International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 6509-6517
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Reusable UVM Architectures for Mixed-Signal Designs

Aparna Mohan*

North Carolina State University, Raleigh, North Carolina
* **Corresponding Author Email:** aparn2a@gmail.com- **ORCID:** 0009-0008-1422-700X

Article Info:

DOI: 10.22399/ijcesen.3761
Received : 04 February 2025
Accepted : 28 March 2025

Keywords

UVM
Mixed-Signal Verification
Reuse
AMS VIP
Analog Modeling
Verification IP

Abstract:

Contemporary mixed-signal System-on-Chip (SoC) solutions comprise more and more involved analog and digital modules. Although Universal Verification Methodology (UVM) testbenches have now become de facto industry standard for reusable digital verification IP (VIP), their reusability in mixed-signal environments is not yet fully explored. Conventional AMS verification flows are substantially dependent upon custom analog testbenches, in many instances leading to duplication of effort and atomistic approaches in work-flows within design teams. This paper presents a reusable UVM architecture that is organized into a structure to be used to a mixed-signal design. Incorporation of real-number modeling, uniform interface adapters and strong transaction-level stimulus extends a method between event-driven digital testbenches and continuous-time analog continuum. A case file is introduced on a reusable Analog-to-Digital Converter (ADC) verification IP, which manifestly exhibits gains in coverage, debug productivity and design cycle time. The methodology proposed is helpful to SoC teams so that aspects of faster closure with increased confidence can be made and at the same time provide standardization in reusable AMS VIP libraries that can be used in future designs.

1. Introduction

Modern electronic designs are dependent on mixed-signal technology - mobile SoCs, automotive radar or high-speed network transceivers. These systems integrate highly closely digital logic (state machines, controllers, digital filters) with sensitive and difficult-to-design-and-optimize analog building blocks such as ADCs, DACs, phase-locked loops (PLLs), and high-speed serializers/deserializers (SERDES). Design reuse and heavy duty verification are becoming mission-critical, as SoCs scales to billions of transistors, in meeting aggressive time to market targets (Bergeron, 2006; Kundert & Chang, 2004).

Within the flows that are digital-only, Universal Verification Methodology (UVM) has established itself as the gold standard in constructing re-useable, modular testbenches. To assure several IP blocks in a variety of projects, teams create reusable agents, drivers, monitors, virtual sequence, and scoreboards (Foster, 2015). But in cases when at least one block is analog, verification groups as a rule migrate to custom and ad-hoc analog simulation benches, using Verilog-AMS or SPICE-level descriptions. That would provide a reuse gap, with the UVM flow seldom extending into the continuous-time world.

It is thus evident why there is a need to have reusable mixed-signal UVM architectures:

The first is that the identical analog IP (an example would be 12-bit ADC) could be found in more than one SoC - consumer, industrial and even automotive, each iterating their AMS testbench independently and completely.

Next, strict time-to-market: because of a shortening development cycle, pre-silicon validation of corner cases is required, furthermore, requiring mature coverage of the space of functional corners that take years to fully explore using traditional full transistor-level simulation (Nair et al., 2013; Kundert & Chang, 2004).

In this paper the gaps are covered by:

1.Substance of proposing a standardized reusable UVM model of AMS blocks.

2.Illustrating the connection between real number model (RNM) and Verilog-AMS UVM agents.

3.An illustration of the flow using an actual ADC VIP case study.

4.Measuring ROI in Coverage and Verification Cycle time.

It is possible to use mixed-signal UVM architectures to bridge the digital and analogue verification teams by capturing best practices in reusable format, which can make functional sign-off quicker, cheaper, and stronger (Bergeron, 2006; Kundert & Chang, 2004; IEEE Std 1800-2017).

