

Solutions for Mixed-Signal SoC Verification

By Kishore Karnane, Greg Curtis, and Richard Goering, Cadence Design Systems

Performing full-chip verification of large mixed-signal systems on chips (SoCs) can feel like a hopeless task. As complexity grows, it's no longer adequate to join together pre-verified analog or digital "black boxes." Complex analog/digital interactions can create functional errors, which delay tapeouts and lead to costly silicon re-spins. Cadence helps customers overcome these challenges with an integrated mixed-signal verification solution spanning from basic mixed-signal simulation to comprehensive, metric-driven mixed-signal verification.

Contents

Introduction.....	1
Mixed-Signal Verification Users.....	2
Mixed-Signal Verification Challenges	3
Mixed-Signal Block and IC Level Verification	3
Real Number Modeling for SoC Verification.....	6
Conclusion.....	10

Introduction

Most systems-on-chip (SoCs) today are mixed-signal, and all SoCs will be mixed-signal at 45 nanometers and below. As process nodes shrink and the demand for integration grows, SoC designers are adding more analog circuitry and importing large blocks of mixed-signal intellectual property (IP). This escalating complexity can cause severe problems for mixed-signal SoC verification. In fact, engineers often report they can't complete chip-level verifications, or accurately gauge when enough verification has been done.

Things were simpler in the past, when mixed-signal SoCs contained IP blocks that were designed separately and then bolted together during system integration. Designers simply brought a handful of "black boxes"—blocks of analog circuitry that were presumed to be pre-verified—into a mostly digital SoC design. Now, however, analog IP blocks are not only growing more numerous and complex but also increasingly contain digital control logic. Additionally, today's mixed-signal SoCs typically contain multiple feedback loops and exhibit complex interactions between the analog and digital domains. As a result, teams cannot fully verify these highly integrated SoCs using a traditional black box approach.

According to industry estimates, over 50 percent of SoC design respins at 65 nanometers and below are due to mixed-signal errors. A respin may cost an extra 5 to 10 million dollars and a 6 to 8 week delay in a product rollout, with potentially disastrous consequences. Many respins are due to common-place, avoidable errors such as inverted or disconnected signals. To avoid these errors, mixed-signal SoC teams need to implement verification methodologies that can quickly and accurately validate interfaces between analog and digital domains.

Additionally, top-level, mixed-signal SoC verification is challenging because it encompasses both analog and digital IP blocks at different levels of abstraction. The blocks could be represented in schematics, Spice netlists, analog behavioral models, or purely digital models. This makes it essential to have a hierarchical verification approach—one that supports different levels of abstraction and different simulation engines and modeling languages.

This paper presents solutions for mixed-signal verification. After discussing common verification challenges, it looks at mixed-signal block and IC level verification using analog behavioral modeling and combined analog and digital solvers. It then describes the use of real number modeling (RNM), which replaces analog portions of an SoC with functionally equivalent real number models, for top-level SoC verification. These approaches are supported by Cadence tools and flows. Previous whitepapers in this series discussed mixed-signal design challenges and mixed-signal implementation solutions.

Mixed-Signal Verification Users

Traditionally, there have been two types of mixed-signal verification users. Digital-centric users verify ICs primarily constructed of digital logic developed with a standard cell methodology. Analog blocks that support specific functions and protocols are integrated by importing hard analog IP. These are traditionally black boxes that provide no visibility into the IP. This is sometimes called a “big D, little A” or “digital-on-top” methodology.

Analog-centric users import digital logic blocks into analog, custom digital, or RF circuits. The digital blocks may provide control, calibration, or connectivity functions. This is sometimes called a “big A, little D” or “analog-on-top” methodology.

Today, a new type of mixed-signal verification user is emerging. This user works with complex, mixed-signal SoCs, and needs to run full-chip verification that covers all possible analog/digital interactions. The SoCs may have many analog blocks, along with some mixed-signal blocks that could have been entire chips in previous process generations. As such, a black box approach that provides no visibility into signals and assumes blocks have been completely pre-verified is no longer adequate.

