

# Taking Real-Value Modeling to the next level: Power-aware verification of mixed-signal designs

Subin Thykkootathil  
ADI, Bengaluru

[Subin.Thykkootathil@analog.com](mailto:Subin.Thykkootathil@analog.com)

Nimay Shah  
ADI, Wilmington  
[Nimay.Shah@analog.com](mailto:Nimay.Shah@analog.com)

Jakub Dudek  
ADI, Wilmington

[Jakub.Dudek@analog.com](mailto:Jakub.Dudek@analog.com)

Nagesh Ranganath  
ADI, Bengaluru

[Nagesh.Ranganath@analog.com](mailto:Nagesh.Ranganath@analog.com)

Santosh Singh  
ADI, Bengaluru  
[Santosh.Singh@analog.com](mailto:Santosh.Singh@analog.com)

## *Abstract*

**Every new generation of electronic devices requires higher performance and more sophisticated power management. This poses an exponentially growing verification challenge, especially in the mixed-signal domain. This paper discusses the strategies that we deployed to counter this challenge. Specifically, the focus will be on the techniques in the areas of mixed-signal verification infrastructure that can translate into verification of any mixed-signal design.**

## I. INTRODUCTION

SoC designs use various power reduction techniques to meet power budget requirements. The low power architecture divides the design into multiple power domains and defines valid power modes. Implementation tools insert the necessary power switches, isolation, retention and level shifter cells at appropriate places as specified by the CPF/UPF. Apart from the typical power domain using the CPF flow, we have Analog Power Gate Switches and power domains based on LDO (Low dropout) regulators. As the number of power domains increase, power management becomes complicated and its verification, even more challenging. An important aspect of low-power verification is to verify the interactions of analog blocks such as LDOs, Analog Power Gate Switches, Power-on circuitry, etc. with the digital logic. These mixed-signal interactions and connectivity of control and status signals are typically verified in analog-digital co-simulations, involving both analog and digital simulators. Simulation performance is a big limitation when it comes to co-simulations, severely increasing the turn-around time of simulations. Furthermore, the co-simulation environment will not be ready early in the design development. This compelled us to investigate into creating a SystemVerilog power-aware mixed-signal simulation environment that requires a single SV simulator.

In this paper, the verification challenges around mixed signal low power SoCs will be discussed. The traditional approach and its drawbacks will be mentioned, before moving on to the proposed method. With the proposed methodology, real number models, power awareness, model validation and testbench components will be explored. A conclusion and final remarks will follow.

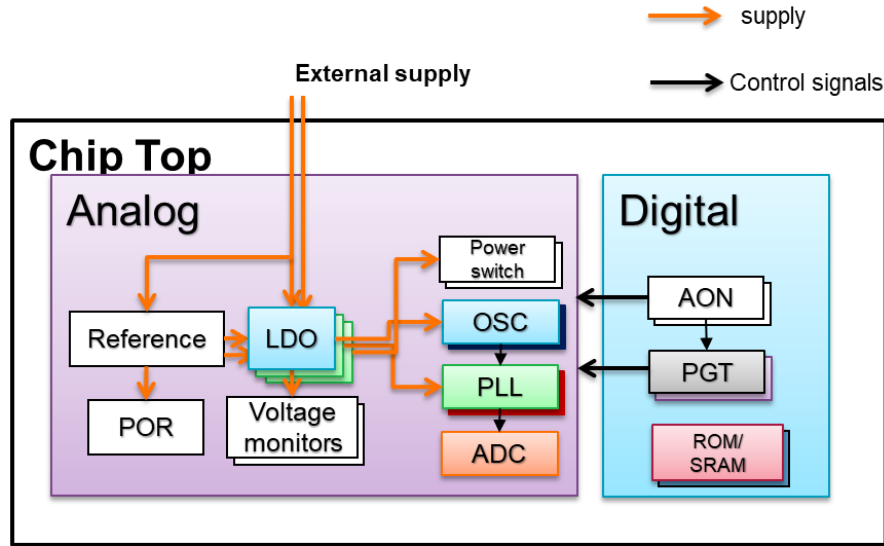


Figure 1: Simplified block diagram

## II. VERIFICATION CHALLENGES

### A. Mixed-Signal Interactions

Control and trim signals of analog circuits like references, LDO, voltage monitors, analog power switches, POR circuitry and clock sources are typically controlled from digital memory mapped registers and state machines. Several critical signals like clocks, failsafe and brownout resets, oscillator lock and other status signals come from analog to digital logic. Hence there are many critical signals crossing the analog-digital boundary.

