Reuse of System-level Circuit Models in Mixed-Signal Verification

Bahaa Osman
Cirrus Logic, Phoenix, USA
bahaa.osman@cirrus.com

Bhanu Singh
MathWorks, Natick, MA, USA
bsingh@mathworks.com

Minghua Li
Cirrus Logic, Austin, USA
minghua.li@cirrus.com

Suhas Belgal
MathWorks, Santa Clara, USA
sbelgal@mathworks.com

Siddharth Maru
Cirrus Logic, Austin, USA
siddharth.maru@cirrus.com

Eric Cigan
MathWorks, Natick, MA USA
ecigan@mathworks.com

Abstract **- This paper discusses some challenges faced in mixed-signal verification of power management ICs. The analog/mixed-signal (AMS) system models that were developed to prove the architecture were then used to generate SystemVerilog DPI-C models for both the analog and digital portions of designs. Use of these models in downstream RTL and AMS simulation environments has been shown to reduce overall debug and verification time. This paper describes the process used to generate the SystemVerilog DPI-C models and incorporate them into the verification framework. It also discusses the limitations encountered and offers suggestions for further improvements.**

Keywords- Mixed-Signal Design Verification, Real Number Modeling, Simulink, Simscape Electrical, SystemVerilog DPI-C models

I. INTRODUCTION

Cirrus Logic specializes in low-power, high-precision, mixed-signal processing solutions for mobile and consumer applications. The company's low-power products include boosted amplifiers, smart codecs, camera controllers, haptic driver and sensing solutions, power conversion and control ICs, and fast-charging ICs.

These semiconductor ICs power devices efficiently sense, process, and communicate information. Power management integrated circuits (PMICs) are vital in portable electronics and medical devices, optimizing power usage to extend battery life. PMICs act as power delivery converters, transforming the voltage from a battery or power source while safeguarding against issues such as overvoltage, undervoltage, overcurrent, and thermal anomalies. Modern power converter designs require co-development of analog and digital subsystems that interact in increasingly complex ways. Evolving silicon processes and advancing performance targets escalate analog design complexity, increasing analog design time and introducing design cycle uncertainty. These factors cause analog design to lag digital design, delaying chip-level mixed-signal verification. Such challenges are not unique to Cirrus Logic; according to a 2022 study [1], analog flaws are the second-largest cause of ASIC respins.

Behavioral models of systems are used for architecture definition and design space exploration for most power delivery network designs. Behavioral models allow early analog and digital co-simulation to prove out architectures under varying conditions. For PMICs, these behavioral models must include modeling of FETs with realistic SPICE-level data to enable running sweeps for power stage sizing and optimization. In fact, implementation specifications are derived from some form of system-level design exploration. Hence, the behavioral system models already embody the required functionality, configurability, and performance.

II. RELATED WORK

Mixed-signal verification became necessary with the integration of analog and digital functions in ASIC designs. The SAMSON simulator from Carnegie-Mellon University researchers was one of the earliest mixed-signal simulators [2]. One of the earliest commercial mixed-signal simulators resulted from the integration of the Lsim logic simulator from Silicon Compiler Systems with MetaSoftware's HSPICE® circuit simulator [3]. Numerous mixed-signal simulators have been developed since, with today's mixed-signal simulators including Spectre® AMS Designer from

Cadence Design Systems [4], Siemens Symphony™ [5], and VCS® AMS from Synopsys [6]. These simulators can provide support for standard analog and mixed-signal modeling languages Verilog-A, Verilog-AMS, and VHDL-AMS as well as Verilog, VHDL, and SystemVerilog.

Starting some years ago, a bifurcation of mixed-signal simulators developed to address the distinct needs of analog designers and digital designers; we now have ***analog mixed-signal (AMS)*** simulators that maintain SPICE-level accuracy and ***digital mixed-signal (DMS)*** simulators that sacrifice accuracy to achieve faster simulation times and higher capacity. DMS simulators make use of real-number modeling (RNM) in SystemVerilog. This approach gained popularity with the advent of the IEEE1800-2012 standard for SystemVerilog [7], which introduced the User-Defined Nettype (UDN). Cadence subsequently introduced the EEnet package with the EEnet SystemVerilog UDN, allowing circuits to be modeled using a combination of Thevenin and Norton equivalents for each EEnet driver [8].

