formats. Thus, an AMS simulator and debug environment needs to be robust, flexible and comprehensive to meet the needs of modern AMS verification flow.

Conventional analog circuit simulation relies on waveform viewers as a primary debug method and text reports for debugging design related information. However, for AMS verification, waveform viewing and text reports are no longer sufficient to meet the debug requirements due to the various complexities described above. A graphical debug environment capable of streaming simulation data, visualizing hierarchy, displaying connectivity, and tracing signal interactions is paramount.

Mixed-Signal Visualizer

Siemens EDA's Symphony™ part of Solido™ Simulation Suite, is a highly configurable mixed-signal technology designed to accurately verify design functionality, connectivity, and performance across analog/digital interfaces at all levels of the design hierarchy and for all IC applications. Symphony Pro is the advanced tier of Symphony and an all-Siemens solution that works with all Siemens analog solvers plus the Questa™ digital simulator, the industry-leading, high-performance, multiple-language digital simulator. Symphony Pro can support any combination of analog blocks described in SPICE syntax, digital blocks described in HDL languages (Verilog, VHDL) and AMS behavioral models (Verilog-AMS, VHDL-AMS, VerilogA). It produces rich mixed-signal simulation data that can be explored using the Mixed-Signal Visualizer also known as “MS Visualizer” - a comprehensive graphical debug environment. MS Visualizer offers a seamless debug experience across the entire mixed-signal design hierarchy with comprehensive analysis, automation, and ease-of-use.

Below, we highlight key features of MS Visualizer that IC design and verification engineers can leverage for efficient debug of their Symphony AMS simulations, overcoming challenges associated with waveform analysis or text report-based debugging.

1. MS Visualizer offers multiple windows to efficiently debug designs with each window showcasing different attributes.

These windows include:

- Design hierarchy window

The design hierarchy window helps engineers to quickly confirm that the AMS configuration matches expectations. Users can navigate across the design hierarchy to confirm the analog or digital view of the design unit. Color coding (red for analog, blue for digital) makes it easier to identify the type of elements (Figure 3).

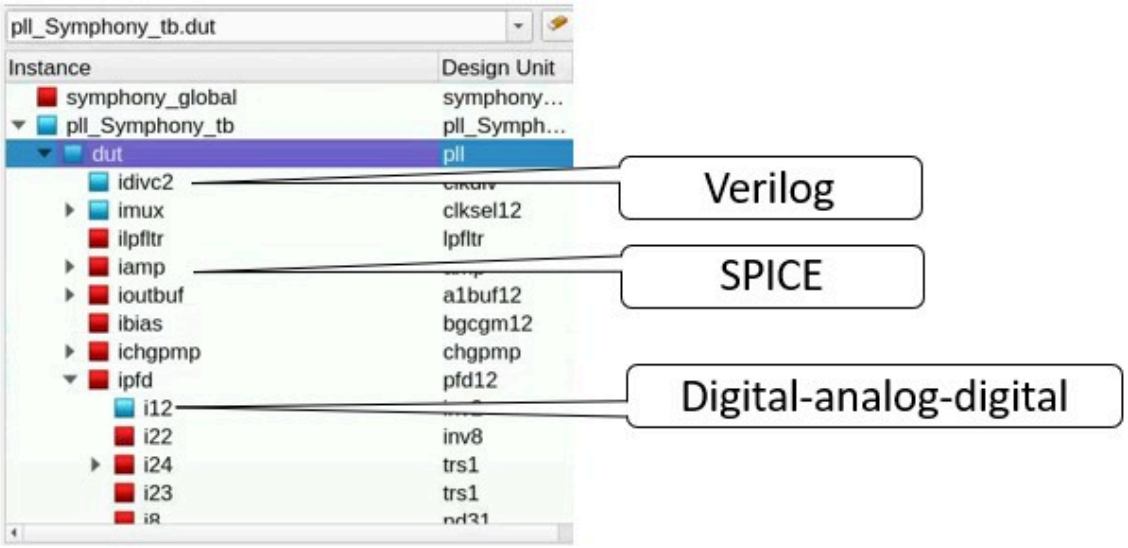

Figure 3: Mixed-signal configuration depicting different elements in the design

- Analog, digital, mixed-signal design netlists or source code and variable windows

  MS Visualizer performs dynamic updates to the source and variable windows during user navigation. The source window displays the SPICE or HDL code of the selected design unit, while the variables window shows object types (reg, real, wire, analog etc.) and their respective values if available (see Figure 4 and Figure 5). This provides additional useful information to verification engineers as they navigate through the design during debug. Figure 5 also shows the inline value annotation which helps to inspect the current state of the design unit.

