Mixed-Signal Design Verification: Leveraging the Best of AMS and DMS

Rock Z. Shi
Analog Devices, Inc.
Wilmington, MA
Rock.Shi@analog.com

Padmashree Bhinge
Analog Devices, Inc.
Bangalore, India
Padmashree.Bhinge@analog.com

Preston Birdsong
Analog Devices, Inc.
Wilmington, MA
Birdsong.Preston@analog.com

Geeta Chaitanya
Analog Devices, Inc.
Bangalore, India
Geeta.Chaitanya@analog.com

Kunal Jani
Analog Devices, Inc.
Bangalore, India
Kunal.Jani@analog.com

***Abstract*-Analog and mixed-signal (AMS) design and verification methodology exists from the very beginning of mixed-signal IC design practice, but its role has gradually become less clear since the emerging dominance of digital mixed-signal (DMS) methodology. In this paper, we seek to analyze individual roles of AMS and DMS, why both are necessary and complementary, and how we can take advantage of each flow's strength to optimize verification resources and job efficiency.**

I. Introduction

Mixed-signal design of integrated circuits keeps making rapid progress with a never-ending demand for more complex digital functions and features in parallel with requirement for higher analog performance within smaller area and lower power consumption. Verification of mixed-signal chip design has become a more challenging task. On one side, the ever-increasing digital features and complex A-D interactions in a chip often require additional tests to hit full coverage and have led to the trend of DMS (digital mixed-signal) flow adoption for full chip functional verification ([1-3]). On the other side, many performance metrics and detailed circuit behavior of analog design need to be checked at full chip level, but often cannot be accomplished due to the limitation or quality of DMS models.

The Analog and mixed-signal (AMS) design and verification methodology exists from the very beginning of mixed-signal IC design practice. The AMS flow, which has a SPICE engine working with either analog behavioral models (e.g., Verilog-A models) or SPICE primitives, is less favored in modern mixed-signal IC design due to its slower simulation speed compared to its DMS counterpart. Its role has gradually become less clear since the emerging dominance of DMS methodology. In this paper, we seek to re-evaluate the role of the AMS flow vis-à-vis DMS flow, analyze when and why both are needed, and how we can take advantage of each flow's strength to optimize verification resources and job efficiency.

Verification of a mixed-signal chip often requires a team of design verification engineers, the bulk of whom to focus on the digital part of the chip, and a few to be dedicated to the verification of analog blocks and the interaction between analog and digital. This sub-group of the design verification team is referred to as mixed-signal design verification (MSDV) team. Indeed, although MSDV is becoming an emerging industry discipline as well as a dedicated career path (other than analog design or digital design verification), its role and function have not been well defined, and its industry practice is much less reported. It is the interest of this paper, using a typical mixed-signal chip design we have recently developed at Analog Devices (ADI) as an example, to present the spectrum of common MSDV activities. By presenting typical MSDV workflows and analyzing their roles in the full chip verification, we wish to provide some insight on how AMS and DMS can work together, and how we can combine the two flows to optimize resources and tasks.

In the sections that follow, we start in Section II by arguing that analog behavior models are the key to the success of MSDV. The availability and quality of those models largely define the partition of AMS and DMS scope in a project. In Section III, we describe how we can bring the AMS and DMS flows together under single test infrastructure with the UVM platform. In Section IV, we present our results by analyzing test coverage and bug reports from AMS and DMS flows to demonstrate how we can partition and optimize the resources and tasks between two flows. In Section V, we suggest that MSDV is becoming a separate discipline and opening a new career path. We conclude that AMS is still critical and can be combined with DMS to fulfil MSDV in a more efficient way.

II. Modeling for Analog Circuits: Key to the Success of MSDV

The availability and quality of models for analog circuits have a large impact on the flow to use and the quality of the verification. As in real life of digitization where not all analog circuits can be easily replaced by digital equivalent, DMS modeling has been found its limitation in capacity and efficiency when modeling some type of analog circuits. The real number modeling, being SystemVerilog real nettype or Verilog-AMS wreal, is in essence a digital modeling approach that analog designers are not familiar with, that always requires a digital simulator in addition to the SPICE engine for analog designers' test benches. Analog designers continue to prefer using AMS models in their design phase for good reasons. As such, DMS models that are desired for full chip verification are usually not available from the design team. It is thus a task left to the verification team to develop and validate those DMS models, which would put a big constrain on project schedule.

A. Comparison of AMS and DMS Modeling Approach

Real number modeling with SystemVerilog real nettype (referred as SVRNM) or Verilog-AMS (wreal) allows us to achieve the same quality of AMS behavioral models but often with extra efforts. To facilitate the comparison between AMS and DMS model, we define the "quality" of an analog behavior model in the following context:

i) The intended use of the model is for full chip verification. Therefore, the quality of a model is mainly related to its test capability to verify all design specification.

ii) The model should facilitate full chip A-D interface connectivity and function check, as well as any special analog behavior and performance specification verification that must be done at chip level.

iii) The model should be validated (often against schematics) and kept updated with analog design change.