Isolation/Clamp values of these signals during power up and other power modes must be carefully selected and verified. For example, if the trim value to the reference block is incorrect during power-up, the design will not power-up correctly. Since the supply to the digital is not available during power-up, these control signals must come from isolation cells. This is shown in Figure 2. Trim value for the reference block (`ref_bg_vtrim`) is generated from a power gated block `pmuPGT`. During powerup this signal is isolated to `5'h10`.



Figure 2: Power up isolation

### B. Complex Power Mode Transitions

Figure 3 shows the power supply architecture of the design. There are multiple external supplies driving the part. The design needs to support ramping up of the supplies in any order. For example, external supply VDDIO could ramp up to the expected level before external supply VDDL. The design can also power-up without all the external supplies being available. This can lead to valid power-up scenarios due to various combinations of the order in which the external supplies get switched on.

Beyond the power up sequence, there are multiple LDOs in the design, as shown in Figure 3. Blocks that are powered up by an LDO are grouped together into a separate power domain and the LDO acts as a power switch to the corresponding power domain. Isolation between these power domains needs to be controlled based on the LDO state. De-assertion of the isolation control signal depends on the power-up time of the LDO and after LDO supply is stable.

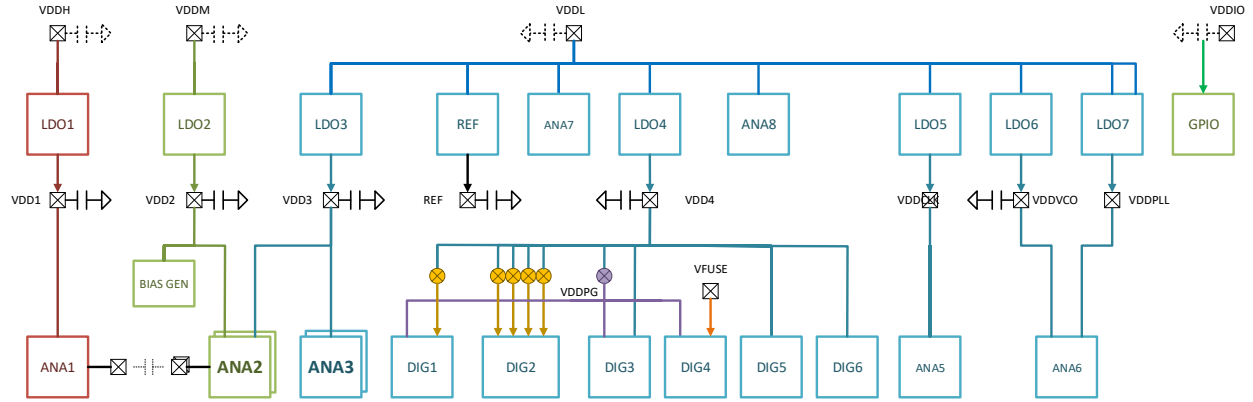


Figure 3: Power Architecture

The number of power mode transitions is large as any combination of turned on LDOs is essentially an individual power domain. The number of possible crossings between power domains thus grows quickly. A first pass exercise consists of establishing which crossings are valid and which are not expected in normal operation. Hence the scope for bugs and corner cases going unverified is ever present.

### III. TRADITIONAL APPROACH

Given the complexity of power domain profile and the number of critical mixed-signal interactions, it is highly important to verify analog and digital together and across all possible scenarios.

Traditional testbench drivers and monitors to mimic the analog should be avoided for several reasons:

1. They are a very crude approximation of the analog circuitry.

Main intention of the model will be to get the powerup and power modes working with minimal level of modelling. Mostly it used to sequences of analog control signals with #delays in between.

2. They do not respect the SoC connectivity.  
Analog design hierarchy and control signals between them were ignored.
3. Power intent in test bench component is not easily definable.

Since there isn't any CPF for the analog blocks, power intent of these blocks were ignored.