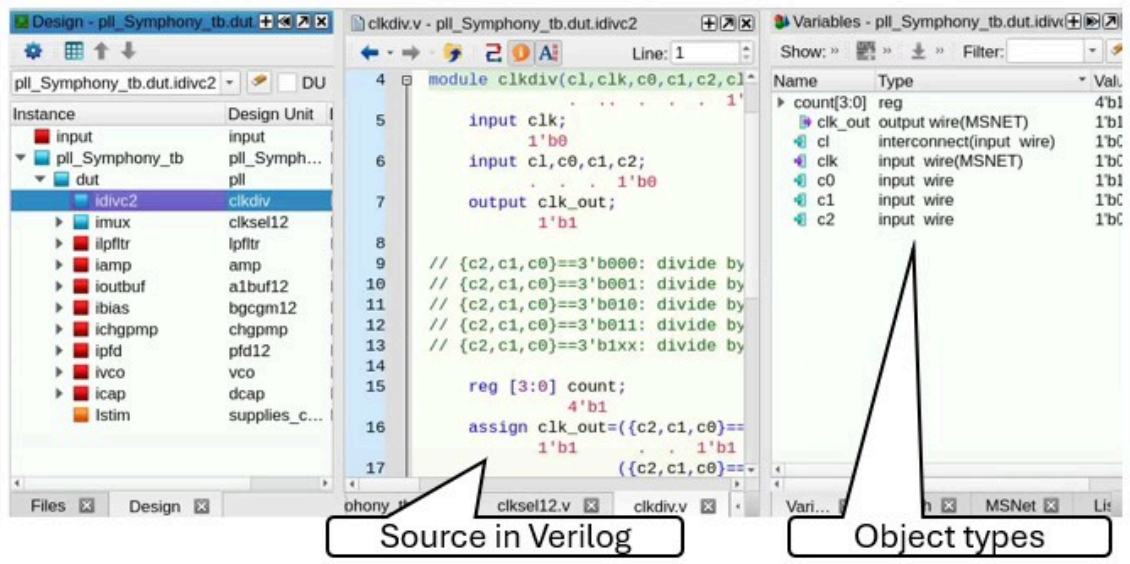

Figure 4: Digital source and object types of a selected design unit

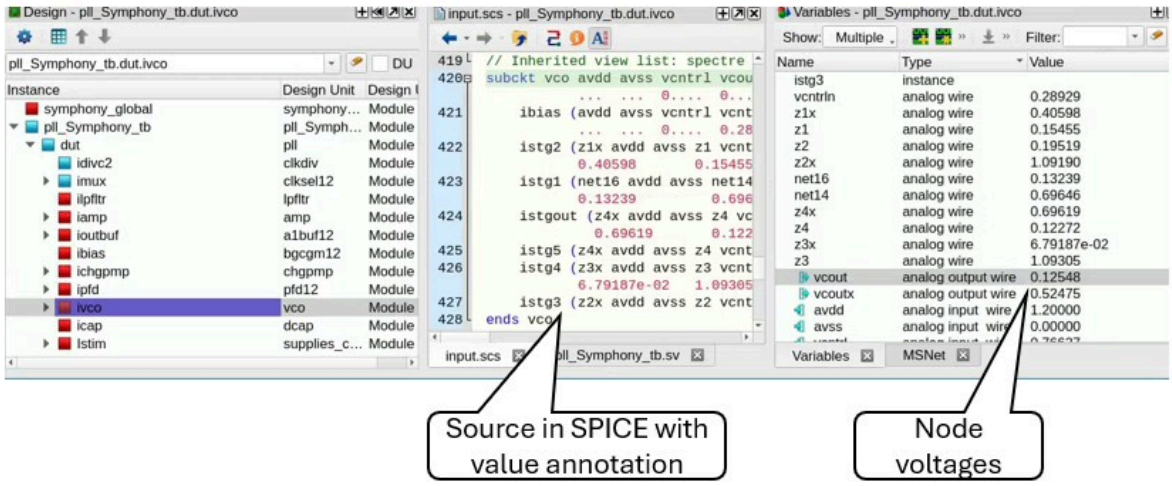

Figure 5: Analog source and object types

- Schematic and hierarchy visualization window

  AMS simulations are commonly run in batch mode with command-line execution and a digital-on-top (DOT) configuration. Often, these setups lack separate schematic views, which makes it difficult to navigate through the design and visualize the connectivity, resulting in suboptimal debugging experience.