Mixed-signal design and verification methodologies have undergone evolution as well. Kundert and Chang posited that top-down mixed-signal design improves communication in design teams, allows project efforts to be done more in parallel, and improves verification by enabling teams to identify errors earlier [9]. High-level tools such as MATLAB and Simulink have been adopted for top-down mixed-signal design and verification [10] and have been connected to top-down workflows through the generation of SystemVerilog DPI-C models for DMS simulators [11]. In previous DVCon papers, engineers showed how to use generation of DPI-C models from MATLAB® [12] and Simulink® [13] for use in logic simulators to shorten verification cycles.

Those papers showed DPI-C generation from purely digital models, but PMIC designs by necessity include analog components. The Simscape™ tool from MathWorks [14] complements the signal flow paradigm of Simulink by enabling the assembly and simulation of physical systems adhering to continuity and conservation equations such as Kirchoff's current and voltage laws. Simscape uses a dedicated textual modeling language based on MATLAB programming language that supports the definition of custom components as text, complete with parameterization, physical connections, and equations represented as acausal, implicit differential-algebraic equations. Simscape™ Electrical™ extends Simscape with models of semiconductor devices, motors and drivers, and power electronics [15]. In this paper, we extend the generation of DPI-C models to Simscape Electrical to implement a top-down mixed-signal design and verification workflow for use in efficient DMS simulation.

III. MOTIVATION AND PROPOSED WORKFLOW

In ASIC design projects, system models are built for architecture exploration, definition, and refinement. System architects may use a high-level modeling language to build mixed-signal behavioral models and run simulations to validate transient behavior and performance metrics such as bit error rate. The system model is then refined during the design cycle as architecture reviews occur. The model thus acts as an executable specification and is always ahead of the design. The primary benefit of this approach is that an executable model can be simulated by designers and verification engineers to gain understanding of system behavior rather than static text-based specifications.

Mixed-signal design verification presents significant challenges. In most cases, verification teams manually create behavioral models for analog circuits using SystemVerilog Real number modeling (SV-RNM) with User Defined Type (SV-UDT) or Verilog-AMS based Wreal (wired-real) modeling. SV-UDT models are signal flow-based: the outputs are directly computed from the inputs, with signals values that vary continuously but are computed at discrete time steps. These models are primarily used for running digital mixed-signal (DMS) simulations and use a digital solver or discrete event solver rather than a continuous-time solver that evaluates Kirchhoff's current and voltage laws. The development of real-number models involves significant effort, resulting in their availability much later in projects, long after specifications are released. Verification engineers are also challenged by the requirement to have fast but accurate models. The omission of circuit behaviors or compromising on accuracy carries the risk of introducing errors. Verification engineers must build models that accurately capture circuit behavior with adequate precision without slowing down simulations.

We show that with proper structuring, system models can serve as the source for automatic generation of SystemVerilog DPI-C models that may be used in chip-level verification [11, 16]. This approach allows verification engineers to reuse system-level circuit models that have been proven to function correctly by architecture teams as DPI-C models. Just as with purely digital designs, using DPI-C models with mixed-signal designs affords simulation performance improvements, though at some loss in accuracy. Automation of SystemVerilog DPI-C model generation from both analog and digital system models during the architectural exploration phase consequently enables verification to start earlier than in conventional workflows. This automation also enables regeneration of models

throughout the project lifecycle. As the analog and digital designs evolve, the design team can address implementation issues and make system-level design trade-offs with fewer simulator convergence issues.

We built our model using Simulink and incorporated Simscape Electrical components such as resistors, capacitors, op-amps, etc. Simscape Electrical supports a familiar schematic approach for building circuit- and system-level models of PMIC designs. Figure 1 shows a sample Simscape model for a passive high-pass filter.