Traditionally, a mixture of simple non-pin compatible Verilog model and class based test bench components have been used to close on power management verification. This has typically been good enough for verification of the digital state machine and some level of A/D connectivity. Signoff quality verification is performed at the Analog Mixed Signal (AMS) verification level, where RTL and analog circuit models are co-simulated with tools such as Cadence® MSDV or other proprietary mixed signal simulators. The AMS co-simulations suffer from late readiness. Analog models are typically provided by analog designers who create models based on finished schematics. Pulling together the full environment, including resolving partitioning and convergence issues is performed at a late stage in the design cycle. Power management verification becomes the long pole in the project's schedule and bugs are found late. Furthermore, coverage closure of the digital circuits can only be achieved after merging coverage between digital only (DMS) simulation and analog co-simulation. Merges are traditionally error prone and time consuming and add an extra step in what is already the long pole in the schedule.

Issues of mixed signal nature found in power management verification often fall into the following categories:

- Wrong isolation values leading to POR failure or wake from sleep failures
- Wrong sequencing of analog blocks

These types of issues have the potential of delaying tapeout and have traditionally done so. They have the potential of requiring analog design changes, with impact on layout and implementation which can prove costly to a tight schedule.

#### IV. METHODS ADOPTED

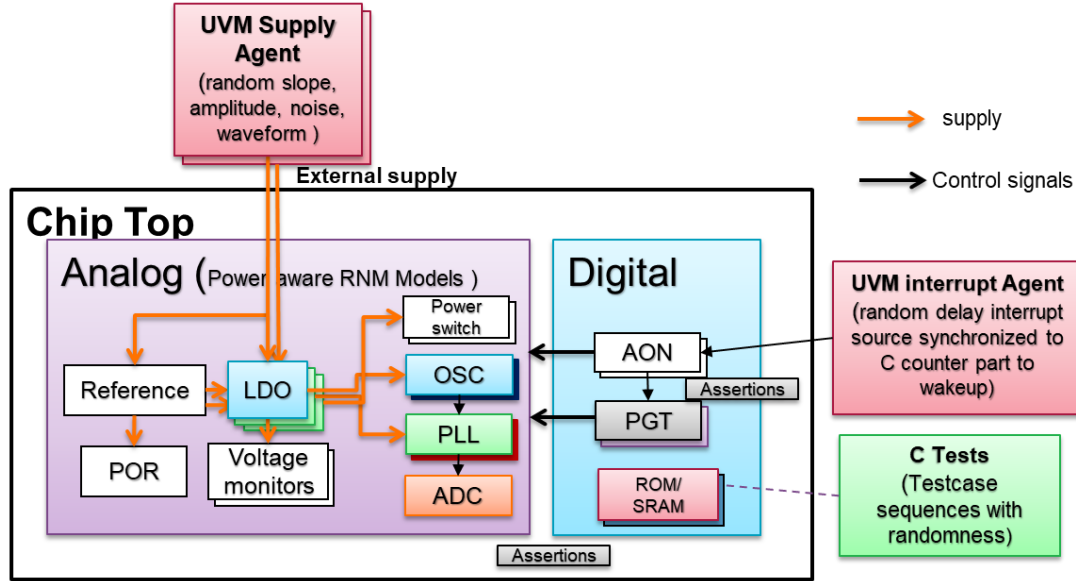


Figure 4: Verification Environment

Figure 4 shows a single simulator DV environment for our mixed-signal SoC design. The following subsections discuss the techniques adopted to verify the power management unit and how the challenge of verifying analog-digital interactions early in the verification cycle was addressed.

##### A. Real Number modeling

SystemVerilog real number modeling has been used for many generations of IC design to quickly and efficiently model analog circuits using the SystemVerilog language. Emphasis is put on functionality and connectivity while performance and accuracy are secondary goals. The SystemVerilog real type is used to represent real number quantities such as voltages and currents. Analog transfer functions can be modeled with the use of oversampling (to approximate continuous time) and mathematical functions such as additions and multiplication. Figure 5 shows a schematic of the LDOCORE. The following code snippet shows how the behavior of LDO(ldo\_hp\_capped\_core) is captured. LDO output voltage is modeled as a real number.

```
...
    real out_int;
    assign out_int = limit(out_int + I_ldo * step_size/C1, .min(0.0), .max(vref));
...

function real limit (input real in, input real min, input real max);
    limit = (in > max) ? max : (in < min) ? min : in;
endfunction
```

