MS-SoC Best Practices – Advanced Modeling & Verification Techniques for first-pass success

Neyaz Khan
Maxim Integrated
14460 Maxim Dr.,
Dallas, TX, 75244
Neyaz.Khan@maximintegrated.com

Greg Glennon
Maxim Integrated
14460 Maxim Dr.,
Dallas, TX, 75244
Greg.Glennon.maximintegrated.com

Dan Romaine
Cadence design Systems
2655 Seely Avenue
San Jose, CA, 94087
dromaine@cadence.com

Abstract*—Mixed-signal applications are among the fastest growing market segments in the electronics and semiconductor industry. From watching mobile digital TV to reading on your tablet to auto-piloted cars, consumers expect electronics to do more—in more places than ever before.*

Most systems have to interface their millions of gates, DSPs, memories, and processors to the real world through a display, an antenna, a sensor, a cable or an RF interface. Driven by growth opportunities in mobile communication, networking, power management, automotive, medical, imaging, and security applications, there is an increased emphasis on the integration of high performance digital with high-performance analog, RF and mixed-signal designs [3].

Due to this trend, a predominantly high percentage of SoCs today are mixed signal in nature. As process nodes shrink and the demand for integration grows in the era of ubiquitous communications and always-connected devices, the trend is projected to tilt more towards increased Mixed Signal contents on SoCs of the future. This paper examines some Advanced Verification techniques used for the functional verification of modern Mixed-Signal SoCs.

Keywords*—UVM, UVM-MS, MS-SoC, Mixed Signal, CDV, MDV, vPlan, wreal, RNM, Functional Coverage*

I. INTRODUCTION

A. Mixed Signal SoC Verification Challenges

As more analog and Mixed signal IP gets added onto the SoC, the task of Functional Verification, already a daunting task in the digital-only domain just keeps getting harder and harder. Over the years there has been a huge investment in digital verification – spurred the development of tools and methodologies for systematic and cost effective functional verification in the digital domain. In the last several years a similar need is building-up in the analog design space [4].

According to industry estimates, more than 60 percent of SoC design re-spins at 45 nanometers and below are due to mixed-signal errors with expensive and often disastrous consequences. Many re-spins are due to commonplace, avoidable errors such as inverted or disconnected signals. To avoid these errors, mixed-signal SoC teams need to implement verification methodologies that can quickly scale and accurately validate the interfaces between analog and digital domains.

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. In digital design, the metric-driven verification approach—standardized for reusability as the Universal Verification Methodology (UVM)—helps engineers build confidence in the verification by increasing coverage to a desired level. 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.

In addition, 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.

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.

This paper will document the step-by-step process for taking a design from schematic through tapeout using Advanced MS-SoC verification methods. We will discuss the particular roles the various engineers will play at each step throughout this process. We will also discuss the current limitations at the various stages of this process.

B. What is the MS-SoC Flow?

The MS-SOC flow involves applying advanced verification methods typically used within a digital environment and applying them to a Mixed Signal System. These methods range from utilizing executable verification plans to applying assertions and coverage to determine the level of verification completeness.

There are several common questions that come up from design teams with regards to the MS-SoC flow. Will analog designers now have fewer simulations to run? Will block level characterization be done at the full chip level? Will there be less overall work to do? These questions are often accompanied by the common digital concerns such as "What percentage of verification must be done at the block level if we are now placing so much emphasis at the chip level verification environment?"

These methods do not offer a replacement or short cuts to the current methods of simulating and verifying the electrical characteristics of an analog block. In fact this method often requires additional work by the analog designers. They must now support a modeling effort for their blocks. The tradeoff for this additional effort is lower risk of costly respins later on.

The question of spice accurate block level characterization at the full chip level is still unreasonable due to costly simulation speeds. This approach does not change the spice simulation speeds.

The MS-SoC flow focuses on verifying the interactions between the analog and digital domains as well as the functionality of the analog blocks themselves using the proven verification techniques from the digital design world [5].

The MS-SoC flow is described in more details in the subsequent sections.

C. What is UVM-MS?

Digital verification engineering emerged in the last 20 years as an indispensable part of chip design. As complexities grow and productivity pressures rise, the expansion of verification engineering into the analog space in the short term is inevitable. The extended methodology is named **UVM-MS**. Methodology extensions include verification planning for analog blocks, analog signal generation, checking and assertion techniques for analog properties and analyzing analog functional coverage. The methodology features abstract, high level modeling of analog circuits using *real number modeling* (RNM). Automation and management aspects include batch execution and regression environments, as well as progress tracking with respect to the verification plan [3].