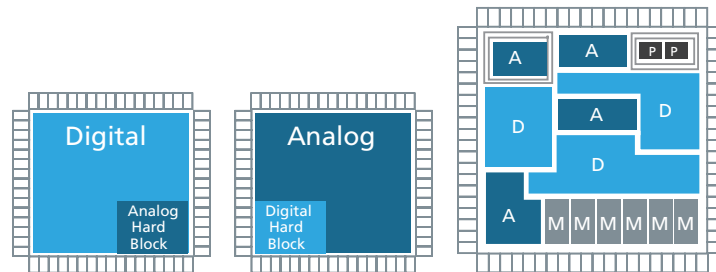


Figure 1: Mixed-signal verification users may be digital-centric (left), analog-centric (center), or focused on complex mixed-signal SoCs (right).

Ideal features for complex SoC verification include the following:

- Block importation with full visibility into signals and design details, providing an ability to debug the block if errors emerge
- The ability to model discrete real data in an all-digital simulation
- Integrated analog/digital debugging
- Support for modeling and simulation at various levels of abstraction, including Spice, analog behavioral modeling, and digital HDLs
- An understanding of the impact of low-power design techniques—such as power shutoff—on both analog and digital IP
- Single-kernel integration of analog and digital solvers
- Verification planning, testbench automation, and coverage metrics applicable to the entire mixed-signal SoC
- Support for verification reuse and verification IP
- Fast mixed-signal regression runs

Mixed-Signal Verification Challenges

In all types of IC design, the verification task is growing exponentially as complexity increases. For digital ICs, it is often said that functional verification now takes up 70 percent of the logic design phase. Add analog and mixed-signal IP, and that task gets even more complex. Even in digital verification environments, simulation is never fast enough. Yet digital RTL simulation is orders of magnitude faster than Spice-based analog simulation.

Analog and digital simulations use fundamentally different paradigms. While digital simulators solve logical expressions sequentially by triggering events, analog simulators must solve the entire analog system matrix at every time step. Each element in the analog design can have an instantaneous influence on any other element in the matrix. There is no obvious signal flow in any direction, and time is continuous rather than discrete.

The analog verification methodology is traditionally ad-hoc by nature, lacking the formalized methodology that is available on the digital side. Digital verification teams now have access to executable verification plans, constrained-random stimulus generation, testbench automation, assertions, and coverage metrics. The metric-driven digital verification approach helps engineers determine when verification is complete. On the analog side, verification is driven by directed tests run over sweeps, corners, and Monte Carlo analysis. Several analog solvers today provide low-level device checks, but there is little or no support for verification planning or coverage metrics.

As noted previously, many silicon respins stem from mixed-signal verification issues. Customer experience shows that many respins are caused by what some might call “highly embarrassing” errors, including pin connection errors, inverted polarity, incorrect bus order, or pins connected to the wrong power domains. In the absence of simple checks, such errors are often found only in lengthy analog simulation runs, if they are found at all.

Advanced low-power techniques are causing new complications for mixed-signal verification. For example, consider a digital control logic circuit that feeds into an analog block. If the power is shut off in the digital circuit, the simulator will model data corruption internal to the power domain by setting all the internal values to Xs (unknowns). If the simulator does not understand the impact on the analog block, it may be difficult to determine whether the X states derive from the shutoff or from a functional failure.

Mixed-Signal Block and IC Level Verification

Verification of a mixed-signal SoC involves many different levels of abstraction. In general, transistor-level simulation with Spice remains the gold standard for analog IP verification. While it provides very high accuracy, Spice is much too slow for chip-level simulations, unless it is used extremely selectively.