Below we analyze the case for PLL where we have developed both full set of AMS and DMS models that are almost identical in quality. As an illustration, Fig. 1 presents visual comparison for the PLL modeling.

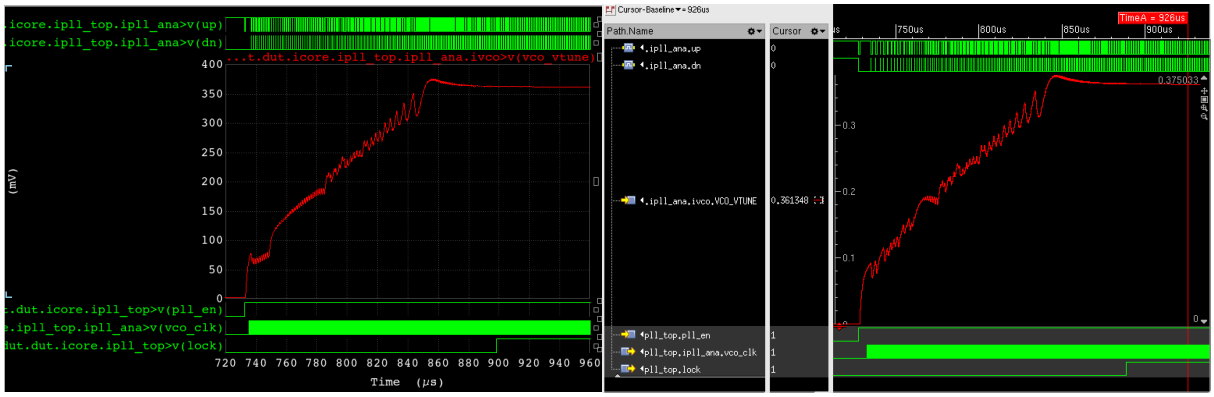

Figure 1. A full set of SVRNM model for PLL block is developed that has about the same quality of the full set of AMS models. Left: AMS model behavior. Right: DMS model behavior. Top two traces are up (v(up)) and down (v(dn)) control signals from the phase detector. The red trace is the VCO turning voltage control signal output from the loop filter. Note that from PLL enable (pll_en) to the time PLL getting in lock (pll_lock), the two models spend about the same time.

To compare the different style and approach between AMS and DMS models, we use the charge pump and loop filter as two examples.

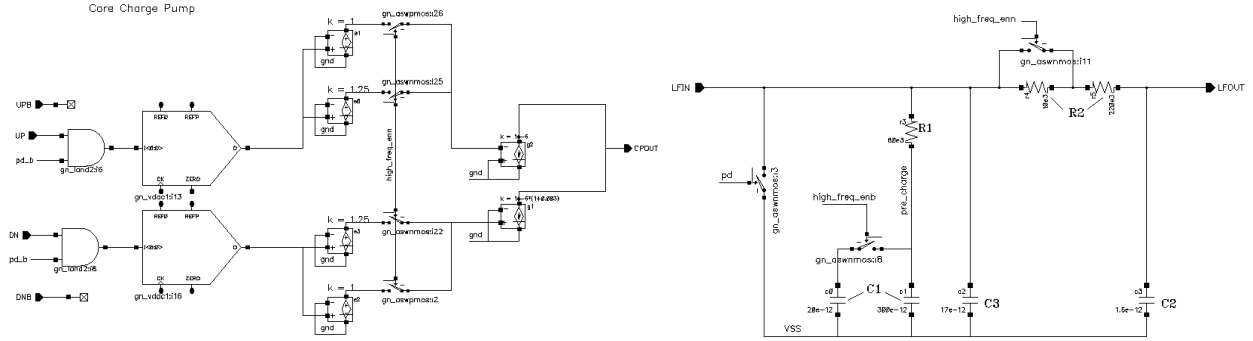

Figure 2. AMS model (core part) for charge pump (left) and loop filter (right). In the charge pump model, charging and discharging currents are modeled by voltage controlled current sources, controlled by UP and DN input respectively. In the loop filter model, AMS model has the advantage of using R and C primitives directly and counting on SPICE engine to solve the transformation from input current (LFIN) to output voltage (LFOUT).