MS Visualizer can automatically generate and render schematics to illustrate the connectivity of components in the design. Users can navigate across the design hierarchy within a schematic view, which is especially useful for debugging large hierarchical designs.

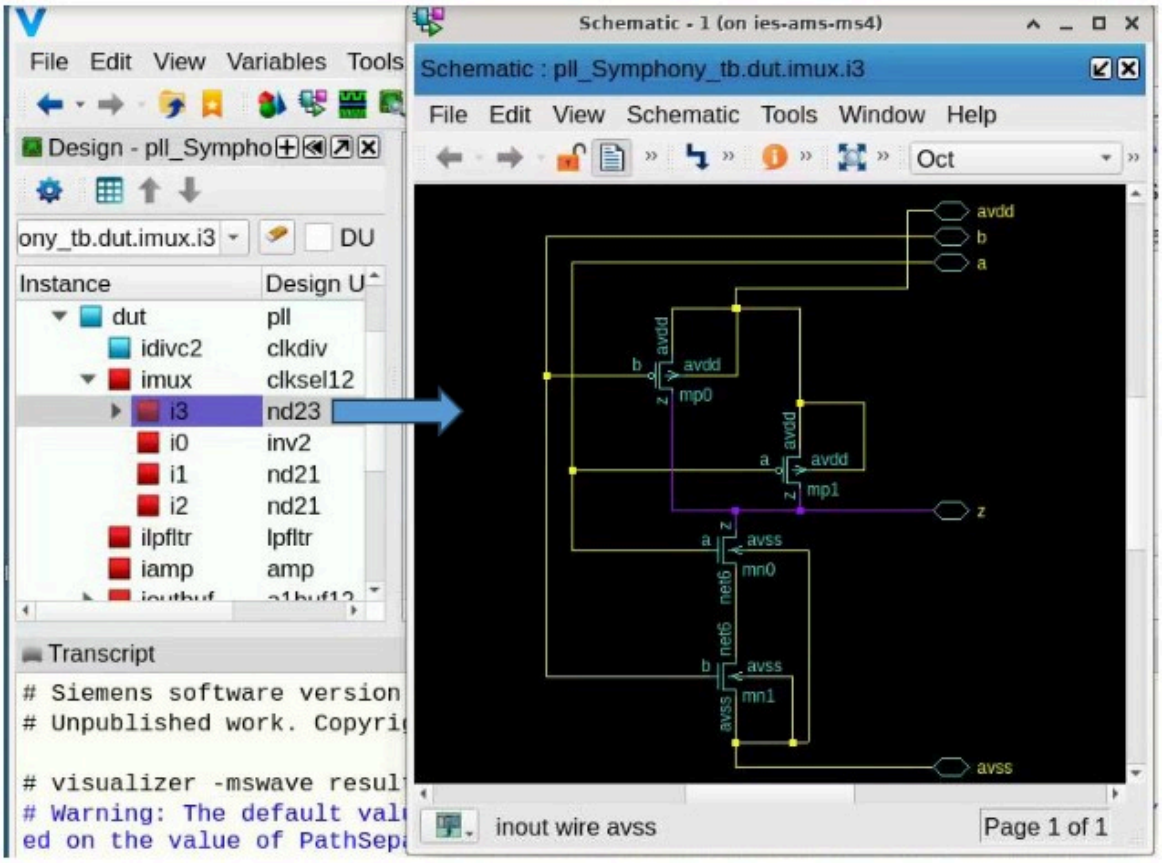

Figure 6: Schematic visualization in MS Visualizer

It is very common to replace an analog circuit with its equivalent digital model (or vice versa) for AMS simulation. Having a schematic allows users to rapidly verify consistency in the port mapping performed by order or by name. As shown in Figure 7, examining the schematic makes it straightforward to identify the incorrect port mapping. The instruction specified to replace the digital design unit with a SPICE model was performed by order; however, the correct approach should have been to use port mapping by name.