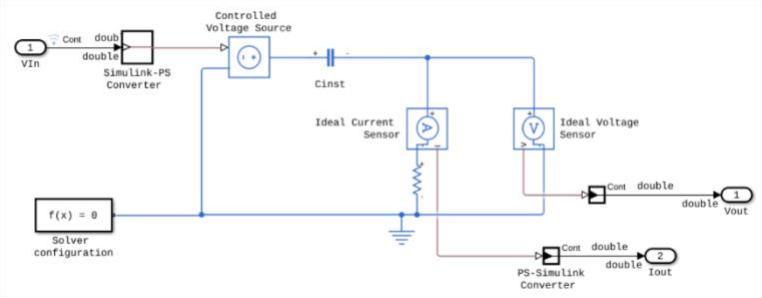


Figure 1: Sample Simscape model for an analog high-pass filter

A fundamental difference between Simulink and Simscape is the nature of signals. Simulink models use the paradigm of signal flow, wherein the value of the signal is determined by the block that produces it; in effect, the input to which a Simulink signal is connected has infinite impedance. In contrast, Simscape uses physical signals that lack implicit directionality – i.e., they behave like a wire in an analog circuit – and have explicit units of current and voltage. Because of these differences between Simulink signals and Simscape physical signals, it is necessary to insert signal conversion blocks when exchanging signals between Simulink and Simscape models. [17].

To transfer the current value in any electrical branch, an ideal current sensor is placed in series; its output is then connected to a PS-Simulink converter block as shown in Figure 1. Likewise, to transfer a voltage value across a circuit element, an ideal voltage source is placed in parallel and connected to the PS-Simulink converter block. Signal values from Simulink are first connected to the Simulink-PS converter block and then can be converted into voltage by connecting to an ideal voltage source. Simulink uses built-in solvers to simulate the model; by default, it uses variable-step solvers that dynamically adjust simulation time steps to optimize accuracy and simulation time. Variable-step solvers employ standard control techniques to access local error at each time step and reduce step size if the error for any state exceeds a specified tolerance [18].

The process of generating DPI-C model from Simulink and Simscape models is illustrated in Figure 2. First, system architects start architecture definition and exploration in Simscape before design starts and simulate the model to check design behavior. To enable code generation, the Simscape circuit model must be encapsulated into a single Simulink subsystem. This subsystem can then be exported as a SystemVerilog DPI-C model[1] that incorporates subsystem behavior into generated C code. However, DPI-C code generation requires fixed-step solver, as the generated C code requires a consistent time step to maintain and update its internal states. It is essential that we select a fixed step size that corresponds to sampling rate faster than frequency of continuous signals in the model to accurately capture the block's behavior while minimizing errors caused by solver configuration changes. Choosing the appropriate step size is vital for translating the model execution from continuous to discrete time, but selecting the time step inherently involves a tradeoff; a smaller step size enables more accurate representation of the continuous model, but it generally comes at the cost of slower simulation speed.

The Simscape circuit also has a solver configuration block where we specify the local solver settings for code generation. This makes the fixed-step Backward Euler solver the default for the connected Simscape circuit and uses our specified sample time. After transitioning to fixed-step solver, we can compare our baseline run that used a variable-step solver against a run with a fixed-step solver. To ensure result consistency, we can compare the output difference and choose a step size that meets the specified maximum error tolerance.

[1] Generation of DPI-C models from MATLAB, Simulink, and Simscape requires the HDL Verifier product.