At ADI, a centralized library of general behavioral models and analog primitives are developed and maintained by a central CAD team. To model an analog design block, a designer often creates and maintains equivalent alternative schematics consisting of behavioral model schematic symbols. The level of modeling of any block can be based on simulation speed and performance trade off. When a block is configured to simulate as a behavioral model, ultimately C++ model of each cell in the alternative schematic will constitute the design. These behavioral models are not only used for design but also can be used for stimulus/testbench and for micro-modeling of the blocks which haven't been designed yet. ADI's simulator is a single Kernel simulator, uniquely supports ADI's event-driven (C++) models into analog solver. A C++ behavioural model can have Logic/Voltage/Electrical ports (carrying actual voltages and currents, not just a real number).

For the charge pump in Fig. 2, everything to the left side of the two current sources is the same as in Verilog, but with an ADI's modelling language and simulator. The right half is SPICE simulation again. The extra things are dependent sources which convert voltages/currents to voltage/currents. On the other hand, the loop filter is a SPICE (transistor-level) simulation of resistors and capacitors, using voltages, currents, etc. The switches are behavioural resistors which have an on and an off resistance, controlled by a gate input.

Listing 1 illustrates the DMS model for the charge pump. Note CPOUT can be a "real" variable or "real" nettype (wreal) datatype representing the resultant current to charge (if >0) or discharge (if <0) the loop filter.

```
always @( UP or DN or HIGH_FREQ_EN ) begin
      iup = UP*(1.0e-6 + HIGH_FREQ_EN*0.25e-6);
      idn = DN*(1.0e-6 + HIGH_FREQ_EN*0.25e-6);
end

// output
assign CPOUT = (pwr_bias_ok & PDB)? (iup - idn) : 0.0;
```

Listing 1. Code snippet of DMS model for charge pump. UP and DN are logic/digital inputs in the model.

Without SPICE engine in DMS flow, the DMS modeler himself is the analog solver: He often needs to perform node or loop analysis and solve the differential equations in a digital way. In the simpler case like the charge pump, the DMS model can be derived by applying Kirchhoff's current law (KCL). In a more complex electrical network like the loop filter in our PLL case, more analysis work is needed. One way to implement the DMS model for the loop filter is:

i) Find out the Laplacian transform function of the block, as in (1).
ii) Convert to z-domain transfer function H(z) (3) with the Bilinear Transform (2).
iii) Solve for the output new iteration value by realizing that z^{-1} is the unit delay operator (see [4] for more details)

$$H(s) = \frac{LFOUT}{LFIN} = \frac{n_0 + n_1 s}{d_0 + d_1 s + d_2 s^2 + d_3 s^3} \quad (1)$$

$$s = \frac{2/T_S}{(1 - z^{-1})(1 + z^{-1})} \quad (2)$$

$$H(z) = \frac{N_0+N_1z^{-1}+N_2z^{-2}+N_3z^{-3}}{D_0+D_1z^{-1}+D_2z^{-2}+D_3z^{-3}} \quad (3)$$

Listing 2 illustrates the DMS model for the loop filter.

```
always_comb begin
    C1=C1_TOT-HIGH_FREQ_EN*20e-12;
    R2=R2_TOT-HIGH_FREQ_EN*10e3;

    K=2.0/(Ts*1e-9);
    n0= 1.0; n1=R1*C1;
    d0=0; d1=C1+C2+C3; d2=R1*C1*C2+R1*C1*C3+R2*C2*(C1+C3); d3=R1*R2*C1*C2*C3;
    N0=n0+n1*K; N1=3*n0+n1*K; N2=3*n0-n1*K; N3=n0-n1*K;
    D0=d0+d1*K+d2*K**2+d3*K**3; D1=3*d0+d1*K-d2*K**2-3*d3*K**3;
    D2=3*d0-d1*K-d2*K**2+3*d3*K**3; D3=d0-d1*K+d2*K**2-d3*K**3;
end

initial begin
    while ($realtime == 0) begin //DC op point
        X0 = (LFIN <1e20)? LFIN : 0.0; //consider when input is NaN (Z/X)
        X1 = X0; X2 = X0; X3 = X0;
        Y0 = X0; Y1 = X0; Y2 = X0;  Y3 = X0;
        @(LFIN);
    end // while ($realtime == 0)
end

always #Ts begin
    X3 = X2; X2 = X1; X1 = X0;
    X0 = (LFIN < 1e20)? LFIN : 0.0; //remove Z/X condition
    if (Y0<0.1 && LFIN <0) X0=Y0*LFIN; // discharging stop below 100mV

    Y3 = Y2; Y2 = Y1; Y1 = Y0;
    Y0 = (N0*X0 + N1*X1 + N2*X2 + N3*X3 - D1*Y1 - D2*Y2 -D3*Y3)/D0;
end // always

assign LFOUT = (pwr_ok & PLL_PDN)*Y0;
```

Listing 2. Code snippet of DMS model for the loop filter. Ts is the sampling interval.