Analog behavioral modeling

To achieve reasonable simulation speeds, many mixed-signal teams employ analog behavioral modeling. This approach can be 5 to 100 times faster than Spice. The actual speedup varies widely depending on the application and the level of detail in the model. Analog behavioral models are typically written in one of the following languages:

- Verilog-AMS – a mixed-signal modeling language based on IEEE 1364 Verilog that can define both analog and digital behavior, providing both continuous-time and event-driven modeling semantics
- Verilog-A – the continuous time subset of Verilog-AMS, aimed at analog design.
- VHDL-AMS – similar in concept to Verilog-AMS, this language provides analog and mixed-signal extensions to IEEE 1076 VHDL

The creation of analog behavioral models can be challenging. Analog designers are in the best position to create these models, since they are familiar with their own circuits. But many analog designers lack the programming skills or knowledge required to construct behavioral models, and few are familiar with Verilog or VHDL. Digital designers have that familiarity, but know less about the analog circuits.

The chart below shows comparative accuracy and simulation performance between Spice, FastSpice, analog behavioral modeling (with Verilog-AMS and VHDL-AMS), real number modeling (with real and wreal data types) and pure digital simulation. These numbers are generic and can vary significantly for different applications. Note the wide range of accuracy and performance that is possible for Verilog-AMS and VHDL-AMS behavioral models. Pure digital simulation can only represent an analog signal as a single logic value, but this may be sufficient for connectivity checks in mixed-signal SoCs.

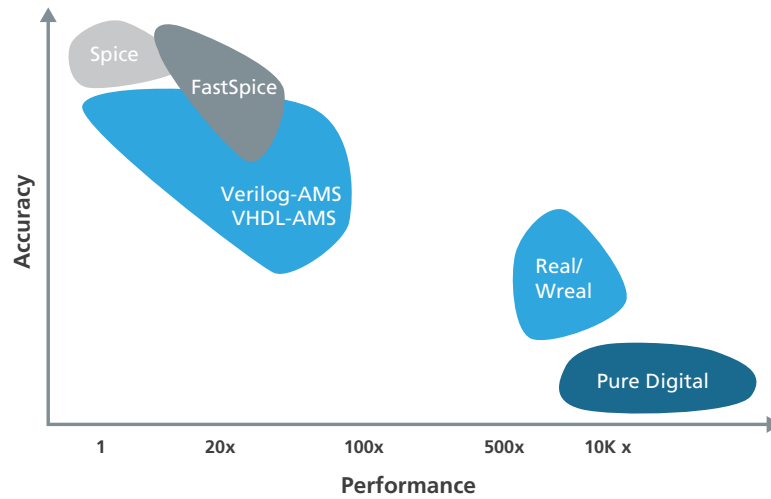


Figure 2: Model accuracy versus performance gain for mixed-signal simulation

Another important factor is the effort required to set up a simulation and create the model. While Spice simulations run slowly, they are relatively easy to set up. The time required to create a high-quality analog behavioral model, however, can range from hours to days, or even weeks. Real number modeling is restricted to a signal flow approach, analog convergence is not an issue, and there is no new language to learn. Consequently, the real/wreal modeling effort is less than that for Verilog-AMS or VHDL-AMS.

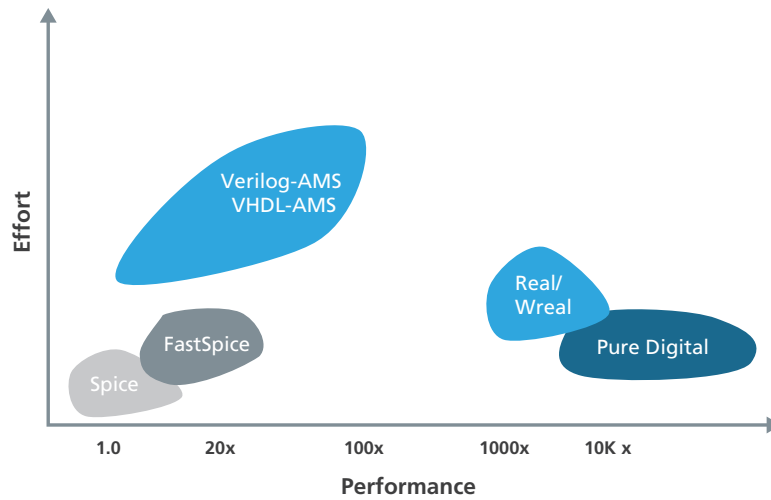


Figure 3: Modeling effort versus performance gain

The modeling goals of analog behavioral models may differ. A performance model needs to precisely capture critical circuit behavior. Functional models capture circuit behavior only to the level of detail that is needed to verify the correct design functionality.