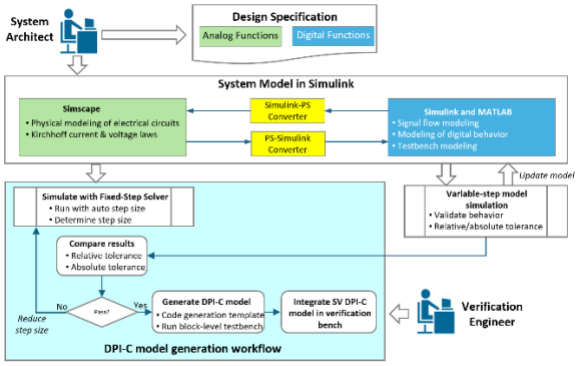


Figure 2: Workflow of generating DPI-C model from a System model

Simscape allows the user to control several circuit component parameters; for example, the capacitance value in Figure 1 can be parameterized and make run-time configurable. The generated C code then represents capacitance as tunable field of a global parameter structure and provides an API function for a set value. The generated DPI-C model is a SystemVerilog module and can be then instantiated in a testbench.

The DPI-C generation workflow provides an option to generate a block-level testbench; a successful simulation with this testbench gives confidence that the DPI-C model's behavior is consistent with the Simulink results. The system model is generally created as a floating-point model. In use cases where an analog model is connected to a digital design that has fixed-point interface signals, data type conversion blocks can be added for float-to-fixed and fixed-to floating conversion. Figure 3 shows an example of buck converter system modeled in Simulink and Simscape Electrical. At the beginning of a project, the DPI-C analog model can be generated from Simscape, while the DPI-C digital model can be derived from the Simulink loop control. This approach enables the rapid deployment of a complete DUT model for simulation. Concurrently, the DV team can initiate the deployment of a constrained-random testbench, incorporating essential checkers to explore the state space thoroughly and ensure robust coverage of primary operating modes. Once the closed-loop system is functional, the digital designer can use chip-level simulations and the DPI-C digital loop control as references for developing the RTL loop control. Then when the RTL loop control is ready, the transition from the DPI-C digital model to RTL is direct.

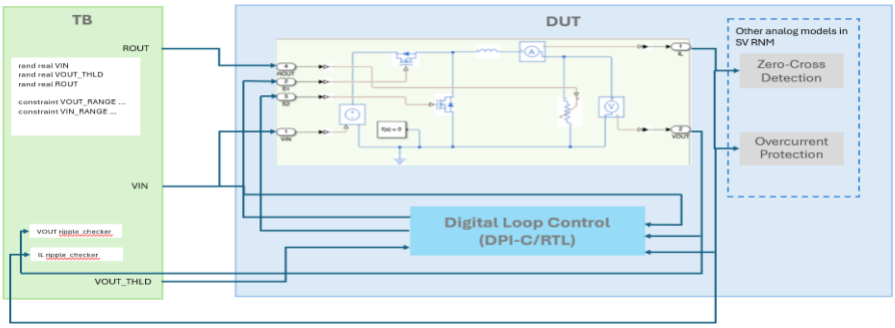


Figure 3: System-level buck converter model built using Simulink and Simscape Electrical

IV. DPI-C MODEL USE CASES

Three DPI-C model use-cases follow the evolution of test environment from initial stages of the project when test bench is being developed to the point where we have fully functional self-checking test environment.

Case 1: DPI-C model for bringing up simulation environment and faster test case development

In the early project stages, the analog design and the equivalent SV-UDT model are not available. In this case, the DPI-C models generated from the corresponding system model can be used for testbench development, enabling test case development with design functionality represented by the DPI-C model. This approach is useful where only

empty stubs (i.e., symbol view) are available for analog blocks. DPI-C models serve as a SystemVerilog view of the analog cell to represent its behavior for initial simulations. Figure 4 illustrates how a DPI-C representation of a DC-DC converter and an ADC design can be used for test case development while those components are being developed.