There is a trade-off in selecting the value of Ts, the sampling interval. Better model accuracy requires smaller sample period, but this slows down the full chip simulation. In practice, the value of Ts can be chosen from model validation process (e.g., comparing the filter step response between the model and the schematic). The Ts used in Fig. 1 is 1ns.

B. Some Challenges of Real Number Modeling Approach

Signals in real number models by default are voltages (V) for nets or ports. A real value in DMS models can be interpreted or modeled as a current (I) (as CPOUT in the PLL charge pump case), but it cannot be both (V and I). This limitation in general is not a problem, but it does introduce difficulty in modeling where both V and I are in a play on the same net. The output current inside the charge pump is coupled with the voltage on the CPOUT net, however, since we model CPOUT as the equivalent current, the voltage information is out of the module as it is mainly determined from inside the loop filter block. Verilog and SystemVerilog allow out of module reference (OOMR), but OOMR is often found lack of portability and should be avoided in general.

One solution is to use SystemVerilog user-defined type (UDT) to enable a net to carry both voltage and current simultaneously and use SystemVerilog user-defined resolution (UDR) to resolve multi-driver functions (e.g., resolution of summation for KCL). It however would introduce extra effort to handle this special nettype in the whole verification environment. It also makes AMS simulation more complicated such as special interface handling between electrical and digital UDT models.

The need for having both V and I on a single net can be minimized if the circuit partition allows this type of nets sit in the same block. This indicates that adopting the DMS flow may have some impact on circuit structure. In general, DMS modeling flow prefers well partitioned and organized analog circuit blocks (boxes in schematic drawing) to primitives or discrete elements (like metal resistors, decoupling capacitors, or pull-down transistors seen outside any boxes).

To model the CPOUT current dependence on its net voltage, a practical technique used in Listing 2 is to move this modeling part from the charge pump block to the loop filter. The current input (X0) received by the loop filter is modified from the voltage-independent output current (LFIN) to a voltage dependent current source, like the NMOS transistor drain-source voltage effect on its drain current:

```
if (Y0<0.1 && LFIN <0) X0=Y0*LFIN; // discharging stop below 100mV          (4)
```

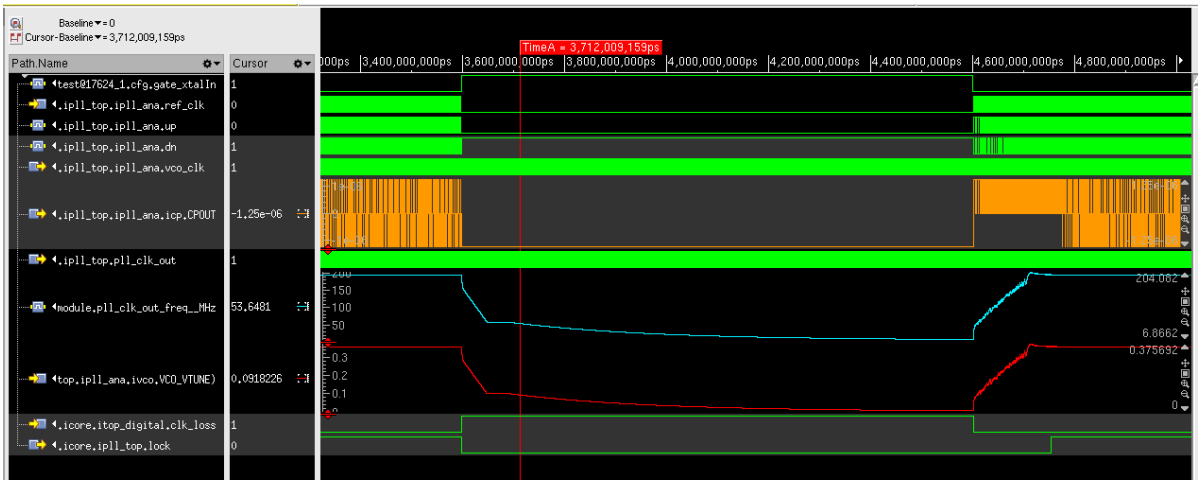

Figure 3. A PLL test case that requires SV real net to have both voltage and current. In the case of "clock loss", the discharging current will gradually be turned off by sensing the voltage feedback.

C. MSDV with a Hybrid Use of AMS and DMS Models

Developing high quality of DMS models can be time consuming. This is especially true if the DMS models are to be developed from scratch, and analog performance specifications need to be modeled and verified. High quality of DMS models often require modeling down to lower level of hierarchical analog design, which in turn makes the model validation process long (since lower-level analog design tends to keep changing) and makes the DMS models less re-usable as they become so chip specific. In other words, the time a DMS model saves in simulator can be traded off by the time to produce it. AMS models, though slower in simulation, can be produced faster with less validation effort. In our case, the analog design team has already developed a full set of AMS models, some are high level, but many are detailed models that have been used in the design phase. The available AMS models allow us to allocate our limited MSDV resources to only develop key DMS models for critical blocks or for general functions that must have to support digital design and digital verification. In Table I, we present how we accomplish the MSDV task through a hybrid use of AMS and DMS models.