Some of the main highlights of the MS-SoC flow [4]:

- Creation of an executable verification plan (vPlan) for analog DUT
- Real Number Model (wreal)
- UVM-MS based verification components that contains:
  - Digital MS based Analog signal generation using a wire UVC
  - Analog monitors that measure the envelope of a signal – with built in coverage
  - Driving and monitoring configurations controlled by analog sequences with programmable resolution
  - Functional Coverage collection on analog parameters in design
  - Mixed Signal assertion based checks that span between digital & analog
- Closing the loop – backannotating analog coverage & checks onto the vPlan for verification closure.

D. Benefits of Mixed Signal SoC Flow

The benefits of the MS-SoC flow are many. By removing the analog elements and replacing them with real number models, the simulation is now able to remain entirely within the digital domain turning multi-day analog simulations into minute long runs. This balance of speed versus accuracy is key to determining just how much of a performance increase one will see. It is up to the designer of the analog block, the model creator, and the system integrator to determine the proper level of abstraction for the real number model. Too much detail will require evaluation to happen too often and as such slow down the simulation, however, too little detail will fail to provide enough information to truly verify the interactions properly.

The ability to apply random stimulus concurrently in a digital system has already proven invaluable to finding bugs within a system and these methods enable the same advancements to be applied within a mixed signal environment. Random stimulus if far more likely to catch hidden bugs than directed testing and this is true for analog just as much as it is for digital.

The ability to provide coverage and checkers through the use of assertions and covergroups provides measurable results for determining verification complete. This enables tracking of the progress of the project as well as helps in determining what areas have yet to be tested. All of this leads to more efficiency in the verification process, which leads to faster cycle times for projects.

II. The Advanced MS-SoC Flow

What are the steps to the Advanced MS-SoC Flow?

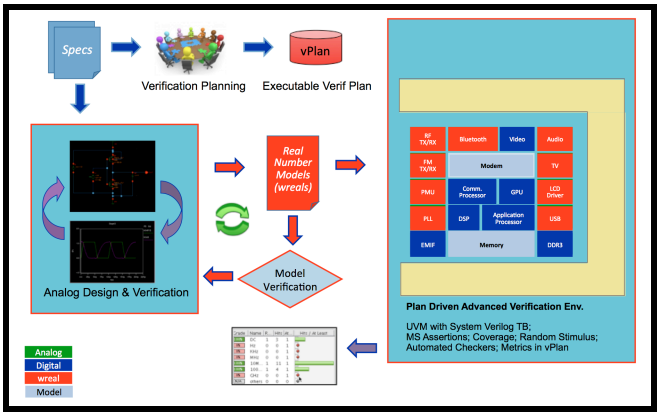

Figure 1: MS-SoC Flow using Real Number Models

As illustrated in Figure 1, state-of-the-art functional verification methodologies like UVM are applied to the verification of Mixed Signal (MS) designs using the guiding principles of Metrics Driven Verification (MDV). MDV is a broadly used concept for verifying large digital designs. Modern designs have huge state spaces hence it is impossible to simulate all their possible conditions, or even a small fraction of those. MDV is guided by the functional specification, rather than the design implementation. The functional specification is parsed down to a hierarchy of features in a *verification plan*, where each feature can be shown to meet the specification by some measurement. These measurements are called *functional coverage*. The resulting functional coverage space is many orders of magnitude smaller than the design state space – making it a practical metric. A reusable verification environment (VE) is created based on UVM to exercise the design, check its functionality and measure coverage. Layers of automation are added to run large volumes of simulations with random perturbations. The collected coverage is aggregated and compared with the verification plan. Areas lacking in coverage are targeted to get an over-all balanced coverage [4].

The VE allows trading of accuracy of analog with speed needed for large regressions used to comprehensively verify the MS design. The accuracy of spice models for analog is traded off with Real Number Models (RNM), typically written using wreals, which allows simulating the entire MS netlist on a purely digital simulator. These high-performance models while trading off the accuracy of spice & Verilog-AMS with simulation speed and throughput still provide the level of accuracy and detail required for effective verification of a MS netlist at the SoC level [4].