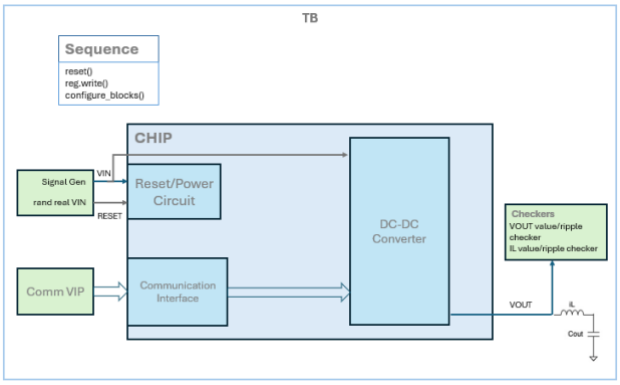


Figure 4: DPI-C model usage in initial stages of project

The DPI-C model can be used in simulations of the chip's complex subsystems where feedback loops and interactions are necessary. For example, in a digitally controlled DC-DC converter where ADCs are needed to close the loop, the system-level ADC model can be replaced with its DPI-C model to allow full functionality while the ADC development is ongoing. This provides a complete design behavior for test case development, removing roadblocks to verification progress and dependencies on the design schedule if the pieces of the subsystem are not ready together.

Case 2: DPI-C model as reference model in scoreboard

Once the design has matured and the corresponding RNM models are available and verified, the DPI-C model can be plugged into a scoreboard to generate bit- and cycle-accurate expected values for comparison to the design's outputs. A suitable tolerance must be incorporated into the scoreboard to allow for design/model variance.

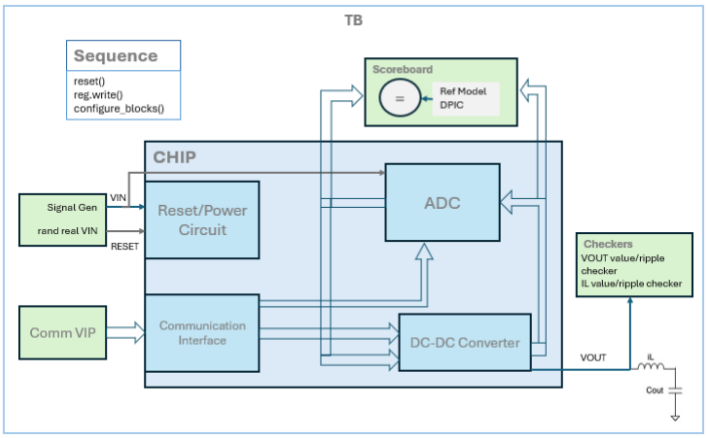
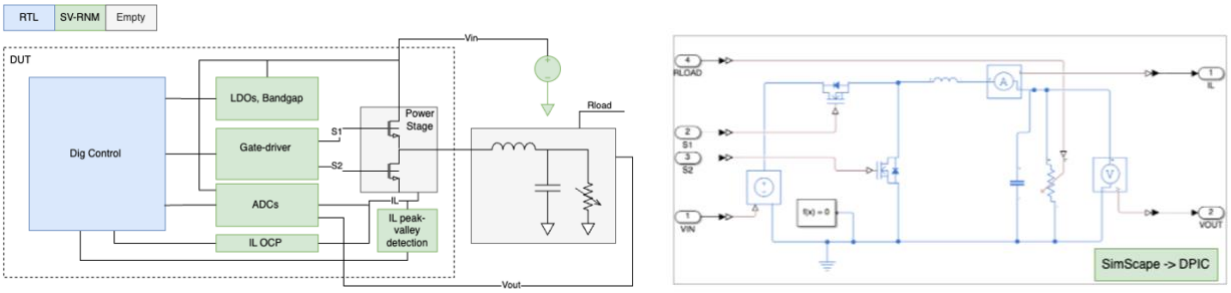


Figure 5: DPI-C Reference model in scoreboard

Case 3: DPI-C model as core engine for simulating power delivery network

Another modeling approach is to use DPI-C as the core engine for simulating the power delivery network. In the buck converter example (Figure 6), the DPI-C model generated from Simscape contains an input-controlled voltage source, two FETs, an inductor, an output capacitor, and a variable resistive load. IL and VOUT are outputs, which sense the inductor current and output voltage.