TABLE I
AMS AND DMS MODELS: QUALITY AND APPLICATION

Blocks	Description	DMS vs. AMS	Application
Padring	I/O pads and ESD cells	Same quality, fully modeled as in AMS	Digital I/O pads are in Verilog models from vendor, used by both AMS and DMS.
Reference, Level shifters. Crystal oscillator driver	Bias voltage and current generation, level shifting, supply monitoring (UVLO), crystal oscillation driver or external clock selection.	Same quality, fully modeled as in AMS	SVRNM models can be quickly developed for functionality and connectivity check.
Digital_Powergen	An LDO for all initial power up digital supply and a power-on-reset (POR) generator, plus a more complicated charge-pump (CP) regulator.	DMS has a more detailed modeling for the LDO, but it is left for AMS to model those more complex charge pump mode regulator functions.	A detailed digital power supply generation from the LDO, together with the power-on-reset process are critical for digital DV's PMU verification and UPF based low power flow.

PLL	A completely new design with integer and fractional mode.	Same quality, fully modeled as in AMS	PLL has a significant digital design part that needs a full set of DMS model to facilitate digital design process and digital DV tasks.
Analog Front End (AFE)	A programmable amplifier (PGA) and a complex continuous sigma-delta ADC. Three AFE channels each consisting of a PGA and ADC	DMS only has a high-level abstract PGA-ADC models. A full set of AMS models (with multiple modeling levels can be selected) from analog design team are developed and used in design phase.	More than 70% of verification tasks are using AMS models in the AMS flow, these include performance metrics, detailed clocking and control sequences, verification of digital islands.
Analog Back End (ABE)	A class-D power amplifier that has a previous working silicon from a separate tape out.	DMS only has a model of the DAC block that interact with digital design. A full set of high quality of AMS models inherited from the previous project.	More than 90% of verification tasks are using AMS models with the AMS flow. As this block is from a silicon-proof design, digital verification requirement is largely reduced.

As illustrated in the Table I, we have cases where DMS models have more details than AMS to enable digital DV's PMU verification and UPF low power flow. More importantly, it is by leveraging the AMS models developed and used by the analog design team that enables us to bridge the gap for verification tasks such as verification of detailed analog behavior and performance metrics.

Case 1: ADC: An abstract one-level DMS model and a detailed low level AMS model

The ADC is a key design component of the project. It has two main part: a 2nd order continuous-time delta sigma modulator (block "adc_top" inside analog_top, referred as "ADC") and a digital decimator (inside digital_top). A set of detailed and low-level AMS models for the ADC have been developed by the analog team and used in their block level design and verification phase. Rather than attempting an equivalent detailed DMS model of the adc_top which would require dedicated MSDV resources spending a large portion of project time, we chose to adopt an abstract one-level ADC DMS models. This simple DMS model, though losing lots of details about clocking, peak detection, calibration etc., was good enough to enable the digital decimation block design and verification. Fig. 4 and 5 illustrate visual comparison between DMS and AMS for ADC output and FFT performance, respectively.

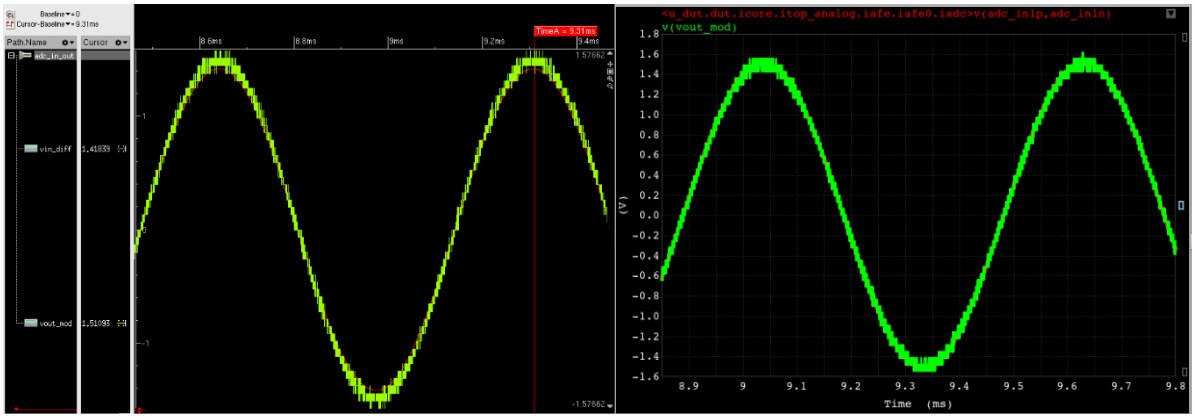

Figure 4. Visual comparison of ADC modulated output versus input from ADC DMS model (left) and AMS (right). The red trace is the differential input to ADC, and the green trace is ADC 6-bit output after a test bench DAC.