As with the digital Metric Driven Verification flow, the MS-SoC flow begins with the planning process. This includes planning on both the design and verification side. The design side must determine how to partition up the design. This partitioning is important in many different aspects including both place and route, power, and verification. Where to divide the analog from the digital has ramifications on many levels. Ideally, all of the digital logic will be removed from the analog blocks to create a very defined line. This makes it easier from a simulation and modeling aspect as well as a place and route and power verification aspect.

Once the analog has been divided out, the granularity of the modeling must now be determined. The more abstract the model, the faster the simulation, but at the cost of losing information regarding the system interactions. However, modeling all the way down to the transistor level gains little if anything to speeding up the simulation. A common ground must be determined for what level of modeling accuracy is acceptable for the specific analog feature being modeled.

In parallel, the planning process must include the verification plan. This plan must now extend to the system level interaction between the analog and digital domains. Understanding the various signals that will be interacting and what checks and coverage needs to be applied to them is key to this flow. This has the secondary effect of helping to resolve any ambiguity issues early in the design process. A simple difference of signal intent between the analog and digital designers at the start can result in large amounts of redesign and time lost during the system integration phase.

Once the planning and partitioning have completed, the design at the block level can begin. In parallel with this effort, the verification engineer can begin to build the top down environment by utilizing existing models to create a fast mock up of the design allowing them to begin to validate the solutions reached in the planning and partitioning phase.

The analog designer will proceed as always with the design and verification effort at the block level within their chosen schematic simulation environment. Once the design has been verified and characterized at the block level, it is ready for integration and real number model generation. The real number model may be generated in any number of ways and verified using already available tools which compare spice level simulation results from both the model and the actual design to help determine the model's accuracy.

While the analog engineer is going through the design / verification / modeling stages, the verification engineer has completed the initial top level system check and has begun creating the digital testbench for the system.

The testbench creation will also include creating the coverage written into the verification plan. This will also include any assertions that are required to generate the checkers specified in the plan.

As the design converges to becoming a complete system, the verification engineer can focus on the digital and mixed signal regressions. By using the advanced verification flows, the analog can be randomly exercised within the digital environment providing coverage results in the same manor as the digital items.

The simulation results are then fed back into a coverage tool along with the verification plan and then the cycle of writing and running tests while monitoring the coverage continues until the verification plan shows that all cases have been verified.

A. Real Number Model Creation

How does one generate a real number model?

The first step is ensuring that the analog design has been completed and verified at the block level. At this point, the analog designer or a modeling expert creates a digital model of the analog intent using real number equations. The resulting model can be verified against the actual analog design using industry tools such as AMS-DMV. The purpose of tools like this are to verify the model integrity is high enough to fulfill the requirements of the verification plan.

Once the model has been created, the system level integrator will take it and integrate it into the design. This may or may not be the role of the verification engineer.

Real number models have opened many doors to advanced verification techniques, but are still limited in their current scope as it is still a single value. If a designer needs to model both voltage and current for a particular signal, the current implementation falls short. There are also issues with multiple drivers on a single net. In the digital realm, this resolves to an X, but it is completely valid in the analog realm. There are proposed solutions to these concerns being looked at by committees in Accellera & IEEE. While solutions exist in some existing tools like IUS and languages like VHDL to define resolution functions for multiple drivers on a net, newer solutions are being proposed which would enable multiple analog parameters like current & voltage to be passed on a single wire. This would be a welcome feature designed to take care of current deficiencies in the use of wreals for analog/MS modeling that exist today.

B. The Verification Plan

Metric-driven verification relies on a verification plan to be used as a basis. The plan lists all the features that need to be verified, what to check for and how to measure coverage. A typical plan describes test scenarios that would exercise each feature and important feature combinations [3].

Verifying analog features often require measuring continuous values, such as voltage or current at a certain node, continuous (real) values can be sampled, but in order for them to make sense as coverage they need to be quantized into bins. For example, a supply voltage may be classified as nominal, low, high, or off—creating a four element coverage vector. More complex continuous properties, such as gain and signal-to-noise ratio can be computed based on several direct measurements, for example the signal amplitude at various locations in the data path. Such computed quantities need to be similarly quantized when captured as coverage [4].