We use hierarchical reference to interact DPI-C module with DUT. The rest of the supporting blocks are still modeled with SV-UDT. Since the power stage and output load blocks are no longer needed, we insert an empty model for hierarchy integrity.



(a) (b)

Figure 6: Simulink model of a power delivery network (left) using a DPI-C model generated from a Simscape buck converter model (right)

In summary, each of the three modeling approaches – namely, SV DPI-C models generated from Simulink / Simscape, SV-UDT, and Verilog-AMS – provides distinct strengths for mixed-signal verification.

- **Generation of SV DPI-C models from Simulink and Simscape** allows use of a predefined library of physical components and subsystems with more intuitive real-world behavior modeling, reducing the need for custom modeling. In addition, DPI-C generation scales effectively to handle large models, with robust debugging tools, making it well suited for simulating complex power delivery networks without significant degradation.
- **SV-UDT** modeling is simpler for basic modeling tasks, offering a less complex modeling approach for basic elements. Its straightforward model setup process makes it suitable for rapid prototyping of simpler analog systems.
- **Verilog-AMS** excels in accurately modeling fine-grained analog behaviors, making it ideal for scenarios that require high analog fidelity. It is also well integrated into many existing verification workflows, making it easier to adopt without significant workflow changes

V. RESULTS

To demonstrate the variation in simulation times between DPI-C, SV-UDT, and Verilog-AMS models, we utilized a 3-level boost converter as the DUT, operating in an open-loop configuration (Figure 7). For a fair comparison between the digital and mixed-signal solvers, we first simulated the Verilog-AMS model. The output voltage (vout) was converted from an electrical signal to a real number using Cadence's default connect module, and the event count over vout (real) was monitored. Over a 10 ms simulation duration, vout (real) was updated 2,514,522 times, resulting in an effective sample rate of approximately 250 MHz. This sample rate was then used to generate DPI-C code and to configure the SV-UDT trigger frequencies for the capacitors and inductor.

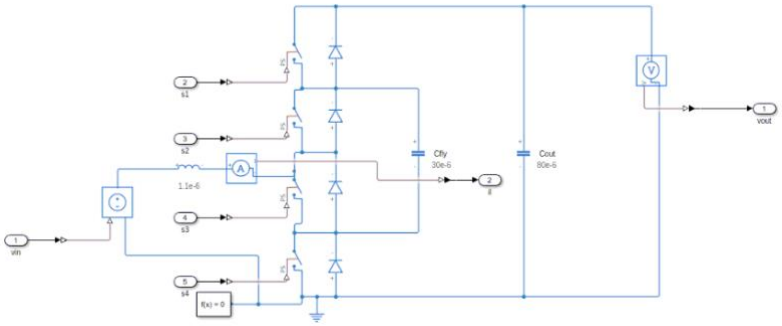


Figure 7: A 3-level boost converter

(Runtime=10ms, Fsw=250K)	DPI-C (Fs=250MHz)	SV-UDT (Fs=250MHz, Vtol=10nV)	Verilog-AMS
Boost 3-level OL	30s	51s	54s

Table 1: Simulation performance comparison

As illustrated in Table 1, DPI-C models achieved a simulation speedup of 1.7x compared to SV-UDT models and 1.8x compared to Verilog-AMS models. In this experiment, the sampling frequency of the digital solutions (DPI-C and SV-UDT) was aligned with the step size of the Verilog-AMS simulation to ensure comparable results, corresponding to a sample rate of 250 MHz with a switching frequency of 250 kHz. In practical designs, the sampling frequency of digital solutions can often be reduced further, enabling even greater throughput compared to Verilog-AMS models.

To illustrate the impact of this proposed approach on actual projects, we offer a comparison of the RTL drop timelines for two recent Cirrus Logic projects (Figure 8). While it can be difficult to compare the timelines of different projects, this comparison indicates the positive impact of DPI-C usage. Project A was done using our previous approach, under which the DV team needed to wait for the RTL, while in Project B, the DV team worked with DPI-C models generated from Simulink and Simscape.