Virtuoso AMS designer

Co-simulation between analog and digital solvers is one methodology that has been used for mixed-signal block and chip verification. Nonetheless, traditional co-simulation approaches have been plagued with limitations. Early co-simulation environments, for example, typically employed Verilog and Spice operating in separate simulation kernels linked through interprocess communications (IPC). This made it difficult to keep analog and digital simulation engines in lockstep. Users typically had to partition the circuit, deal with two netlists, and cope with two disparate debugging environments.

Cadence offers an advanced mixed-signal verification solution—the Cadence® Virtuoso® AMS Designer Simulator. Providing better performance than traditional co-simulation solutions, this product utilizes a single, executable kernel for both analog and digital simulation engines. Virtuoso AMS Designer also provides extensive language and modeling support, including behavioral models in Verilog-A, Verilog-AMS or VHDL-AMS; transistor-level analog circuit models; and support for digital languages such as Verilog, SystemVerilog, VHDL, SystemC, and e.

Virtuoso AMS Designer links the Cadence Virtuoso custom design platform with the Cadence Incisive® digital verification platform. It provides an integrated GUI, integrated embedded simulation engines, and a common verification methodology (Figure 4). Virtuoso AMS Designer supports simulation engines including Cadence Virtuoso Spectre Circuit Simulator, Cadence Virtuoso UltraSim Full-Chip Simulator, Cadence Virtuoso Accelerated Parallel Simulator, Cadence Virtuoso Spectre RF Simulation Option, and Cadence Incisive Enterprise Simulator.

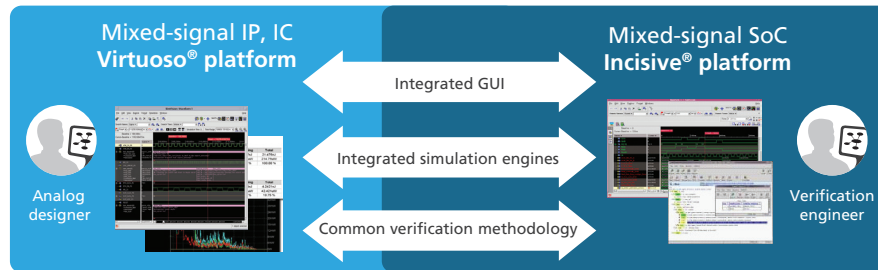


Figure 4: Virtuoso AMS Designer links Virtuoso and Incisive simulation engines.

Virtuoso AMS Designer automatically inserts bidirectional “connect” modules that translate signals between digital (logic) pins and analog (electrical) pins. This is important because digital simulators traditionally understand only 0, 1, X and Z, while analog simulators work with continuous values. Connect modules translate digital signals to and from analog voltage levels.

A “power smart connect module” goes one step further, allowing the Common Power Format (CPF), which defines digital low-power structures, to be leveraged in a mixed-signal simulation. If an analog signal’s source can be traced to a digital signal that has a CPF definition, Virtuoso AMS Designer can automatically insert a power smart connect module that can distinguish between an X resulting from a functional error and an X resulting from power shutoff, nominal conditions or power modes.

Virtuoso AMS Designer lets users interchange different levels of abstraction, allowing the design to change over time from full behavioral to full transistor level. A hierarchy editor configures the design, facilitating the viewing and design preparation for a mixed-signal simulation. Aimed primarily at applications with significant analog content, AMS Designer is tightly integrated with the Cadence Virtuoso Analog Design Environment (ADE) for mixed-signal block design.

Finally, Virtuoso AMS Designer supports real number modeling (RNM) in both Virtuoso and Incisive platforms, allowing the simulation of discrete, floating-point real numbers that can represent voltage levels. RNM enables users to describe an analog block as a signal flow model, and then simulate it in a digital solver at near-digital simulation speeds.