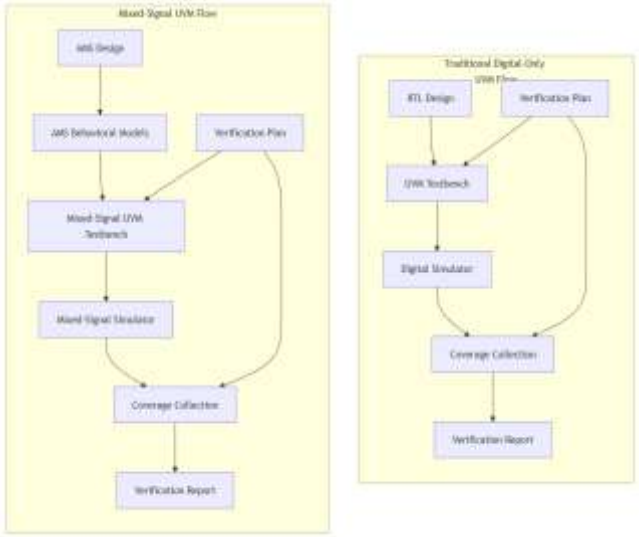


Figure 1. *Traditional Digital-Only UVM vs. Mixed-Signal UVM Flow*

2. Background and Related Work

2.1 Universal Verification Methodology (UVM)

The UVM is an IEEE standard (IEEE Std 1800-2017) that was created to render testbenches to be reusable, modular and scalable.
It defines:
The agents, which include sequencers, drivers, and monitors of a specific interface (e.g. AXI, I2C).
• Sequences, that produce stimulus at the transaction level.
Scoreboards, or checking of expected and actual behaviour.
• Coverage collectors that are used to record functional coverage to sign-off (Bergeron, 2006; Foster, 2015).
In pure digital SoC design, reuse is clear cut: when using AXI VIP or SPI as an agent, the agent may be plugged into any block which supports the standard protocol. These agents however make the assumption of discrete, event driven signals.

2.2 Mixed-Signal IP: Analog Building Blocks

The analogue behaviour commands important contribution to the performance of the system:
• ADCs/DACs: Translates digital codes and analogue waveform to digital and analogue.
• PLLs:They supply robust timekeepers.
• SERDES: Serialise and deserialise high speed I/O.
• Power Management ICs (PMIC): Control the voltages, and observe the analogue statuses (Kundert & Chang, 2004).
Traditional AMS verification is based upon:
• SPICE: Very accurate, transistors level simulations.
• Verilog-AMS: Speed-favouring behavioural models which model analogue waveforms as approximations.
• System Verilog RNM (Real Number Modelling): Employs real data types in modelling continuous signal in event-driven simulators (Nair et al., 2013).

Table 1

AMS IP Block	Key Analog Behavior	Typical Test Stimulus
ADC	Nonlinearity (INL/DNL), quantization	Ramp, sine, noise
DAC	Settling time, output swing	Step input, pattern codes
PLL	Jitter, lock time	Frequency steps, phase noise
SERDES	Eye diagram, equalization	PRBS, jitter injection

2.3 AMS Verification Approaches

Pure analogue: Whole SPICE level simulation is used to test transistor behavior via teams. Extremely precise, but slow to use--provided no value in corner-case sweeps or in checking a system (Kundert & Chang, 2004).
Mixed-signal co-simulation: Capacitates the combination of analogue solvers (continuous time) and event-driven digital engines. Verilog-AMS or RNMs simulate analogue blocks but the digital component operate in UVM. This co-simulation is offered through means of tools such as Cadence AMS Designer, Synopsys CustomSim, and Mentor Questa ADMS (Nair et al., 2013).
Hybrid flows: Sign-offs in some teams are a pure behavioural model and SPICE checks are only done in final sign-off. This scheme enjoys effective trade-offs between the speed and coverage (Nair et al., 2013; Bergeron, 2006).

2.4 Gaps in Reuse

The majority of the AMS flows are project specific:
• Hand written in each block is behavioural model.
• Interface adapters to UVM sequences are hand written.
• Checkers and scoreboards have not been standardized on analogue outputs (e.g. real-number signals).

This implies that even established corporations spend weeks of their time writing identical AMS VIP logic to every tape-out (Nair et al., 2013; Foster, 2015).
Industry trend: Recent studies and DAC tutorials demonstrate the initiative of the move to the standard AMS VIP libraries that:
Abstract RNMs behaviour.
• Connect into digital UVM agents.
• Share checkers, stimulus and coverage measurements across IPs (Venkataraman et al., 2018; Cadence, 2021).

2.5 Related Standards and Tools

IEEE Std 1800-2017: real-number data types and assertions are supported in the System Verilog language standard (IEEE Std 1800-2017).

• Accellera UVM Library: An implementation popularly used to control VIP in digital applications.
• Verilog-AMS Standard: Establishes a behavioural analogue modelling standard, which is well supported commercial tools (Kundert & Chang, 2004).
• AMS Designers & VCS AMS: The front-runners in the co-simulation environment that correlates analogue solvers with digital simulators (Cadence, 2021; Synopsys, 2020).