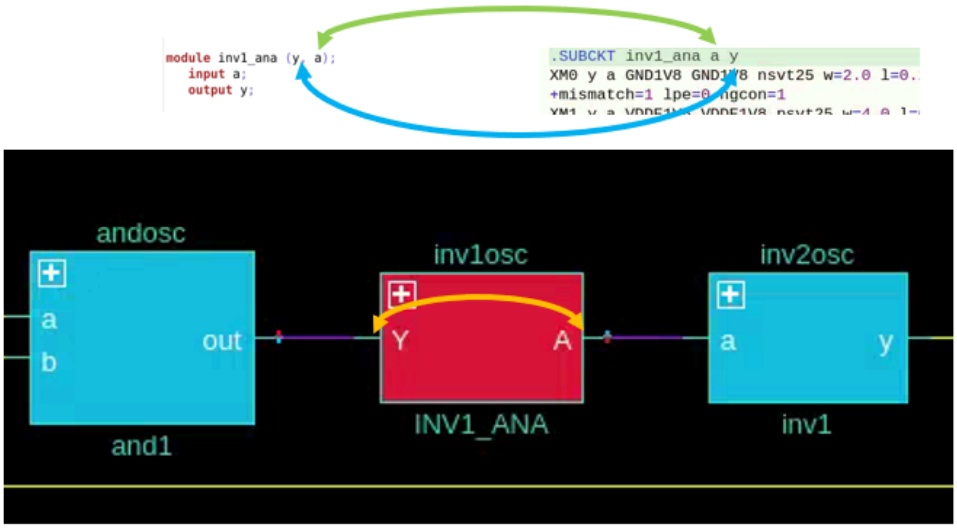

Figure 7: Port mapping check with the schematic view

- Visualize analog and digital waveforms

  Signals that cross analog and digital domains are classified as msnet type (mixed-signal net type). For msnets, MS Visualizer allows users to select the net under test and displays both analog and digital representations to confirm proper conversion behavior from analog to digital (A2D) or digital to analog (D2A), as shown in Figure 8. If the conversion is not working as expected, users can investigate the associated boundary element definition.

Figure 8: Analog (red) and digital (green) waveforms of a given msnet

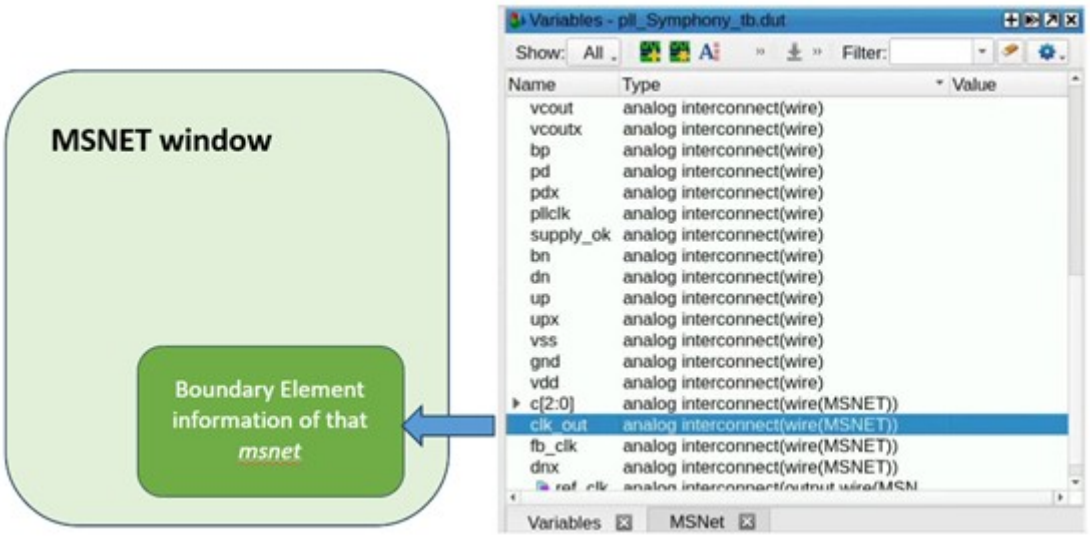

Figure 9: Boundary element (BE) window

- MSNet window

  For any msnet, it is possible to open the MSNet window to query the boundary element definition and verify the reference voltage supply and conversion parameter settings to confirm the design intent (see Figure 9).This is particularly important in multi-voltage supply contexts where hundreds or thousands of msnets exist. The MSNet window serves as the most appropriate reference to identify any suspicious boundary element setup.