Project A faced delays in achieving a closed-loop design due to the lack of a reference. This impacted the overall verification process and project scheduling, and it took 43 weeks after the first RTL drop to achieve closed-loop functionality. Project B established a chip-level testbench much earlier in the project life cycle, allowing the DV team for Project B to achieve closed-loop functionality just 18 weeks after the first RTL drop.

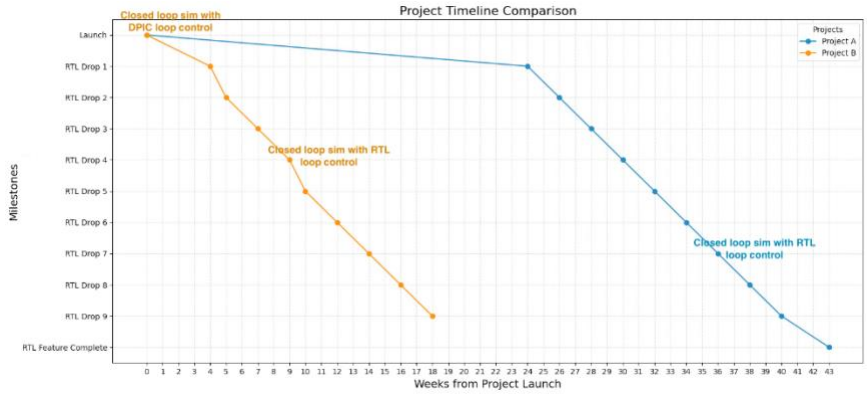


Figure 8: Comparison of RTL drop timelines for Projects A and B

VI. LIMITATIONS ENCOUNTERED

- The current version of SV DPI-C models generated from Simscape lacks support for EEnet.
- Support event-driven simulation to speed up DPI-C performance during POR, low power standby state, etc.
- Create grouped Simscape components to reduce the number of nodes (e.g., group a switch with a body diode).

VII. CONCLUSION

We can summarize the benefits of using SV DPI-C models generated from Simulink and Simscape:

- **Global solver efficiency**: DPI-C uses a global solver, ensuring better numerical stability and accuracy across simulations and leading to more reliable results.
- **Reduced maintenance effort**: As DPI-C has greater precision, users encounter fewer discrepancies and glitches compared to SV RNM. This reduces time spent in debugging, which can be complex in models having delta-cycle based operation. The delta cycle-based execution leads to hard-to-track issues, requiring extensive debugging at the driver level.
- **Consistent performance**: Due to its deterministic nature, DPI-C provides consistent simulation outcomes, avoiding unexpected artifacts that can arise.

- **Faster simulation speed**: DPI-C simulations typically run faster than Verilog-AMS making it better suited for large verification campaigns.
- **Efficient resource utilization**: DPI-C uses fewer computation resources compared to Verilog-AMS, allowing for more parallel simulations and higher verification coverage.
- **Better performance with digital verification**: DPI-C integrates seamlessly with digital verification flows, supporting mixed signal co-simulation in a stream-lined manner, which can be challenging with Verilog-AMS.
- **Robust debugging tools**: The debugging tools available in Simscape-DPI-C are more aligned with digital verification, making issue resolution more efficient.

We observed these overall benefits from our adoption of this workflow:

- Use of generated DPI-C models enables high confidence verification collateral development, reducing rework.
- Verification schedules can be shifted left because stimulus and checks are developed concurrently with hardware design and automatically updated for specification changes.
- DV engineers spend less time developing reference models for score-boarding.
- Reuse of a proven model using DPI-C model generation enables higher quality checking.