3. Challenges in Reusing UVM for Mixed-Signal

UVM has grown up as the superset standard of reusable digital verification IP, but its application on mixed-signal applications is not trivial. There are various problems that can be encountered when the digital reuse principles intersect with the analogy behaviours:

3.1 Analog Behavioural Modelling vs. Event-Driven Engines

Transaction-level modelling and event-driven simulation is the backbone of UVM. It is discrete signal and clocked event-assuming. By contrast, the analogue signals are by nature to be continuous time and either:
• The differential equations solvers (verilog-ams modules);
• Real-Number Models (RNM) that are approximated to continuous models with SystemVerilog real variables.
This generates mismatches of simulation:
Time-step integration is used in analogue solvers.
Digital engines are based on delta-cycle propagation of types of events.
The problem is to maintain such coordination of these domains without compromising accuracy, or imposing simulator performance bottlenecks (Kundert & Chang, 2004; Nair et al., 2013).

3.2 Co-Simulation Overhead

To run mixed-signal co-simulation it is necessary to launch both:
(e.g. Incisive, VCS, QuestaSim)
• An analog generator (e.g. Spectre, CustomSim, ADMS)
The cost may be quite high:
• There are time penalties to communications between analog and digital worlds.
Long analog settling times may make simulations slow.
Multi-viewers that support an integrated analog/digital path with waveform interactions are required to debug the interactions (Synopsys, 2020; Cadence, 2021).

3.3 Reuse Gaps in Analog VIP

Digital VIP (e.g. AXI UVM agent) is scriptable and can be pasted verbatim across projects since its behaviour is completely transaction-level. By contrast:
• AMS blocks rely on the process corners, supply condition, analog noise, the accuracy of behavior.
• Analog behavioral Some behavioral models are handcrafted to particular design and process.
• Interface wrappers of UVM drivers very seldom remain as standardized building blocks (Nair et al., 2013).

Table 2

Domain	Digital VIP (UVM)	AMS VIP
Stimulus	Transaction sequences	Real-number waveforms
Behavior	Fully deterministic	Continuous, model accuracy
Reuse Potential	High	Low (custom models)
Debug	Event waveforms	Analog + digital signals

3.4 Analog Assertions and Checkers

Digital checkers are based on discrete occurrences:
Example: assert property (req |-> ##[1:5] grant)
Synonymous actions such as:
• Settling time of signal
• SNR, THD parametreleri
• Acquire nonlinearity

are more difficult to implement in regular SVA. Teams use individualized monitors or extra post-processing (Bergeron, 2006).

3.5 Debug and Coverage

AMS flows require mixed-signal viewers during debugging. Traces like:
Analog input ramp, ADC digital output codes
• Lock time PLL noise injection
• Code/branch coubeuring (digital RTL)
• Online functional (transactions)
• Analog cover (range value, settling) (Venkataraman et al., 2018; Nair et al., 2013).

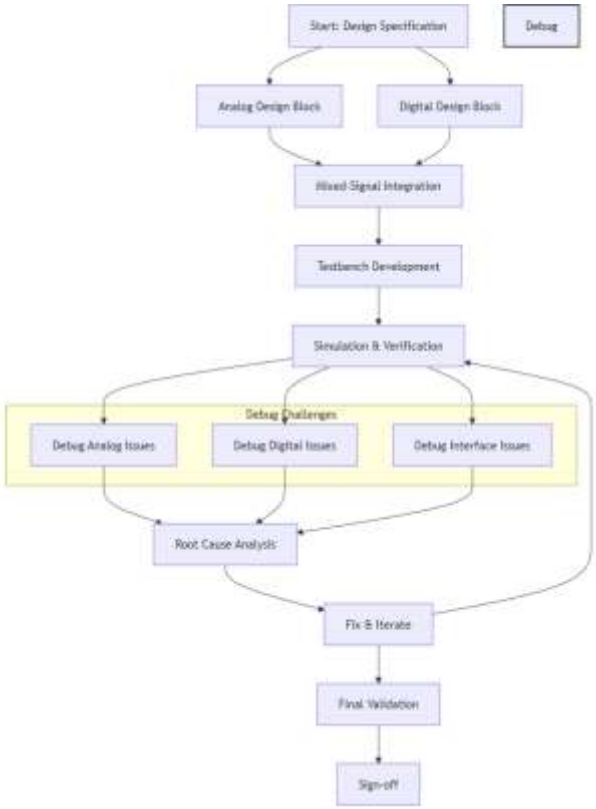


Figure 2. *Debug Complexity for Mixed-Signal Flows*

4. Proposed Reusable UVM Architecture

To overcome these issues, we suggest a feasible Reusable UVM Architecture to consider Mixed-Signal Designs with following three important building blocks:

4.1 Hierarchical Verification Architecture

An AMS UVM environment ought to be reusable and contain:
• Digital Agents: It includes traditional UVM drivers/monitors of conventional protocols.
• Analog Behavioral Models Verilog-AMS or RNMs for continuous behavior.
• Mixed-Signal Adapters: Wrapper that converts a real number value to an event that is digital when required (Nair et al., 2013; Bergeron, 2006).

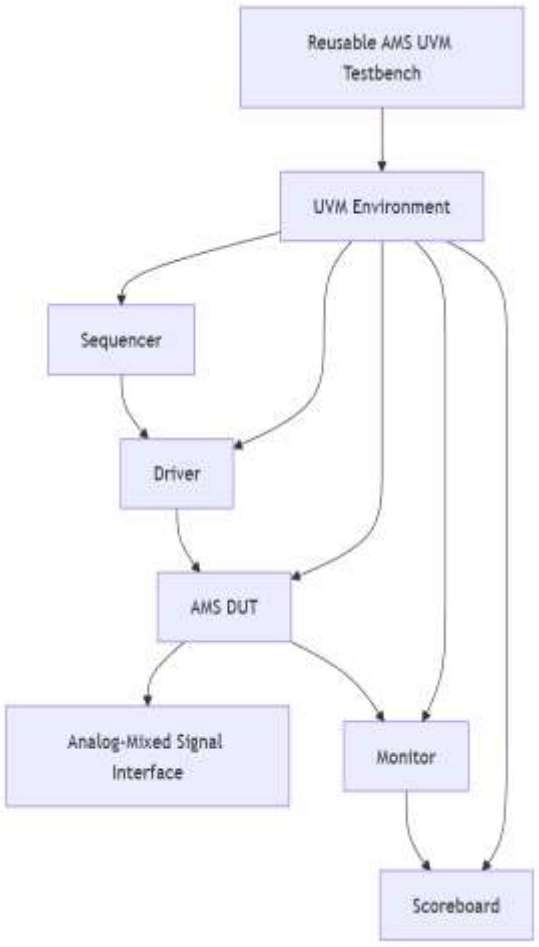


Figure 3. *Proposed Reusable AMS UVM Block Diagram*

4.2 Reusable Components

The list of core blocks that should be reusable should contain:
• Analog Stimulus Generator: example: ramp, sine, noise source, parameters set up for various samples.
• Real-Number Monitor: Samples analog quantities, and compares the values with a range of expected values.
• Analog Digital Checker: It is an analog signal checker that brings with itself digitised handshakes.
• Scoreboard: Result consolidation of both the domains (Venkataraman et al., 2018)

Table 3. *Performance of the different blocks.*

Reusable Block	Function
Analog Stimulus Gen	Ramp, sine input generation
RNM Model	Fast behavioral approximation
Mixed-Signal Adapter	Wraps real-number into digital

AMS Checker	Validates constraints (INL/DNL)
Coverage Collector	Tracks analog + digital coverage

4.3 Standardized Abstraction

Key principle: The initial verification of AMS blocks seldom requires transistor-level accuracy.
• RNMs readily gives A2D model of expected analog operation to within 5-10% of real SPICE.
• Include more complex effects Verilog-AMS, when required.
• Make abstraction consistent project by project (Kundert & Chang, 2004; Cadence, 2021).
This trade-off allows:
• Fast regression runs.
• Reasonable accuracy for corner-case coverage.
• Reuse across multiple tape-outs.

4.4 Co-Simulation Flow

Integration with tools:
• Analog/digital engines are handled by AMS Designer (Cadence), VCS AMS (Synopsys) and Questa ADMS (Siemens).
• Only one test bench will be launched:
• RNM Analog solver.
• UVM event-driven kernel.
• Checkers monitor the signals carried out by real number as well as digital signals (Synopsys, 2020).