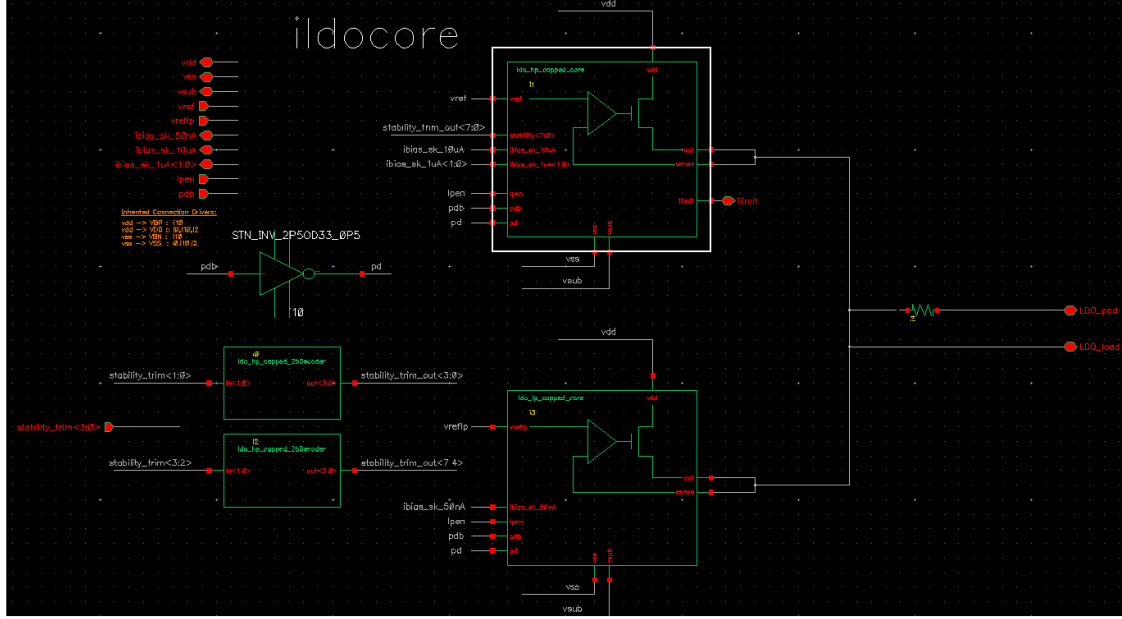


Figure 5: Ildocore schematic

Figure 6 shows an example of real number signaling in a digital simulator.

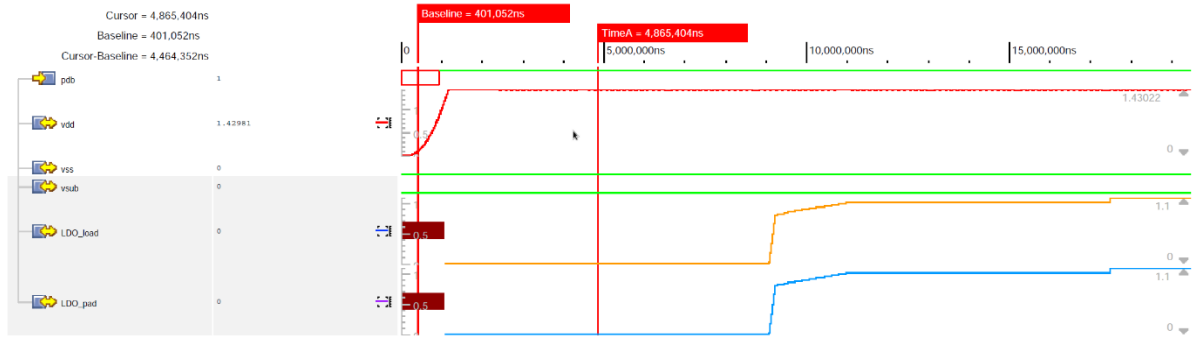


Figure 6: Inputs and Outputs of Ildocore's Real Number Model

### B. Power-Awareness:

While the power behavior of digital blocks is handled in standard low power flows (CPF/UPF), power awareness for analog blocks is coded directly in their behavioral models. On the source side, LDOs must generate the voltage models with correct levels and timing.

```
...
assign out_int = limit(out_int + I_ldo * step_size/C1, .min(0.0), .max(vref));

assign out = (valid_inputs && bias_good && pwr_good)
? ( (pdb === 1'b1)? out_int : ( (pdb === 1'b0) ? `wrealZState : `wrealXState))
: `wrealXState;

...
function real limit (input real in, input real min, input real max);
    limit = (in > max) ? max : (in < min) ? min : in;
endfunction
```

On the sink side, all blocks need to respond to a lack of correct power levels on their power and ground ports. To that effect, real number models can be enhanced with simple behavioral power awareness. Power ports are checked against expected values within some tolerance, as shown in the following code snippet.