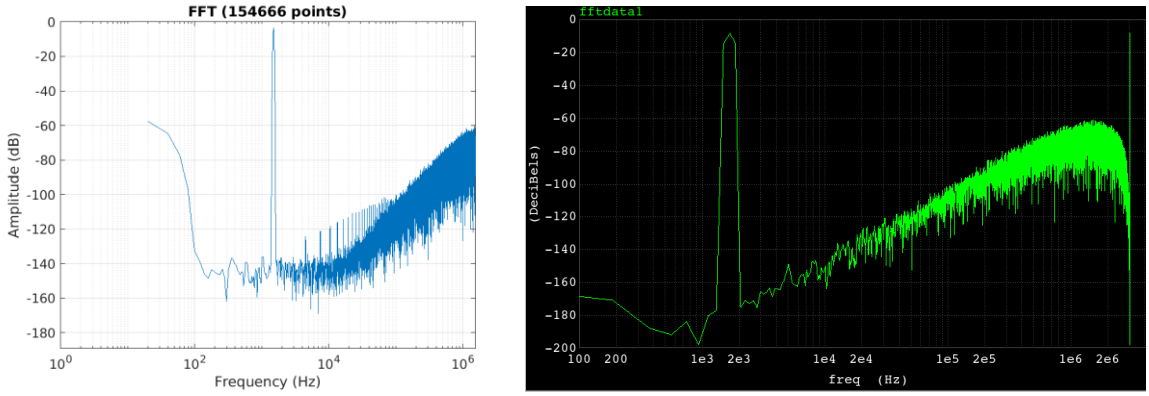

Figure 5. Visual comparison of ADC FFT performance results from DMS model (left) and AMS model (right).

Case 2: Class-D amplifier: A partial DMS model and a low level detailed AMS model

The last stage of the analog_top is a class-D amplifier that has 3 main blocks: current DAC, sigma delta class-D modulator and power stage. The block is simply a re-use of a silicon proofed separate chip. Because the big-A small-D nature of that chip design, its verification was accomplished solely by the AMS flow. That is, no DMS models were developed, only a full set of high quality AMS modes available. Because only minimal digital design change is needed for this block, rather than developing a full set of new DMS models, we chose to only spend time on a DMS model of the first stage, the current DAC to support necessary digital design and digital verification. We counted on the full set of AMS models from previous project to verify analog behavior and performance of the whole class-D block.

III. UVM: AMS AND DMS UNIFIED UNDER ONE INFRASTRUCTURE

The existence and necessity of two flows – AMS and DMS for MSDV, do not suggest two separate and independent flows to develop and maintain. In fact, testbenches based on Universal Verification Methodology (UVM) can very well unify the two flows in one infrastructure. While some overhead and extra setup from DV's UVM environment might have some impact on AMS compilation and run time, in general however, there is a huge advantage by re-using digital testbench and running AMS tests in the same platform as in DMS. For examples:

i) With the same input clocks, drivers, and environment setup, the two flows can be combined to accomplish the verification closure through a unified flow for test plan, test development, regression maintenance, and coverage analysis.

ii) AMS flow can benefit from sharing DMS's assertions, auto-checking, randomization, and different test scenario. DMS can benefit from AMS thanks to different model quality and different level of abstraction. When verification process goes wrong, two flows under one infrastructure helps to quickly identify if there is a design issue (a bug), or a model issue, or a test scenario issue.

Fig. 6 is a high-level abstract illustration of our UVM testbench for both AMS and DMS. Without a detailed description, it is basically an ADI customized Cadence Metric Driven Verification platform. The central UVM testbench is the "chip_top_tb_harness.sv" that can communicate both DMS and AMS netlists. DMS flow is the default flow, in which the SystemVerilog netlist of the chip is used by the digital simulator. To run the AMS flow, a compiler switch is defined to change to use the SPICE netlist of the chip. In AMS flow, an ADI simulator is running with either AMS models or transistor level schematics for analog blocks, and RTL or gate level for digital blocks.