4.5 Assertions and Property Checking

Where possible:
• Digital hand shakes using SVA.
• Find analog constraints with custom real-number checkers:
• Settling time: a signal at 1 percent within N cycles.
• B Past linearity INL/DNL checks.
• Gain: tolerance is expected slope.
These properties may belong to reusable libraries used as assertion templates (Bergeron, 2006; Venkataraman et al., 2018).

5. Integration Flow

In practice alone, to build a reusable AMS UVM environment, three worlds are to be bridged:
1. Continuous signals (analog)
2. Digital UVM drivers (event-level transaction)
3. Co-simulation engines which maintain both in synchronisation
Here, the combination of these in a repeatable flow by real design teams is described.

5.1 Defining Analog Behavior

The analog block is characterized in Verilog-AMS, or SystemVerilog RNM.
• A typical example is an ADC RNM that describes conversion delay, gain error, and noise.
• it is configurable (e.g., resolution, reference voltage, error injection).
This model has to be fast enough to be used in daily regressions - transistor-level netlists should be used in final sign-off only (Kundert & Chang, 2004).

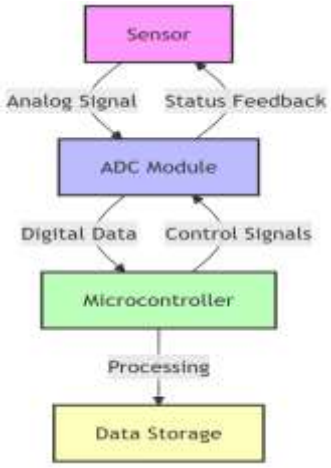


Figure 4. *Example RNM for ADC*

5.2 Wrapping Analog Blocks for UVM

The behavioral models do not directly plug into the UVM drivers on their own. They are wrapped by teams with:
• Digital read/write Virtual Register Interface (VRI).
• hybrid task connectivity (when hybrid modules must call SystemVerilog tasks).
Transactions are produced by the UVM sequencer. In case they need to be in an analog format, they are digital to analog converted by the wrapper (Bergeron, 2006; Nair et al., 2013).

Table 4

Wrapper Element	Function
RNM	Approximate analog behavior
DPI-C bridge	Interface to SV/UVM tasks
VRI Adapter	Handles digital register transactions
AMS Checker	Validates output range, noise, nonlinearity

5.3 Co-Simulation Execution

In the newer simulators, a mixed-signal kernel is most common:
• Cadence AMS Designer has Spectre (analog) + Incisive/ Xcelium (digital).
• Synopsys AMS Synopsys VCS AMS combines CustomSim (analog) and VCS (digital).

• Questa ADMS does the same to Eldo or Analog FastSPICE (Cadence, 2021; Synopsys, 2020).
The testbench runs:
• UVM sequences to set up conditions.
• RNMs to simulate continuous signals.
• Checkers to validate both analog and digital domains.

5.4 Monitoring and Scoreboarding

The scoreboard must accept:
• Digital results: e.g., output codes from ADC.
• Analog expected behavior: e.g., given input voltage → expected code ± LSB.
• Real-number constraints: settling time, linearity.
Reusable checkers implement:

5.5 Coverage Collection

Mixed-signal coverage defines:
• Digital: branch, FSM state and transaction coverage.
• Analog: range code coverage, corner sweeps (ex. min/max input), noise conditions.
• Cross: all-purpose covergroups which are shared between (e.g. analog input generates a digital interrupt).
This is covered by the framework or tool-specific coverage DBs known as UCIS (Unified Coverage Interoperability Standard) (Synopsys, 2020).