---

```

...
inout vdd;
cds_rnm_pkg::wrealsum vdd;

always @(*)
begin
    if ((vdd != `wrealXState && vdd != `wrealZState) && (vdd >= vddl_min) && (vdd <= vddl_max)
    && !vss && !vsub )
        pwr_good <= 1'b1;
    else
        pwr_good <= 1'b0;
    end
end
end

```

---

The models propagate ‘x’ (or a cadence real package `wrealXState) on outputs when the supply voltage is outside the normal operating range. Care must be taken for x propagation to be as pessimistic as possible to aid in finding issues. Not only does invalid power generate x outputs, but invalid inputs or control signals do as well. This ensures that invalid states propagate properly through the chain, even if the power status of some downstream block is locally correct. Figure 7 illustrates this concept.

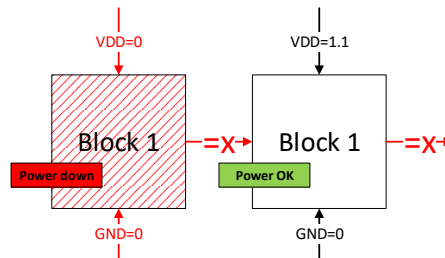


Figure 7: X propagation

The interactions of these analog blocks with digital logic are also captured in these models. For example, voltage monitors must respond to supply voltage levels to generate interrupts or resets to the digital core. It is also worth noting that many functional parts of the analog design are constructed with standard cells, which require no modeling at all (Figure 8).

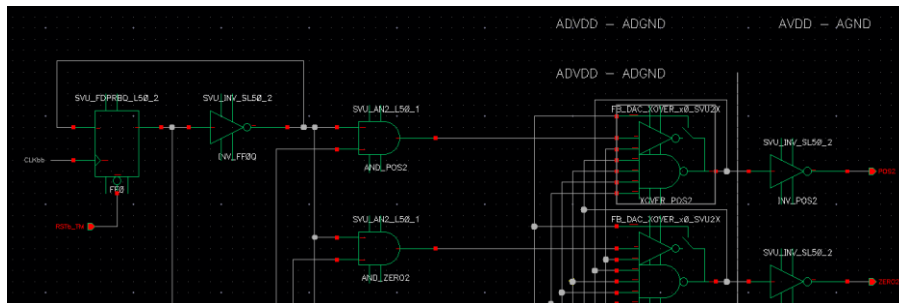


Figure 8: Standard Cell Netlisting, no modeling code required

Standard cells are extracted from their schematics with power connections and power-aware libraries are used. The power network is thus entirely extracted from the schematics and the models must drive power domains appropriately for the device to power up, sleep, hibernate, and operate correctly in active mode. Netlisting is described in greater detail in section C.

### C. Analog Design Environment Customizations and Analog Top Netlisting

All the SystemVerilog behavioral models are maintained as text views alongside the actual schematics in the analog design environment. This method presents two key benefits. First, it ensures pin compatibility between the schematic and the behavioral model by triggering a custom check anytime either is modified. Secondly, it uses the same source database for all forms of analog top netlist generation, thereby ensuring correct connectivity, be it a spice netlist for AMS simulations, a CDL netlist for physical verification, or, a SystemVerilog netlist for DMS simulations. The Cadence® nettype package and other custom-built libraries are also included in the Cadence® Virtuoso® environment as libraries in order to ensure correct by construction SystemVerilog behavioral text views.