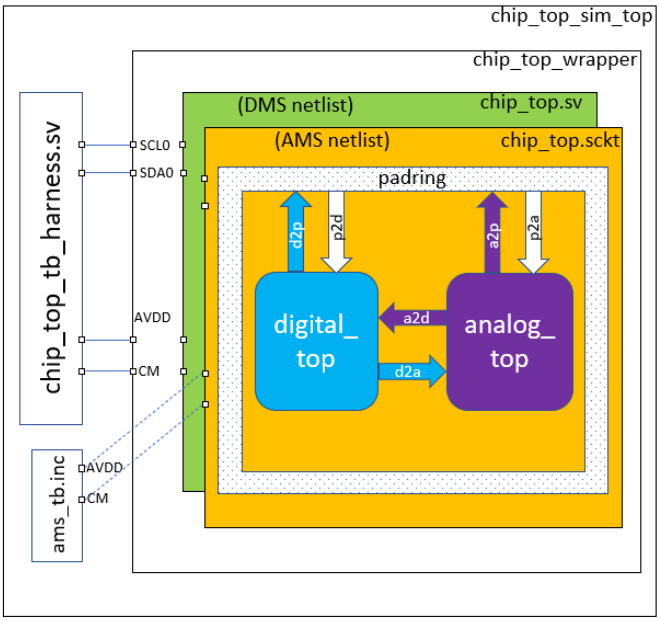

Figure 6. AMS and DMS two flows under one UVM infrastructure: abstract conceptual illustration. The DMS DUT (green block) is a netlist of high-level blocks with DMS models, and the AMS DUT (orange block) is a netlist of lower-level blocks with AMS models. The DUT (chip_top) can be partitioned to 3 main blocks: padring (white), digital_top (blue) and analog_top (purple). The MSDV activity and responsibility are mainly for verification of analog_top inside and its interface with the digital_top (a2d, d2a), and interface with the padring (a2p, p2a). The box "chip_top_tb_harness.sv" on the left is the central UVM testbench that is shared by both DMS and AMS flow. Additionally, AMS flow can have its own analog inputs (AVDD) and any necessary load testbench part (e.g., some external capacitor loads for the common voltage (CM) pin). A wrapper layer (chip_top_wrapper.sv for DMS and chip_top_wrapper.sckt) is used to enable accommodating some transformation or conversion needed between SystemVerilog and SPICE (e.g. a wreal input that by default is voltage from SV to convert to an input current source). The top cell is chip_top_sim_top.sv for DMS and chip_top_sim_top.ckt for AMS.

IV. Results and Discussions: Where AMS and DMS Differ in Their Capabilities

In this section, we present our detailed results based on analysis on how each flow contributes to the full chip verification coverage. We will show that many high-level functionality verification and analog-digital interface connectivity check can be fulfilled by the DMS flow. However, there is still a large portion of verification list that must count on AMS flow. This is especially true in verifying performance metrics at full chip level for the complex AFE and ABE design.

TABLE III
Analysis of The Roles of AMS and DMS Flows

Tasks	Details	AMS versus DMS	Flow Choice
UVM testbench bring up with the first A-D integration	Turn on full chip verification testbench with the first netlist.	DMS has the advantage due to its digital high-level modeling approach. AMS is slow, having more potential issues such as convergence.	Use DMS because more relevant and efficient
Low-power simulation (LPS)	Design needs to run LPS with UPF.	DMS is more relevant with LPS flow. LPS flow in AMS often encounter additional issues.	
Gate level simulation (GLS)	Design runs functional verification with gate level for top_digital.	GLS takes much longer time than RTL tests. AMS slow speed limits its use in GLS.	
Design for Test (DFT)	DFT tests (scan and MBIST) run with RTL and GLS.	AMS may be needed as a "double check" due to DMS model in-completeness, but mainly a DMS task.	
A-D interface with Formal Verification	Interface between Analog and Digital at high level (pin source to pin destination), reset values, and DFT scan mode values.	A-D interface definition from analog design team. Mainly a digital verification task	

Connectivity check	Common connectivity issues include pin mismatch, signal polarity mistakes, missing or incorrect connections, multi-driver contentions.	Only requires high level models so DMS is faster and more efficient in general.	Combination of AMS and DMS depending on model availability and quality
Functional verification of design units with A-D interactions.	Example design units: power management unit, clock and reset, oscillator, bias and reference, level shifter.	AMS models can handle more details; DMS models can also achieve the same quality of AMS models due to relatively simple analog circuitry.	
Analog detailed behavior hard to capture or model in digital way	Design units such as power amplifier, continuous time modulator, analog filter.	AMS models can be developed fast and accurately. DMS modes possible but error prone and requiring extra effort to validate.	
Analog performance measurement	ADC modulator and class-D amplifier requires full chip performance verification.	AMS flow has the advantage. DMS flow would need high quality of DMS models that usually is time consuming to develop and validate.	
Register map coverage	Coverage of all registers in the design.	Some registers correspond to detailed analog functions or analog performance can only be covered by AMS. DMS however can cover most registers that have direct connectivity and function impact on analog design thanks to its speed.	
Multiple power supplies or multi-level voltage domains verification	Design has multiple voltage levels (AVDD 1.8V, 1.1V and digital power supply 0.9/1.1V)	AMS flow found several critical design bugs due to voltage level mistakes. DMS flow is not applicable even though real valued powers and grounds in the model.	Use AMS as electrical simulation required
Device checking or safe operation area (SOA) check	Device level over/under voltage and over current density checking	AMS only as it requires transistor-level simulation.	
Full chip transistor-level sanity check test	Full chip power up sequence, state machine, power-on-reset, and PLL locking.	AMS models replaced by transistor-level schematics, RTL for top_digtal. This is classical AMS verification task.	