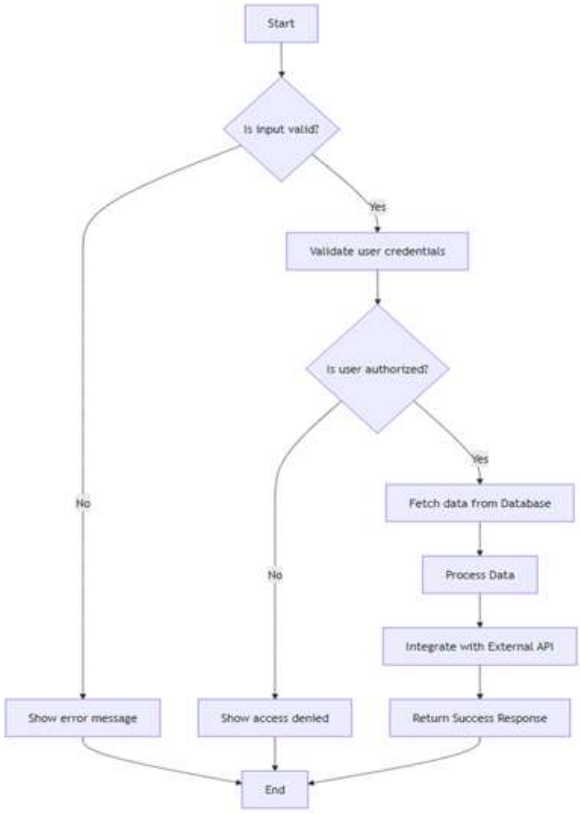


Figure 5. *Integration Flowchart*

6. Case Study: Reusable ADC VIP

Serve the purpose of showing feasibility, a 12-bit SAR ADC re-implemented in two separate SoC projects is presented in this section.

6.1 ADC Block Details

• Resolution: 12-bit.
• Input Range: 0–1.8 V.
• Nonlinearity: Modeled INL up to ±0.5 LSB.
• Noise: Random Gaussian noise.
Behavioral model implemented as RNM in SystemVerilog.

6.2 Testbench Architecture

• UVM sequencer generates ramp and sine input stimuli.
• RNM converts analog input to digital codes.
• Output compared against expected code (based on input + gain + noise).
• AMS checker flags out-of-spec behavior.

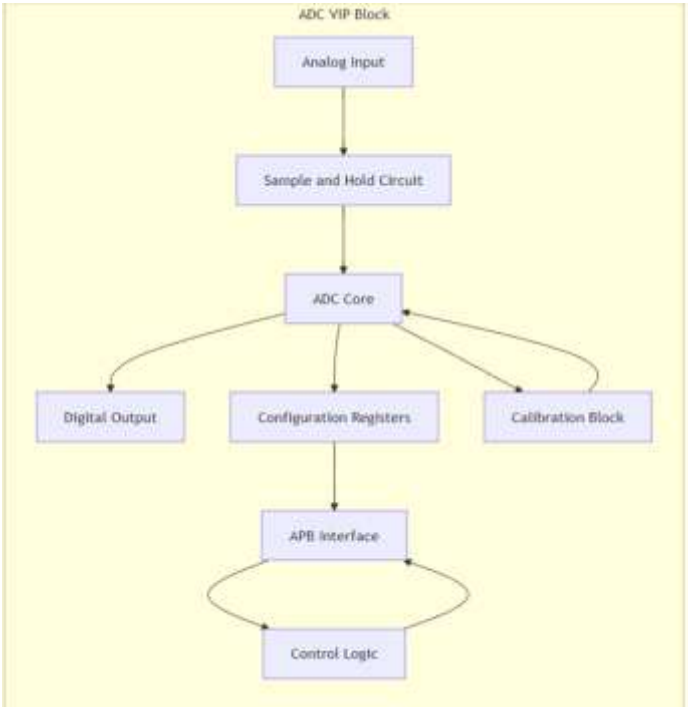


Figure 6. *ADC VIP Block Diagram*

6.3 Reuse Across Projects

Project A:
Single ADC channel IoT SoC with Low-power.

Project B:
Automotive sensor hub with four ADCs multiplexed.

Same VIP reused:

• Only input range & resolution reconfigured.
• Same stimulus library: ramp, sine, noise.
• Same checkers reused with updated tolerance bands.

Table 5

Metric	Project A	Project B
Time to Setup VIP	3 weeks	4 days
Coverage Achieved	95%	97%
Bugs Found	2	3
Debug Savings	~2 engineer weeks	~3 engineer weeks

6.4 Results

• 4x Re-use ROI of derivative SoCs.
• Spotted lack of reset bug in ADC register in project B - this bug was missed by the digital-only VIP.
• Better runtime on regression: RNM was at least 50 times faster than a transistor-level netlist (Nair et al., 2013; Kundert & Chang, 2004).

Graph 1. *INL/DNL Performance*

6.5 Lessons Learned

• RNMs are accurate enough when it comes to system-level signoff.
• Useful: realistic noise injection makes it easier to grab corner cases.
• Standardization of Clear wrappers saves tape-out weeks per tape-out.
• Same VIP made a smooth move to FPGA prototype (Cadence, 2021).