For SystemVerilog netlisting of the analog top, Virtuoso®'s SystemVerilog netlister plugin is used with several customizations. One such beneficial customization is the ability netlist standard cells inside digital islands with full power/ground connectivity from the onset. This is achieved through a combination of configuring the analog top hierarchy so as to netlist standard cells as schematics but at the same time customizing the netlister such that the schematic view was a 'stop' view for the standard cells. A Netlisting wrapper and custom script is also used to help integration in the verification regression flow. They are used together to generate a view of the chip with automated digital and analog connectivity.

### D. Model Validation

As described in [1], model validation can take multiple forms depending on intended use of the model. For SoC functional verification, the validation requirements focus on correct controllability and timing. While performance can be included in signal processing blocks, it adds little value to power management blocks like LDOs and Voltage monitors. Performance of such blocks are described by voltage ripples, supply rejection, current consumption, etc. Such circuit characteristics are not useful in trying to identify functional bugs and are verified and characterized by analog designers.

Models can be validated by comparison with transistor level simulations or low level AMS models. Analog Devices uses a proprietary mixed signal language used to model analog circuits. Model comparison can be established in an integrated testbench and model output compared within agreed upon tolerances. Figure 9 shows a block diagram of such a testbench. Figure 10 shows a waveform comparison between an analog simulation and the RNM of an LDO.

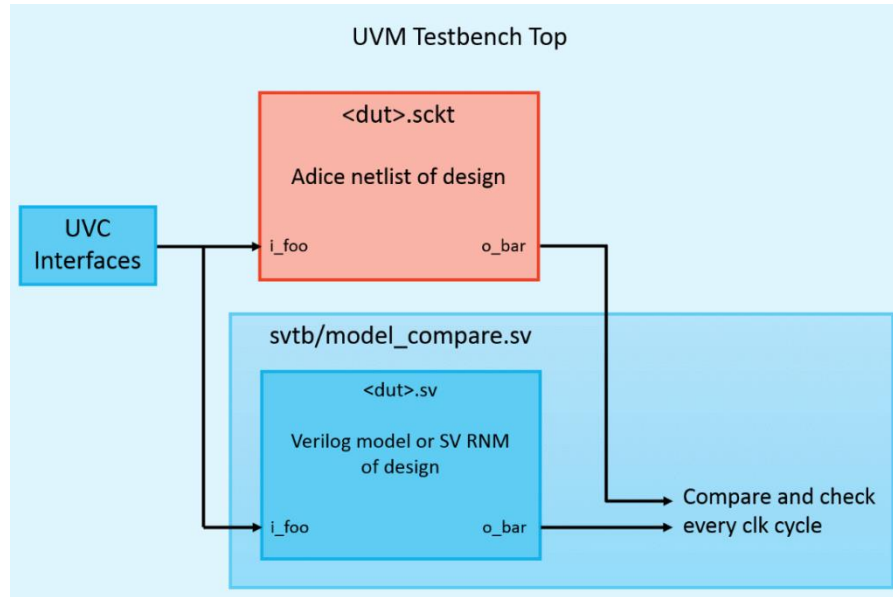


Figure 9: Validation testbench

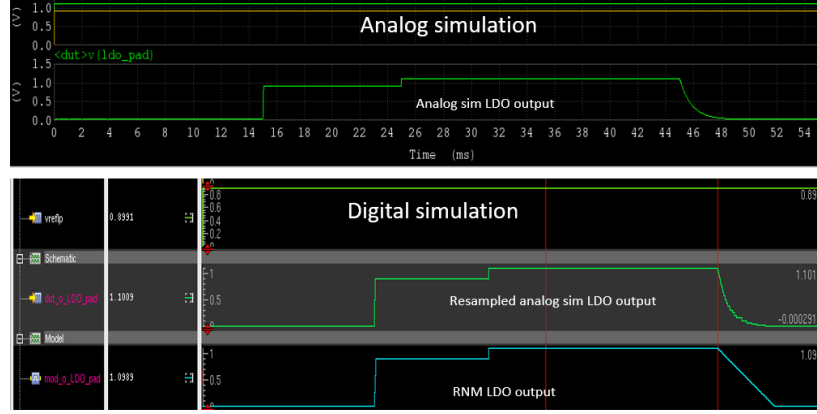


Figure 10: Waveform comparison

A different validation technique consists in validating the spec against the analog specification. The advantage here is very fast simulation time (no analog simulations) and the ability to cover many scenarios as per a pre-defined model test plan. Model code coverage can also be collected. An LDO can be tested against the expected polarity of its control signals, the output voltage and the settling time. Under the conditions where all I/Os are modeled and tested against a spec, the model can be considered valid. With the use of a proper test plan, regression testing and coverage metrics, standard metric driven verification can be applied to a real number model. Metric driven validation is beyond the scope of this paper but will be covered in future work.