- Debug and back tracing

  Once the boundary setup is working as expected, functional verification can begin. MS Visualizer provides the ability to back trace from msnets to the corresponding digital driver signal i.e., the Logic Cone of a signal.

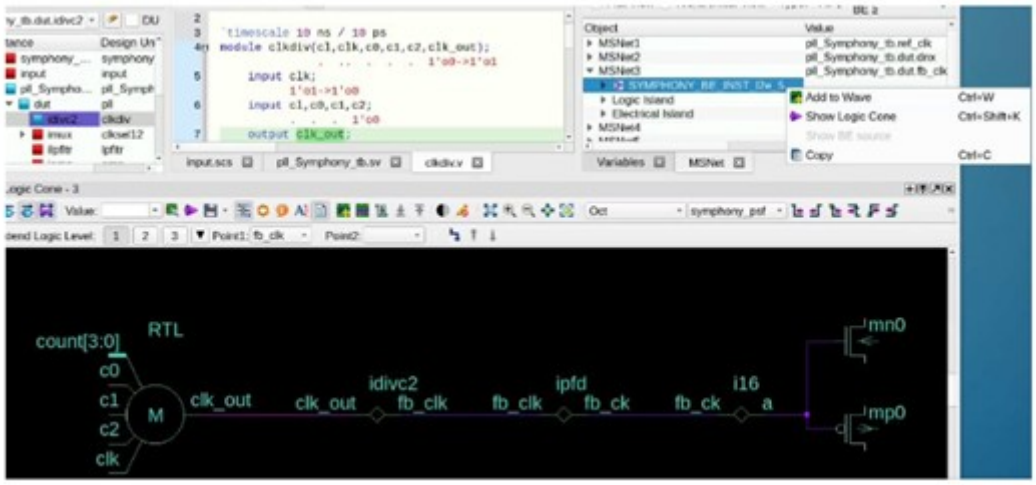

Figure 10: MS Visualizer displaying the trace from msnet to digital driver signal

2. UPF context

MS Visualizer has several useful capabilities to support the UPF power descriptions, as shown in Figure 11.