For analog and mixed-signal block verification, RNM can be used to speed high-frequency portions of the analog signal path—which take the longest to verify in simulation—while DC bias and low-frequency portions remain in Spice. But the greatest advantage of RNM is in top-level SoC verification, where engineers can represent all electrical signals as RNM equivalents and stay within the digital simulation environment.

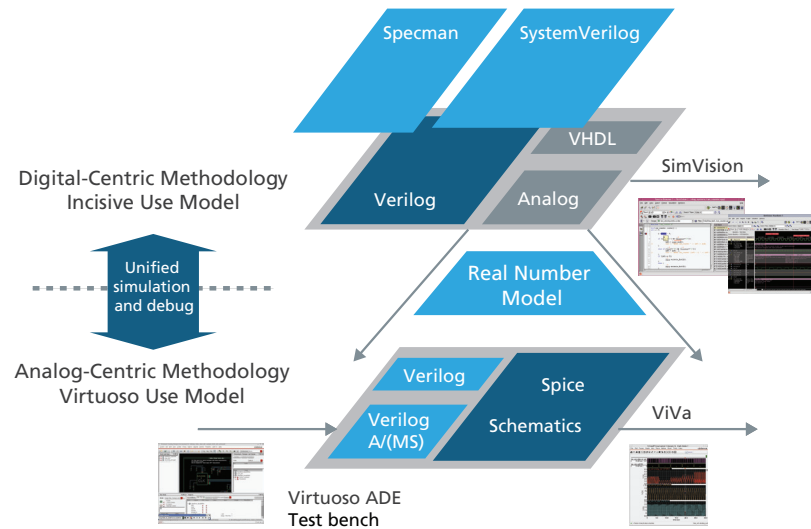


Figure 5: Virtuoso AMS Designer offers a range of simulation methodologies for both analog-centric and digital-centric use models.

Real Number Modeling for SoC Verification

The most obvious advantage of using RNM for top-level SoC verification is that it runs nearly as fast as pure digital simulation, which is orders of magnitude faster than Spice-based simulation or even analog behavioral modeling. This makes full-chip verification possible for large mixed-signal SoCs. Digital simulation speeds permit nightly, high-volume regression tests. With no analog engines, there are no concerns about convergence errors.

Metric-driven verification

Another advantage of staying within the digital simulation environment is the availability of a metric-driven verification (MDV) methodology. In the Incisive verification environment, MDV makes it possible to use specifications to create verification plans, measure progress, and more easily determine when the verification process is complete. Functional and code coverage, checks, and assertions provide the verification metrics used to determine closure. Information from verification job failures, bugs, and design revisions provides insight into the status of a project.

The MDV flow starts with automated planning. The plan specifies the verification environment requirements for a coverage-driven testbench language such as SystemVerilog or e. Verification IP, which can be used with the automated Cadence Compliance Management System, provides immediate access to the MDV methodology by delivering a protocol-specific verification plan and test suite. Progress reports help the verification team make adjustments to their resource allocations in people and tools, making it possible to reach closure more efficiently and measure closure more accurately.

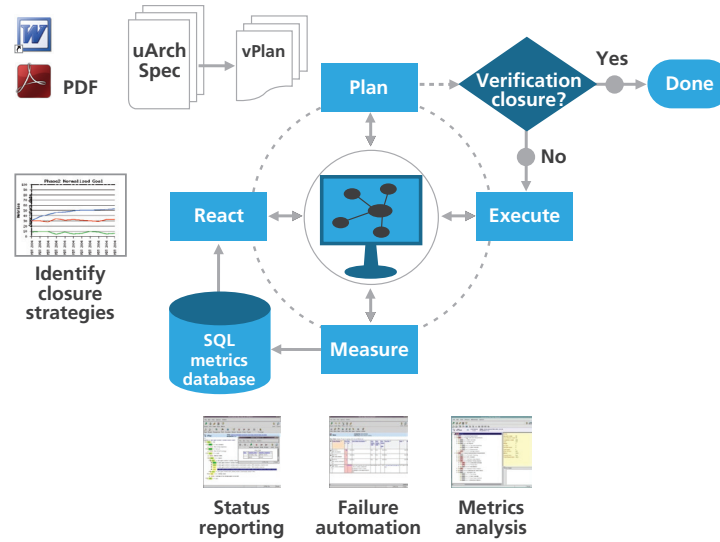


Figure 6: Metric Driven Verification Management Flow

With SystemVerilog and e functional coverage capabilities, MDV permits an advanced coverage-based verification and debug methodology to reach verification closure quickly.