Deciding what quantities to measure, either directly or indirectly, and how to quantize them needs to be part of the verification plan [4]. As device specifications grow in complexity, it is beneficial to partition the plan into smaller plans targeting functional blocks within the design. This is most useful when having to deal with inevitable changes to the design as the life cycle of the part progresses towards tapeout and a clearer understanding is available for any limitations missed when first conceptualizing the design. As these updates are made to the spec and need to be rolled into the verification plan, if the plan has been divided up into functional blocks, the changes will hopefully be contained to a smaller portion of the plan keeping the overall plan still in tact even after a major change.

The planning process must involve the analog, digital, and verification engineers to be effective. This helps to properly flush out ambiguities in the specification, as all parties involved are present to express any concerns when items come up. We have recently expanded this group to also include the applications and test engineers to ensure that the environment will verify both the real world applications of the device as well as all critical features required to enable fully testing the silicon in production.

This plan is not a place to include electrical characteristics that require spice level characterization to validate.

It should also be understood that the coverage and checkers placed into the plan will be coming from the digital regression environment and not from the schematic level analog simulations.

Once the planning has completed, the actual environment can begin being constructed. The environment is developed and owned by the verification engineer. The analog and digital designers are merely users of the environment. The standard methodology utilized by the developer should be UVM with an emphasis on UVM-MS.

C. *The Verification Environment*

The verification environment must begin with the discussion of the top down vs. bottom up approach. We propose that a combination of both approaches arrives at the most efficient method. By beginning with a top down approach, a quick top level system level environment is created using existing IP from previous projects. This system model is simply to justify the solutions and assumptions regarding the system interaction between various blocks determined during the planning and partitioning phase. Power partitioning requires this type of approach to be effective. By utilizing existing VIP and IP blocks, the general system should be able to be assembled and tested to a limited extent to determine that no major mistakes were made during the planning phase.

Once this level has been verified, the bottom up design flow can proceed with less risk knowing that the system integration is already understood. This provides much less risk for required redesign and loss of valuable schedule time later on in the project.

The next item to address is the environment hierarchy and whether it is digital on top or analog on top. A full chip analog on top environment is typically associated with a schematic driven simulation environment controlled by the analog design engineers. A full chip digital on top environment is typically associated with command line simulations run in a regression style environment by the digital designer, system integrator, or verification engineer. The advanced methods discussed within this paper rely upon the ability to utilize a specialized tool to handle regression management and coverage reporting. This tool requires the environment to be digital on top. By utilizing digital on top, the complexity of debugging is also decreased by the myriad of available tools out in the market today. Digital on top also enables the use of low power CPF / UPF methods to assist in further checking the DUT for power correctness.

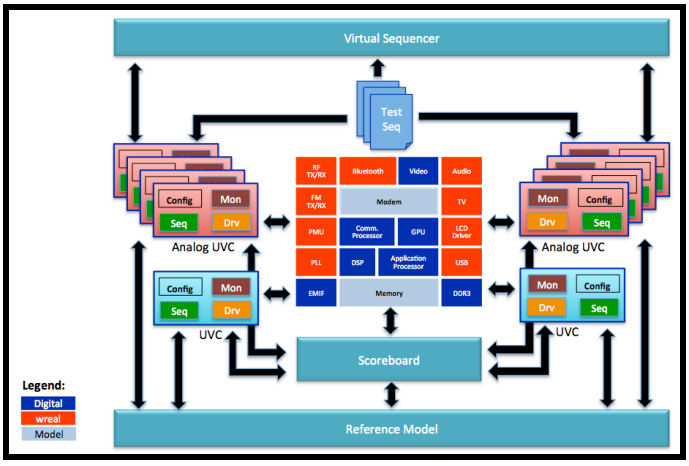

Figure 2: Verification Environment based on UVM-MS

Analog Verification IP must be created in the UVM-MS flow using real number sources to create UVM agents. These agents are created by the verification engineer, within a UVM-MS based Verification Environment as shown in Figure 2.

These real number UVCs enable interactive sources to drive the analog portions of the simulation as opposed to spice level analog sources that are decoupled from the digital simulation. If these spice sources required manipulation, it was typically performed by hand for every test case in a time-consuming trial and error process.

When swapping out real number models for the actual spice level netlist of a block, it is typically recommended to match the signal sources to the model abstraction. This means if you have a real number model, try to drive it with a real number source. If you instead have a spice or Spectre netlist, drive it with an analog voltage source.

It is possible to mix a real number source with an analog netlist in certain situations in order to maintain the UVM random source event interaction throughout the simulation. The user should understand, though, that some type of element will be inserted to interface between the real number source and the analog block. These elements may not provide the signal source the user was intending, as they are often non-ideal sources, which have other ramifications.