#### E. Integrating power-aware analog RNM with digital RTL for low power simulations:

Utilization of core DMS technology of datatype coercion helped us in connecting SV models to the rest of design and test bench without worrying about the interconnects and their types. Coercion is the process by which a Verilog wire can assume a different datatype because of its hierarchical connections. Thus, when a wire/ interconnects is connected to a net of the type wreal, SV real, or VHDL real, it is coerced to wreal as well. Such coercion can occur across multiple hierarchical levels. To enable seamless connection of the standard cells with power/ground connections in analog top Verilog netlist, automatic insertion of bi-directional Real-to-Logic/Logic-to-Real Connect modules with an 'amsd' configuration block is used.

On the digital side, the well-established CPF/UPF flow defines the power awareness of RTL. To close the loop with models such as LDOs and power switches, the real voltage output of the models (for example, an LDO output voltage) is used as shutoff condition for the power domain of a power gated RTL.

#### F. Driving the Voltage Supply from Testbench:

Sections A through D elaborated the process of creating and validating power-aware real number models for analog blocks. The next step is to drive power supply inputs of the chip. As a first order approximation, logic signals can be used for driving power supply inputs. However, to utilize the capabilities of real-value models and to generate more interesting and realistic verification scenarios, real number power supplies are needed. Hence, a UVM-based real number supply agent was developed in order to drive external power supply ports, as shown in Figure 4. The supply agent had the ability to randomize timings, ramp rates, supply noise and other parameters of interest and it lends itself to seamless integration with the rest of the UVM test bench. Supply agent enables simulation of different power-up scenarios, enabling power-up sequences to be a part of constrained-random test cases. As shown in Figure 11, it also provides the ability to create supply glitches, overvoltage, and under voltage scenarios. Voltage monitors can respond to real voltage levels instead of binary on/off signals.



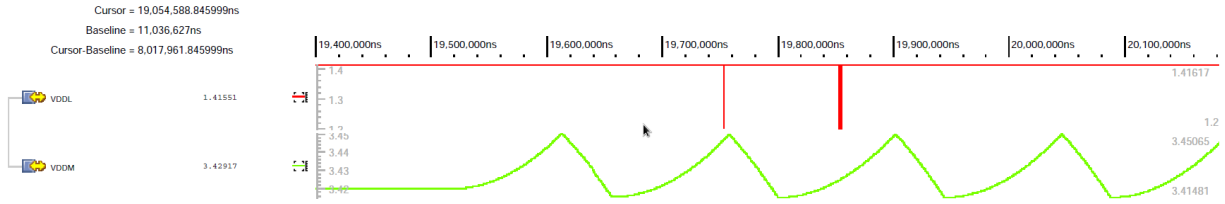


Figure 11: Supply agent generating glitches, over-voltage and under-voltage scenarios.

### G. Checkers at Analog-Digital Boundary:

All the above sections described how the proposed single-simulator verification environment was able to drive “real” stimulus to various analog blocks and digital logic using supply agent and real-value models. Complementing this, assertions were developed to verify the behavior of signals crossing the analog-digital boundary. The specification included an excel sheet that contained reset and isolation values of all the signals crossing analog-digital boundary. A PERL script was written to generate assertions from the excel sheet. These assertions (POR and ISO) compare the logical value of signals during reset and isolation with their expected values from the spec.

Reset check for a signal triggers immediately after the corresponding reset is released. Figure 12 shows the sequence of release of different resets that we had in our design and the waveforms of assertions triggering at each of the resets.