This metric-driven methodology is currently employed mainly by digital engineers, but since the majority of the SoCs today are mixed-signal, more and more verification engineers are looking to adopt this approach for mixed-signal SoCs. The mixed-signal MDV flow takes advantage of RNM to enable customers to perform top-level SoC verification.

Language support for real and wreal

Five languages support RNM: Verilog, SystemVerilog, VHDL, e, and Verilog-AMS. The first four support a real data type, while Verilog-AMS supports wire-real, or wreal. The chart below shows what capabilities each language supports. (“Disciplines” differentiate domains, such as power domains, in Verilog-AMS).

<p>Verilog real</p> <ul style="list-style-type: none"> • Module internal usage of real variables • No real value ports (requires real 2bits/bits2real) • No support for X/Z state • No multiple wreal driver <p>VHDL real</p> <ul style="list-style-type: none"> • Real valued ports • Resolution function • Multiple drivers • User-defined types • Limited connection to analog <p>Specman/e real</p> <ul style="list-style-type: none"> • Mainly for testbenching • Random generation, coverage, checking • Direct access to analog values (receive/drive) 	<p>SystemVerilog real variables</p> <ul style="list-style-type: none"> • Uses real ports directly • No resolution functions • No multiple drivers • No support for X/Z state • Limited connectivity to analog models • No disciplines association <p>Verilog-AMS wreal</p> <ul style="list-style-type: none"> • Easy interaction with analog • Direct connection to electrical nets using E2R and R2E connect modules • Disciplines association • Multiple wreal driver support • Ability for scope-based wreal resolution function specification • Identification of high-impedance/unknown state (X/Z support)
--	--

Figure 7: Language support for real number modeling

As the chart shows, real data types in Verilog and SystemVerilog have some limitations. In Verilog, real variables can only be used internally in modules, not on ports. The `$real2bits()` or `$bits2real()` functions are needed to connect to the outside world. SystemVerilog supports the usage of real variables as ports, but still shares some limitations with Verilog, such as lack of support for X (unknown) or Z (high impedance) simulation states, and limited connectivity to analog modules.

Wreal is a native Verilog-AMS language feature that brings the benefits of digital signals into Verilog-AMS, including capabilities listed on the chart above. For example, wreal allows real variables on ports. The VHDL real data type provides similar advantages. Compared to VHDL real, Verilog-AMS wreal is more advanced in the area of connect modules, while VHDL real is slightly more flexible in terms of resolution function and user-defined types.

The e language supports real number models as well as coverage constructs. On the product side, Specman (included as part of the Incisive Enterprise Simulator-XL product) offers a direct interface into Virtuoso AMS Designer to access and drive analog values to and from e. Specman basically controls the simulator, generates input stimuli, performs the output checks, and summarizes the coverage measured in an integrated verification environment. Combining the Specman approach with the mixed-signal Virtuoso AMS Designer simulator provides the capability to perform checking, randomization and coverage-driven mixed-signal verification today.

The voltage-controlled oscillator (VCO) shown below was modeled using wreal. If this VCO had been modeled using a Verilog real variable, it would have been necessary to use a `$bits2real()` function to get a real number input.

```

module vco(vin, clk);
  input vin; wreal vin;
  output clk;
  reg clk;
  real freq,clk_delay;
  always @(vin) begin
    freq = center_freq + vco_gain*vin;
    clk_delay = 1.0/(2*freq);
  and
    always #(clk_delay) clk = !clk;
endmodule

```

Figure 8 – Model of VCO using Verilog-AMS wreal

RNM is not, however, a replacement for analog simulation. It is not appropriate for low-level interactions involving continuous time feedback or low-level RC coupling effects. It is not intended for systems that are highly sensitive to nonlinear input/output impedance interactions. And, real-to-electrical conversions require some careful consideration. If one is too conservative, there will be a large number of time points. If one is too liberal, there can be a loss of signal accuracy.