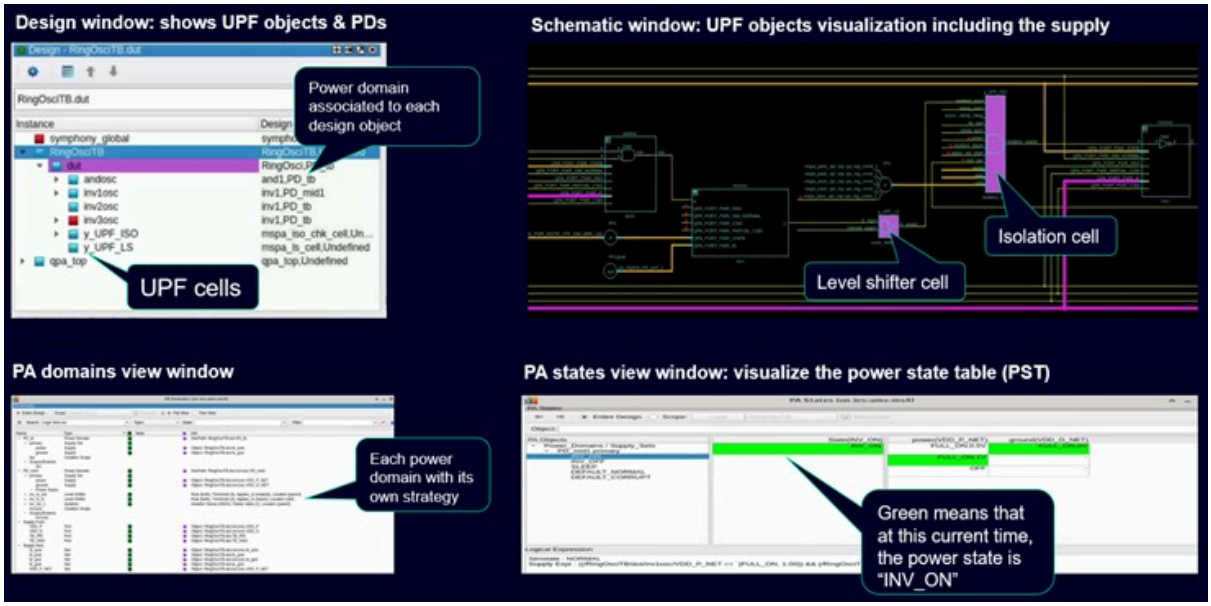

Figure 11: UPF-related capabilities of MS Visualizer

3. Postsim and Livesim features

Siemens Symphony Pro with MS Visualizer provides flexibility to the users to leverage its various debug capabilities in either the Postsim (after the simulation completes) or Livesim (interactive or on the fly during the simulation) modes.

Each mode has distinct advantages:

- Postsim: If key signals were logged, users can perform back-tracing (Show-Logic-Cone) and time cone analysis (Show-Time-Cone)
- Livesim: Users can modify the testbench on the fly using TCL commands and benefit from full signal visibility at the current simulation time without logging everything upfront.

Case studies

We now explore how customers leveraged the MS Visualizer to debug their AMS simulation tests through two real-world customer case studies.

Case study 1: Multiple driver scenario at the analog-digital interface

Challenge:
The customer needed to address a multiple-driver scenario at the analog-digital interface. With a waveform viewer alone, the customer could not clearly back trace the digital driver or diagnose potential X-state or Z-state conditions at the interface. There was a need to identify which digital source was driving the real value into the analog domain at a given time (see Figure 12).

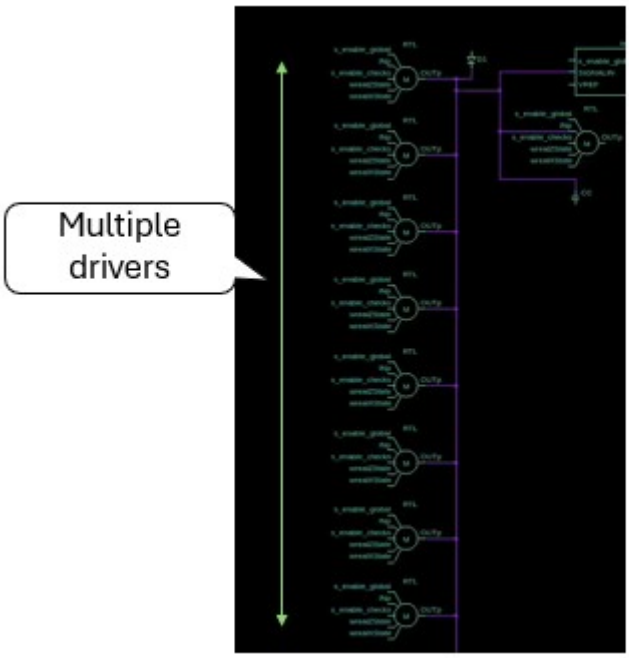

Figure 12: Multiple-driver context at the analog-digital interface

Solution:

To achieve back tracing, the customer displayed the msnet signal waveform under test, positioned the cursor at the transition time of interest, selected the digital representation of the msnet from the waveform window and leveraged the View-logic-cone utility (see Figure 12). The software generated a schematic along the digital driver path. Annotating the schematic with signal values revealed the exact driver passing the value into the analog domain.