VIII. REFERENCES

[1] Siemens EDA, 2022 Wilson Research Group Functional Verification Study. https://blogs.sw.siemens.com/verificationhorizons/2022/10/10/prologue-the-2022-wilson-research-group-functional-verification-study/

[2] Sakalah, Karem A. and Director, Stephen W., (April 1982)"An Event Driven Approach for Mixed Gate and Circuit Level Simulation," DRC-18-45-82, http://shelf2.library.cmu.edu/Tech/15516670.pdf

[3] Odryna, Peter, (07-09 June 1988)"A Unified Mixed-mode Digital/Analog Simulation Environment," 1988., IEEE International Symposium on Circuits and Systems, Espoo, Finland, https://ieeexplore.ieee.org/document/15067

[4] Cadence Design Systems, "Spectre AMS Designer: Advanced mixed-signal simulation solution," https://www.cadence.com/en_US/home/resources/datasheets/spectre-ams-designer-ds.html, Datasheet, Accessed 28 October 2024.

[5] Siemens, "Symphony mixed-signal platform: Industry's fastest and most versatile mixed-signal platform," 81766-D6 Datasheet, June 2022, Accessed from https://resources.sw.siemens.com/en-US/fact-sheet-symphony-mixed-signal-platform on 28 October 2024

[6] Synopsys, "VCS AMS: Mixed-Signal Verification Solution," https://www.synopsys.com/content/dam/synopsys/verification/datasheets/vcs-ams-ds.pdf, 2017, Datasheet, Accessed 28 October 2024

[7] "IEEE Standard for SystemVerilog--Unified Hardware Design, Specification, and Verification Language," in IEEE Std 1800-2012 (Revision of IEEE Std 1800-2009) , vol., no., pp.1-1315, 21 Feb. 2013, doi: 10.1109/IEEESTD.2013.6469140

[8] Caicedo, Alvaro and Fritz, Sebastian. (2019 October 29-30), "Enabling Digital Mixed-Signal Verification of Loading Effects in Power Regulation using SystemVerilog User-Defined Nettype" [Paper presentation]. 2019 DVCon Europe, Munich, Germany. https://dvcon-proceedings.org/wp-content/uploads/enabling-digital-mixed-signal-verification-of-loading-effects-in-power-regulation-using-systemveriloguser-defined-nettyp.pdf

[9] Kundert, Ken and Chang, Henry, (21 June 2005) "Top-down verification guides mixed-signal designs," EE Times https://www.eetimes.com/top-down-verification-guides-mixed-signal-designs/

[10] Halupka, David, et al. "Validation Shift-Left: Enabling Early SerDes Mixed-Signal Validation" DesignCon 2022 conference paper. Retrieved September 11, 2024, from https://www.mathworks.com/content/dam/mathworks/white-paper/validation-shift-left-enabling-early-serdes.pdf.

[11] DPI Component generation with Simulink, https://www.mathworks.com/help/hdlverifier/ug/dpi-c-generation-overview.html

[12] Lakshminarayana, Avinash, et al. (2022 February 28 – March 3), "Flattening the UVM Learning curve: Automated solutions for DSP filter Verification" [Paper presentation]. 2022 DVCon US, virtual conference, https://dvcon-proceedings.org/document/flattening-the-uvm-learning-curve-automated-solutions-for-dsp-filter-verification-2/

[13] Tata, Aswini, et al. (2024 March 4-7), "Leveraging Model Based Verification for Automotive SoC Development" [Paper presentation]. 2024 DVCon US, San Jose, CA USA. https://dvcon-proceedings.org/wp-content/uploads/1031-1.pdf

[14] MathWorks, Simscape documentation, https://www.mathworks.com/help/simscape

[15] MathWorks, Simscape Electrical documentation, https://www.mathworks.com/help/sps

[16] MathWorks HDL Verifier Team, "Best Practice Guide for SystemVerilog DPI Generation (version 1.0.0)," MathWorks File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/176168-best-practice-guide-for-systemverilog-dpi-generation, accessed December 10, 2024

[17] MathWorks, "Essential Physical Modeling Techniques," Simscape documentation, https://www.mathworks.com/help/simscape/ug/essential-physical-modeling-techniques.html

[18] MathWorks, "Variable Step Solvers in Simulink," Simulink documentation, https://www.mathworks.com/help/simulink/ug/variable-step-solvers-in-simulink-1.html