Incisive DMS option

The DMS Option (Digital Mixed Signal) to the Incisive Enterprise Simulator permits real number models to run natively in a pure digital environment. Users can run full-chip verification with digital solvers for functional simulation and interconnect verification. When more accuracy is needed, users can still run transistor-level simulation or analog behavioral models in the same environment. Wreal models are portable between Incisive and Virtuoso, and can be brought back into Virtuoso for diagnosis or revision if bugs are found.

In the Verilog-AMS Language Reference Manual (LRM), wreal nets can have at most one driver, can only connect to other wreal or real-valued expressions, and do not support arrays. Cadence has developed some extensions to the LRM description that remove these restrictions. With DMS Option, the Incisive simulation environment supports:

- Electrical to wreal, and wreal to electrical, connect modules
- Support for wreal arrays
- Support for wrealXstate and wrealZstate
- Support for multiple wreal drivers and resolution functions
- Ability to connect a wreal variables to VHDL real signal, Verilog real, SystemVerilog real or e real variables
- Automatic “type-casting” to wreal, when a wire is hierarchically connected to a wreal, SystemVerilog real variable, or VHDL real signal

Modeling with wreal

There is no “recipe” that shows how to create wreal models. Due to different modeling styles in the electrical and logical domains, it is impossible to provide a 1-1 mapping between all functions and modeling practices. The best way to learn wreal modeling is to study examples that are similar to what an engineering team needs to do. Cadence provides a sample library of wreal models for Incisive and Virtuoso users as well as a modeling guide to help customers develop their own wreal models.

Verification of real number models is essential. In most cases, the original transistor-level representation is used as a reference implementation. To verify the model against the reference, engineers run the same simulation on both and compare the results. Simulation setups and testbenches should be available from the block-level verification flow. Comparisons can be done manually, or a tool called Cadence DCM Model Validation can help set up runs and compare results.

Conclusion

Full-chip verification of large mixed-signal SoCs is a daunting task. As complexity grows, it is no longer sufficient to bolt together pre-verified analog or digital “black boxes” and hope for the best. Complex interactions between analog and digital domains are resulting in more and more functional errors, which in turn are causing delayed tapeouts and silicon respins that may cost millions of dollars.

Fortunately, there are solutions. A wide range of modeling and simulation approaches are available for analog and digital circuits, and most have their place. Spice-based simulators are still needed for verifying individual analog IP blocks. When it is time to move up to the subsystem or chip level, analog behavioral models can provide up to a 100X performance increase.

While traditional co-simulation solutions link separate analog and digital kernels, the Virtuoso AMS Designer provides single-kernel execution for a variety of analog and digital solvers. It also supports a number of modeling languages, including VHDL-AMS and Verilog-AMS. Virtuoso AMS Designer uses automatically inserted “connect modules” to translate between digital and analog signals.

For top-level SoC verification, engineers can convert analog models into real number models. This makes it possible to stay completely within the digital simulation environment, taking advantage of features such as verification planning, random test generation, coverage, and assertions. It also allows near-digital simulation speeds. The Cadence Incisive DMS Option brings real number modeling, including expanded support for the Verilog-AMS wreal data type, into the Incisive verification platform.

Cadence has leveraged its unique breadth of technology and its strengths in analog and mixed-signal design and verification to help customers overcome the challenges posed by today’s SoCs. The result is an integrated mixed-signal verification solution that spans from basic mixed-signal simulation to comprehensive, metric-driven mixed-signal verification.



Cadence is transforming the global electronics industry through a vision called EDA360. With an application-driven approach to design, our software, hardware, IP, and services help customers realize silicon, SoCs, and complete systems efficiently and profitably. www.cadence.com