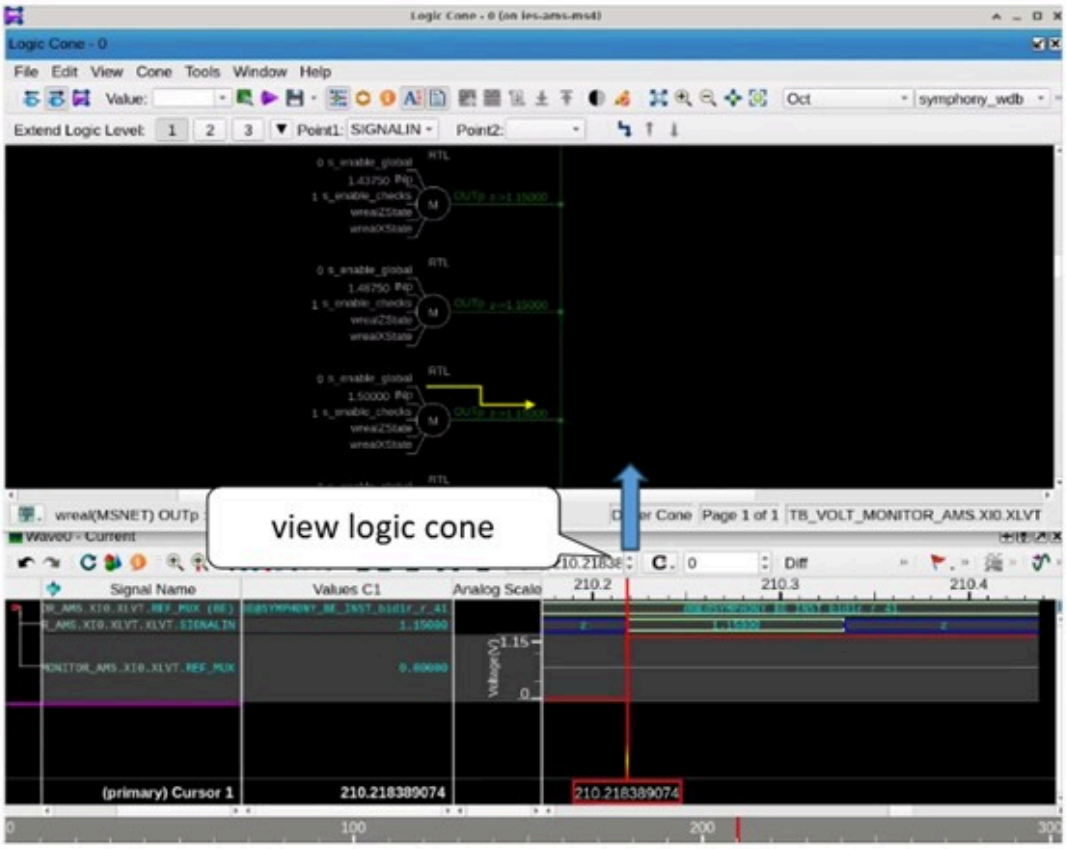

Figure 13: Back tracing signal in a multiple driver context

With MS Visualizer, the customer could therefore verify whether or not the digital value passed to the analog side originated from the correct digital driver at the event time.

Case study 2: Logic Cone analysis for unexpected simulation results

Challenge:

The Logic Cone window helps understand connectivity and is particularly useful when debugging simulation results at a given time that differ from expected results.

Consider the following use case: Digital signal osc_en should transition to '1' when the analog node LDO1V1 rises; however, this was not occurring in the customer use case.

Figure 14: Digital signal osc_en should toggle to '1' when the analog node LDO1V1 rises up, but this wasn't occurring here

Solution:

The customer brought up the Logic Cone window by right-clicking the waveform (or the signal's name from MS Visualizer window) and selecting "Show Logic Cone:"

Figure 15: Initial Logic Cone window illustrating signal origination

This revealed that the signal originated from a buffer. Since there was no ambiguity about which input was driving, MS Visualizer automatically expanded to the next driver, showing that the signal came from an AND gate where one input was '1' and the other was '0'. By following the input driving the signal to '0' and double-clicking on the net connecting to the port of the instance to analyze, MS Visualizer eventually displayed the complete data path, as shown in Figure 16.