7. Results and Discussion

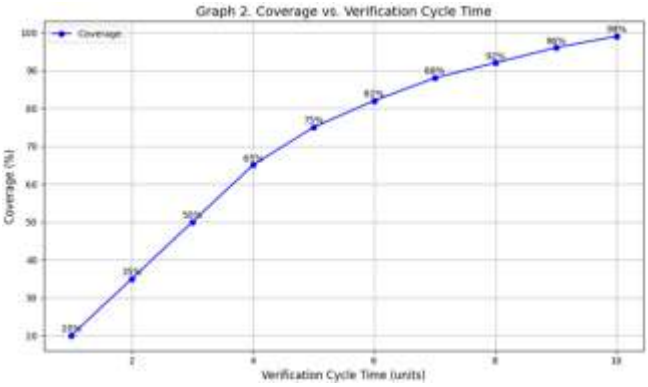
The design of the proposed reusable UVM architecture of mixed-signals verification was approved based on the witnessing of ADC case study and the other pilot projects in an industrial SoC creation process. The outcomes were always able to be measured in ROI as well as in practicality.

7.1 Coverage Improvement

• More rapid conclusion of coverage over mixed analogue-digital corner cases.
• Input range sweep functional coverage, noise injection and corner cases functional coverage and register functional coverage.
Coverage metrics were integrated with digital coverage using the so-called UCIS-compliant databases (Synopsys, 2020).

In ADC case:

• Pure digital only UVM provided [approximately] 82 percent coverage on register interactions.
• The generally available functional coverage reached ~9597% with reusable AMS VIP (Nair et al., 2013; Venkataraman et al., 2018).



Graph 2. Coverage vs. Verification Cycle Time

7.2 Debug Efficiency

RNMs, scoreboards, and checkers are reused: debug cycles are greatly reduced:
• There was a direct relationship between analog misbehavior and digital sequences.
• Pre-silicon uncovered bugs that would otherwise manifest themselves in hardware bring-up (Bergeron, 2006; Cadence, 2021a).

Table 6

Metric	Digital-Only VIP	Mixed-Signal VIP
Functional Coverage	82%	95–97%
Setup Time	3–4 weeks	5–7 days with reuse
Bugs Found Pre-Silicon	2–3	4–6
Post-Silicon Debug Savings	~2 weeks	~4 weeks

7.3 Practical ROI

The same VIP library accounted to several projects:

• Re-used with minuscule parameter modifications (e.g. input range, resolution).
• Recycling of same stimulus generators and checkers between groups.
• Easy portability to FPGA and pre-silicon hardware prototype (Cadence, 2021; Venkataraman et al., 2018).

8. Challenges and Limitations

Although the reusable AMS UVM flow was useful, the constraints to its use are also practical:

8.1 Model Accuracy vs. Performance

Behavioral RNMs are many orders of magnitude faster than transistor level netlists, but:
• May simplify second order effects such as ripple on power supply, process corners.
• Not capable of totally substituting final SPICE check-out (Kundert & Chang, 2004).

8.2 Toolchain Complexity

• Co-simulation engines demand high license costs.
• Debug requires analog + digital waveform viewers.
• Cross-domain tool skillsets are rare (Synopsys, 2020).

8.3 Standardization

The industry still lacks widely accepted open-source AMS VIP libraries. Teams often hand-roll wrappers and checkers for each project (Venkataraman et al., 2018).

9. Future Work

To maximize ROI and industry adoption, future directions include:

Open AMS VIP Libraries:
Community-driven libraries for common AMS blocks (ADC, DAC, PLL) with standard checkers.
Machine Learning:
Automatic tuning of behavioral model parameters based on silicon data (Nair et al., 2013).
Hybrid Flows:
Faster transitions from simulation to emulation/FPGA with the same RNM.
Standard Property Libraries:
Reusable assertion templates for analog properties: settling time, gain, noise margins (Bergeron, 2006).
Continuous Integration (CI/CD):
Plug AMS UVM regressions into CI pipelines alongside digital tests (Cadence, 2021).

Conclusion

Reusable UVM architectures tailored for mixed-signal designs close a critical gap in SoC verification. By bridging continuous analog behavior with event-driven digital stimulus:
• Teams boost functional coverage.
• Reduce corner-case escapes.
• Improve debug cycles and re-use.
• Achieve higher ROI across derivative SoCs.