Once the environment and sources have been developed, the coverage and assertions need to be addressed. Currently System Verilog capabilities do not support binning of real number ranges, which limits the coverage results for analog with regards to covergroups. An indirect way to still obtain coverage is to utilize PSL assertion coverage on the analog signals instead. Other Hardware Verification Languages such as e/Specman do already support binning of real numbers into covergroups.

While newer methodologies are emerging, there are currently no standard flows to incorporate assertions within analog schematic simulations. This leaves the analog assertions to be handled within the digital simulation environment. PSL assertions enable checkers to be easily applied to real numbers enabling analog model checks to be applied within the digital environment.

Once the digital coverage has been captured, the coverage is then mapped back to an executable verification plan. Based upon new tool and language capabilities, this Metric Driven Verification feature can now be extended to include both real numbers and VerilogAMS analog values using the assertion based coverage and checkers previously described. This allows the analog portion of the design to now be pulled into the verification plan and included in the tracking prior to verification complete.

With the environment created, the time for random stimulus has arrived. Random testing is a major focus of UVM and

UVM-MS. SystemVerilog enables randomization of real number values, which can then be applied to the wreal models within the environment enabling the full power of constrained random stimulus to cross over to the analog world.

D. Automated Checks using MS Assertions

While there are various sources of errors in a MS-SoC, there are two leading causes for re-spins (1) wrong hook-up and connectivity issues at the top-level (2) analog sequence and timing. Both of these can be easily handled by writing many assertions that track the behavior of the design, continuously monitoring forbidden and expected behavior. The AMS extension to PSL supports checking electrical properties of signals, like the voltage level of a node. Hence, MS assertions are primarily coded in PSL. For details refer to [4], [3]

E. Low Power Verification

As the simulations are now running, low power verification comes into play. Low power verification has made a lot of progress on the digital side in the past few years. This has been driven by both CPF and UPF. Low power with respect to mixed signal, however, is still a black art.

Many issues still remain unresolved. Level shifters and isolation cells within the analog are not visible to the digital realm. Many other low power artifacts are within a black box with respect to low power verification. These issues cause a false sense of security. As each domain is verified independently, everything works well. However once they are connected together, is can lead to many low power issues.

There are new and developing ways to handle these issues using macromodels for analog in the digital CPF.

The subject of low-power verification in the context of analog/MS is important but vast, and beyond the scope of this current paper. It will be covered in detail in future papers

III. SUMMARY AND CONCLUSIONS

As the trend for integration in the semiconductor industry continues to grow and expand to include more and more analog circuitry, new methods must be utilized to prevent costly system level errors. Continuing to rely upon analog checks at the block level is too risky and simulating using a spice simulator at the top level is too costly due to the time involved. New methods involving the use of real number models help to bring the analog world under the blanket of the already proven digital tools and techniques creating the MS-SoC flow, based on utilizing Advanced Verification Techniques like UVM-MS. Although this flow will extend a product's first pass schedule, it will greatly reduce the risk of requiring additional passes prior to moving to production.

We conclude that the MS-SoC flow is likely to become mainstream in mix-signal design verification. For those interested in more details, the experiences of the authors have been fully captured in a new book [3] and other publications have highlighted various aspects of this very successful and proven methodology [4], [5].

IV. REFERENCES

[1] Kundert, Ken and Chang, Henry, 2006. Verification of Complex Analog Circuits. *Proceeding of IEEE 2006 Custom Integrated Circuit Conf. (CICC)*

[2] Palnitkar, Samir, 2003, Design Verification with e, *Prentice Hall, ISBN: 978-0131413092*

[3] Bishnupriya B., John D., Gary H., Nick H., Yaron K., Neyaz K., Zeev K., Efrat S., 2012, Advanced Verification Topics, *Cadence Design Systems, ISBN 978-1-105-11375-8*

[4] Neyaz Khan, Yaron Kashai, 2012, DVCon From Spec to Verification Closure: a case study of applying UVM-MS for first pass success to a complex Mixed-Signal SoC design

[5] Neyaz Khan, Yaron Kashai, Hao Fang, 2011, DVCon Metrics Driven Verification of Mixed Signal Designs

[6] www.ovmworld.org

[7] www.SystemVerilog.org

[8] www.uvmworld.org