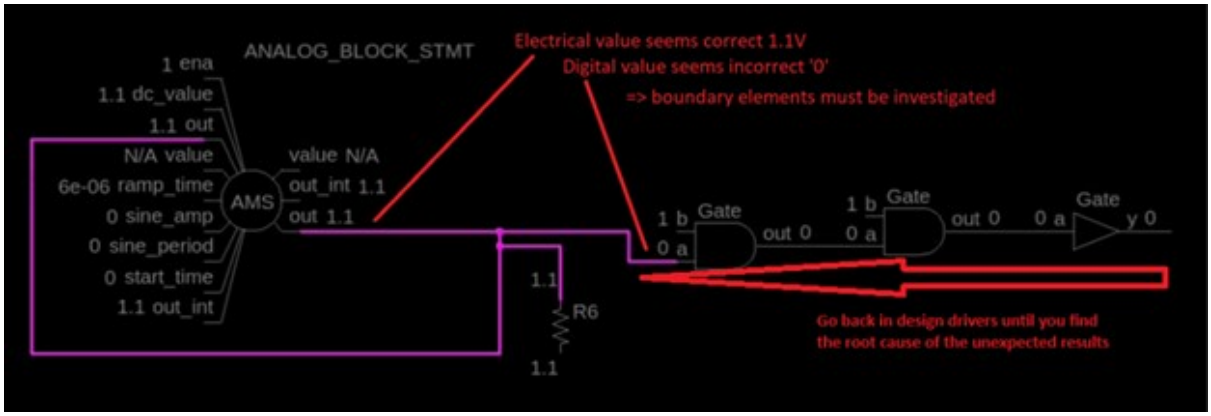

Figure 16: Expanded Logic Cone window illustrating the complete signal path

The analysis revealed that something was not functioning correctly at the mixed-signal boundary: the electrical side of the net was at 1.1V, but the digital port remained at '0.' This indicated an issue with boundary elements. Continuing the debug with the MSNet window (by right-clicking the signal and selecting "Show MSNet Info") automatically opened the MSNet window and selected the mixed-signal net related to the signal being analyzed (see Figure 17).

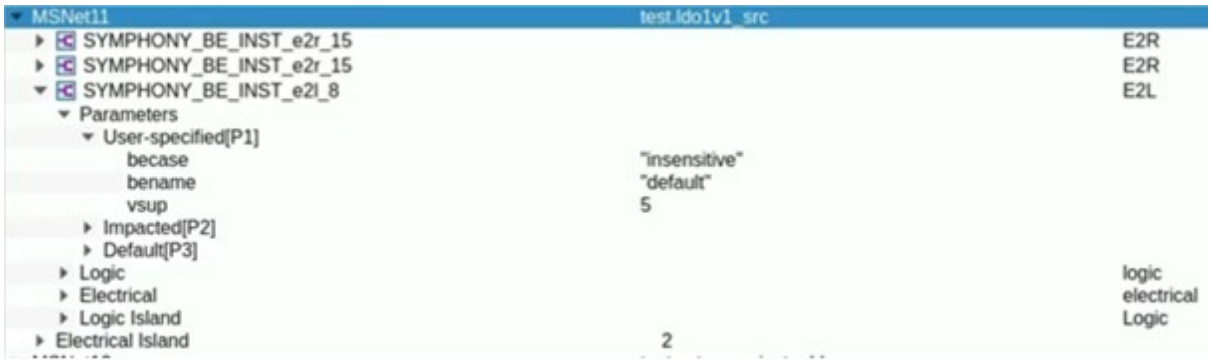

Figure 17: MSNet window revealing the issue with BE configuration

Reading the boundary element parameters revealed the problem: the boundary element was incorrectly configured, with its supply voltage (vsup) parameter set to 5V; whereas the electrical data on the net was at a 1.1V level. Correcting this setup issue resolved the problem and produced the expected simulation results.

Conclusion

As AMS designs continue to evolve and increase in complexity, overall debug time and effort for IC designers and verification engineers continues to expand. Conventional debug techniques using waveform viewers are no longer sufficient, and designers need a solution that can satisfy the requirements of modern AMS designs. Siemens EDA Symphony Pro with MS Visualizer offers a comprehensive and robust solution to address the debug challenges with a unified mixed-signal verification environment. This helps identify issues faster and reduce the overall verification time, thereby helping IC design projects stay on schedule and accelerate time-to-market.

References

[1] H. Foster, "2024 Wilson Research Group IC/ASIC functional verification trend report", Siemens Digital Industries Software, Feb.,2025. https://resources.sw.siemens.com/en-US/white-paper-2024-wilson-research-group-ic-asic-functional-verification-trend-report

Acknowledgements

Thanks to Tanguy Crocy of Siemens EDA for contributing to the case study portion of this paper.

Siemens Digital Industries Software

Contact

Americas (USA)
1-800-498-5351

EMEA (United Kingdom)
0800-279-0464

APAC (India)
1-800-202-6796

For additional numbers, click here.

About us

Siemens Digital Industries Software helps organizations of all sizes digitally transform using software, hardware and services from the Siemens Xcelerator business platform. Siemens' software and the comprehensive Digital Twin enable companies to optimize their design, engineering and manufacturing processes to turn today's ideas into the sustainable products of the future. From chips to entire systems, from product to process, across all industries. Siemens Digital Industries Software – Accelerating transformation.

Luong Nguyen
Neel Natekar

5/2026