V. MSDV: A Specialized Discipline Opens a New Career Path

Being big-A(MS) small-D(MS), or big-D small-A, the fact that both AMS and DMS are needed in mixed-signal chip verification suggests that a team of special work force that is capable of and responsible for the AMS and DMS workflows is a prerequisite of success. While design verification as a discipline or design verification engineer as a career path has been a standard practice in IC industry for more than decades, it is only until recent years that MSDV has been identified as a separate and dedicated discipline and MSDV engineer as an emerging career path.

An MSDV engineer focuses on verification of analog design especially the analog-digital interfacing and interactions. As such, a good knowledge and understanding of analog circuitry and skill set to analyze analog behavior (e.g., DC and transient schematic simulation) are some minimum prerequisites for this career path. It is important to note that, where DMS flows or DMS models are pure digital, the process to develop DMS models involves AMS flow in nature. One may argue that a "DMS engineer" can code DMS models for analog design blocks solely based on design specification. However, this type of specification-based models is often abstract, at high level, and missing implementation details to verify analog behavior and functionality.

While individual practices may vary cross different companies or business units, we can still outline some common activities and responsibilities of an MSDV team.

1) *Full-chip verification testbench creation.* Work with digital DV team for the master UVM testbench and provide support for stimulus for the DUT at pin level. Create and maintain AMS part of UVM testbench and feedback to digital DV team for consolidation and collaboration between two flows.
2) *Modeling of analog circuits*. Develop and validate real number models by Verilog-AMS "wreal" or SystemVerilog real nettype for DMS flow. Integrate and modify Verilog-A(MS) or equivalent analog behavioral models from analog design team for AMS flow.
3) *Netlist release.* Create the first full chip netlist and turn on the verification test flows. Maintain new netlist release process when design changes, this often requires model updates.
4) *Test development*. Develop DMS tests with focus on analog-digital interface connectivity and main functionality of analog circuitry at block pin level. Develop AMS tests to verify detailed analog behavior (usually at lower hierarchical level), performance metrics, and analog test bus. Maintain regression and perform test coverage analysis.

VI. Conclusions: Leveraging the Best of AMS and DMS for MSDV

In this paper, we suggest that the popular industry adoption of DMS flow for the full chip functional verification does not imply that it is a full replacement of the AMS flow. Unlike the early years of mixed-signal verification when AMS was the only dominant flow, we believe that DMS flow is unlikely to become the sole flow in MSDV, perhaps both AMS and DMS will co-exist for the foreseeable future. At least, AMS flows are needed for DMS model validation process and in full chip transistor level verification (electrical sanity check for basic power up process). There is a trade-off between the time DMS models can save in simulation and the time for the models to be produced. AMS models that are easier to develop or already available from an analog design team can compensate for the disadvantage of DMS models for the verification of detailed analog behavior and critical analog performance. The industry also needs to recognize that MSDV, comprising of AMS and DMS, is becoming an emerging separate discipline that calls for well-trained engineers who not only have solid understanding of analog design, but also have good experience in the state-of-art digital verification methodologies.

Acknowledgment

The authors would like to thank Daniel Canniff and Atsushi Matamura from analog design team for their work on AMS modeling and AMS test development for ADC and ABE blocks, Annapurna Itagi from Exiger Technologies Pvt Ltd for developing and maintaining the DMS flow, Timothy Caputo for the UVM testbench creation and leading the whole DV activity, Nimay Shah from ADI CAD team for the guidance and support in MSDV flow development and improvement.

References

[1] D. Juneja, S. Prabhu, S Veluri, “Practical considerations for real valued modeling of high performance analog systems”, *Proceedings of DVCON*, 2016.

[2] J. Dudek, J. Nekl, K. O’Donoghue, “Real number modeling of RF circuits”, *Proceedings of DVCON*, 2017.

[3] S. Thykkoottathil, J. Dudek, N. Ranganath, N. Shah, S. Singh, “Taking real-value modeling to the next level: power-aware verification of mixed-signal designs”, *Proceedings of DVCON*, 2019.

[4] R. S. Vogelsong, “AMS Behavioral Modeling”, in *Mixed-Signal Methodology Guide*, J. Chen, M. Henrie, M. F. Mar and M. Nizic, Cadence Design Systems, Inc., San Jose, CA, August 2012, pp 25-70.