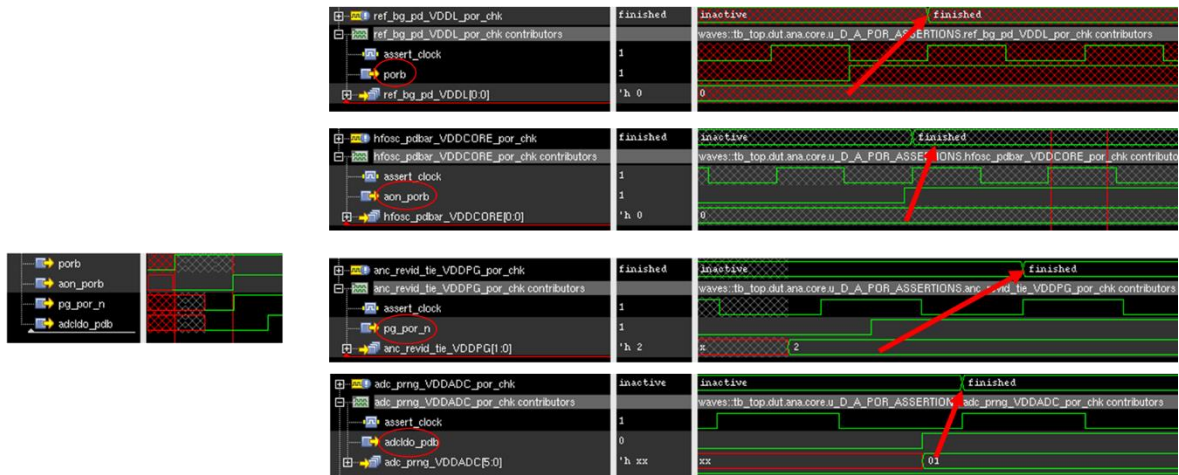
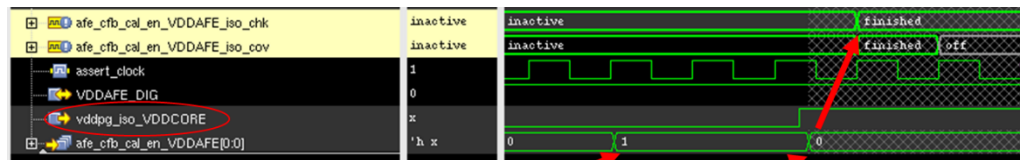


Figure 12: Reset Assertions

According to design specification, when the isolation for a signal is active, the signal may be isolated to a fixed value (0/1) or the signal may hold its value during isolation. Accordingly, depending on the isolation type, PERL script generated 2 types of assertions: ISO check and ISO\_HOLD check



Signal being forced      Signal getting isolated to 0

Figure 13: Assertion for a signal that is isolated to 0

Figure 13Figure 12 shows an example for ISO check where the signal is isolated to 0. The assertion triggers when the corresponding isolation control signal is asserted. For the sake of verification, we forced the signal to a different value and the assertion checks if signal is driven to expected value during isolation.

For a signal whose value is hold during isolation, there were two checks: ISO\_HOLD and ISO\_HOLD\_POR. ISO\_HOLD\_POR assertion checks the reset value of the signal with the isolation value from the spec. ISO\_HOLD assertion checks that the value of the signal is held. Figure 14 shows an example for the above checks.



Figure 14: Assertions for a signal whose value is hold during isolation

These script-generated assertions ensured that the Analog to Digital boundary is thoroughly verified during reset and power state transitions.

Adopting the above mentioned methodology helped to identify many wrong isolations, during the simulation bringup stage itself. For example, the isolation value of the trim to reference block was wrong and was leading to a powerup failure. Power mangament verification closure well before 3 month to tapeout and zero bug caught in cosimulation flow and in silicon was the testimonials to the methods adopted.

## V. CONCLUSION

The paper presented a case for using real-value modeling for low-power verification of mixed-signal designs. It explained how to develop and integrate “Power-Aware” real value models of analog blocks into the SystemVerilog UVM test bench. Constrained randomization of power signals using a supply agent UVC allows for thorough exploration of external power supply state space. Auto-generated assertions played their part in ensuring that the A-D interface is verified. The ability to expose critical issues related to the power up sequences, missing and incorrect isolations and reset values of digital signals became apparent early in the design and verification cycle. Code coverage closure for RTL logic that is controlling certain analog blocks can now, be achieved without the use of a co-simulation environment. A SystemVerilog only environment allows for quick and exhaustive verification of power management unit.

## VI. ACKNOWLEDGEMENTS

## VII. REFERENCES

- [1] J. Dudek, J. Nekl and K. O’Donoghue, "Real Number Modeling of RF Circuits", DVCON 2017, USA.