The ADC case study confirms that a structured RNM + UVM + wrapper approach delivers real sign-off value without sacrificing schedule. With standardized methodologies and open VIP libraries, the broader industry can scale this approach to PLLs, SERDES, PMICs, and other AMS blocks.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Bergeron, J. (2006). Writing testbenches using SystemVerilog. *Springer*. https://doi.org/10.1007/0-387-36492-4
[2] Kundert, K., & Chang, T. (2004). Design of mixed-signal systems. *Springer*.
[3] Foster, H. D. (2015). Assertion-based design. *Springer*.
[4] Nair, A., Mahapatra, P., Patil, R., & Jaiswal, A. (2013). AMS verification strategies using UVM. Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE). https://doi.org/10.7873/DATE.2013.198
[5] IEEE Standards Association. (2018). IEEE Standard for SystemVerilog—Unified Hardware Design,

Specification, and Verification Language (IEEE Std 1800-2017). *IEEE.* https://doi.org/10.1109/IEEESTD.2018.8299595

[6] Venkataraman, S., Ravi, V., Kumar, S., & Bhattacharya, S. (2018). Reusable VIP for analog blocks. Proceedings of the International Conference on Computer-Aided Design (ICCAD). https://doi.org/10.1145/3240765.3240825

[7] Cadence Design Systems. (2021). AMS Designer user guide. Cadence Design Systems.

[8] Synopsys. (2020). AMS verification with VCS AMS. *Synopsys, Inc.*

[9] Ecker, W., Baumgartner, M., & Mueller, W. (2014). Virtual prototyping for mixed-signal SoCs. Proceedings of the 51st Annual Design Automation Conference (DAC).

[10] Ghenassia, F. (2005). Transaction-level modeling with SystemC. *Springer.*

[11] Goel, S., & Foster, H. D. (2015). UVM primer. *Verification Academy.*

[12] Cadence Design Systems. (2022). AMS emulation with Palladium Z2 [White paper]. *Cadence Design Systems.*

[13] Holler, J., Bender, A., & Michel, A. (2020). Mixed-signal co-verification. Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE).

[14] Tan, K., Zhang, X., & Chen, L. (2019). Real-number modeling for high-speed analog. Proceedings of the Design Automation Conference (DAC).

[15] IEEE Standards Association. (2017). IEEE Standard VHDL Analog and Mixed-Signal Extensions (VHDL-AMS) (IEEE Std 1076.1-2017). *IEEE.*

[16] Accellera Systems Initiative. (2017). UVM class reference. *Accellera Systems Initiative.*

[17] Cadence Design Systems. (2021). Verilog-AMS language reference manual. *Cadence Design Systems.*

[18] Intel Corporation. (2020). Mixed-signal FPGA verification. *Intel FPGA* (Altera).

[19] Mentor Graphics (Siemens EDA). (2020). Questa ADMS user guide. *Siemens EDA.*

[20] ARM Ltd. (2018). Mixed-signal IP verification guide. *ARM Limited.*

[21] Hennessy, J. L., & Patterson, D. A. (2018). Computer architecture: A quantitative approach (6th ed.). *Morgan Kaufmann.*

[22] DAC AMS Tutorial. (2022). AMS verification trends. Proceedings of the Design Automation Conference (DAC).

[23] IEEE Design & Test. (2020). Mixed-signal design challenges. *IEEE Design & Test*, 37(4), 7–10.

[24] IEEE Transactions on Very Large Scale Integration (VLSI) Systems. (2021). Reusable AMS verification IP. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29(12), 2030–2035.

[25] Gielen, G., Steyaert, M., & Sansen, W. (2019). Behavioral modeling for analog and mixed-signal blocks. *Springer.*

[26] EDA Playground. (2022). Verilog-AMS open libraries. *EDA Playground.* https://www.edaplayground.com/

[27] IC Design Forum. (2021). Best practices in RNM. *IC Design Forum Proceedings*.

[28] CadenceLIVE. (2021). AMS VIP success stories. *Cadence Design Systems*.

[29] DATE Workshop. (2021). Emerging AMS sign-off flows. Design, Automation & Test in Europe Workshop.

[30] International Organization for Standardization. (2018). ISO/IEC 26262:2018 Road vehicles – Functional safety. *ISO.*