ISSN: 2634 - 8853

Journal of Engineering and Applied Sciences Technology

SCIENTIFIC Research and Community

Review Article

Open Access

Functional Verification Methodologies for Mixed Signal Designs

Niranjana Gurushankar

Hardware Verification Engineer at Cisco Systems

ABSTRACT

This paper explores the intricate methodologies for functional verification of mixed-signal designs, crucial in ensuring the correct operation of increasingly complex integrated circuits. These designs, combining analog and digital components, present unique verification challenges due to the interplay between continuous-time and discrete-time signals. Traditional digital verification techniques fall short in addressing these complexities, demanding specialized approaches. This paper provides a comprehensive overview of state-of-the-art methodologies for verifying mixed-signal designs. It explores challenges and solutions in detail, analyzing critical aspects like advanced simulation techniques, formal verification methods, and complexity reduction strategies. The paper also highlights the importance of metric-driven verification and examines emerging trends like emulation, FPGA prototyping, and AI/ML integration. This knowledge equips engineers to ensure robust and reliable mixed-signal designs in today's complex technological landscape.

***Corresponding author**

Niranjana Gurushankar, Hardware Verification Engineer at Cisco Systems, USA.

Received: April 04, 2022; **Accepted:** April 11, 2022; **Published:** April 18, 2022

Keywords: Mixed-Signal Verification, Co-Simulation, Hybrid Simulation, Real Number Modeling, Emulation, FPGA Prototyping, AI/ML

Introduction

Modern integrated circuits (ICs) increasingly incorporate both analog and digital components, leading to complex mixed-signal designs. These designs are prevalent in various applications, from wireless communication systems and sensor interfaces to biomedical devices and automotive electronics [1]. Verifying the correct functionality of these designs is crucial to ensure their reliable operation. However, mixed-signal verification presents unique challenges due to the interaction between continuous-time analog signals and discrete-time digital signals. Traditional digital verification techniques fall short in addressing these complexities, necessitating specialized methodologies [2]. This paper gives you a complete overview of the latest and greatest methods used to verify mixed-signal designs. We'll dive deep into the challenges and explore a whole range of solutions, giving you the knowledge to make sure your mixed-signal ICs are robust and reliable. This paper is your guide to the cutting edge of mixed-signal verification. By understanding these techniques and staying ahead of the curve, you can ensure your mixed-signal ICs are ready for the demands of today's technology.

Challenges in Mixed-Signal Verification

Verifying mixed-signal designs is like navigating a complex maze with unique challenges lurking around every corner. Here's a closer look at the key hurdles that make this field so intricate,

Bridging the Gap between Analog and Digital

This is the core challenge. Analog signals are continuous, while digital signals are discrete. These two worlds don't naturally mesh, and this creates several issues. Both of these worlds have different simulation needs, analog circuits require simulators that can handle continuous-time behavior and complex equations, while digital circuits need event-driven simulators that focus on discrete changes. Combining these approaches could be a little hard [3]. One of the other main tasks is synchronizing both, there could be designs where analog output goes as digital block input making the analog and digital simulations "talk" to each other accurately and efficiently is crucial. Mismatches in timing can lead to inaccurate results and mask subtle bugs [4]. When analog and digital components are connected in feedback loops, it can create instability and make it difficult to ensure that the simulation converges to a stable solution [5].

Taming Analog Complexity

Analog circuits are inherently more complex than their digital counterparts. They are nonlinear, and because of which their behavior doesn't follow simple, predictable patterns. Small changes in input can lead to large and unexpected changes in output. Analog signals are super sensitive to noise, real-world analog circuits are affected by noise from various sources, which can distort signals and impact performance. Capturing this behavior accurately in simulations is essential [6]. Creating accurate models for analog components, especially for complex behaviors and extreme operating conditions, is a constant challenge [7].

Managing Different Speeds

Mixed-signal designs often involve components operating at vastly different speeds. This can cause problems during simulation. Simulating both high-speed digital clocks and slow-changing analog signals simultaneously can be computationally expensive and time-consuming [8]. If the simulation time steps are not chosen carefully, it can lead to inaccuracies in capturing the behavior of both fast and slow components [9]. The wide range of time scales can sometimes lead to numerical instability in the simulation, making it difficult to obtain reliable results [10].

Limited Visibility

Observing and analyzing signals in the analog domain can be more challenging than in the digital domain. Analog signals can have complex shapes, making it harder to extract meaningful information compared to the clean "ones and zeros" of digital signals. Noise can obscure the underlying signal, making it difficult to identify subtle errors or unexpected behavior. Accurately measuring analog parameters in simulations can be tricky, especially in the presence of noise and parasitic effects [11].

Mixed-Signal Verification Methodologies

Advanced Mixed-Signal Simulation

Co-Simulation approach involves using separate simulators for the analog and digital portions of the design and exchanging information between them at specific time points. This allows for the use of specialized solvers for each domain but can introduce synchronization and accuracy challenges. Let's consider an example: Verifying a phase-locked loop (PLL) requires co-simulation between a SPICE-like simulator for the analog circuitry and a digital simulator for the control logic. Hybrid Simulation on the other hand, combines analog and digital simulation within a single kernel, offering tighter integration and potentially better performance. However, it requires sophisticated solvers that can handle both continuous-time and discrete-time behaviors. Let's consider an example: Simulating a sigma-delta ADC, where the analog modulator and the digital filter are tightly coupled, can benefit from hybrid simulation [12].

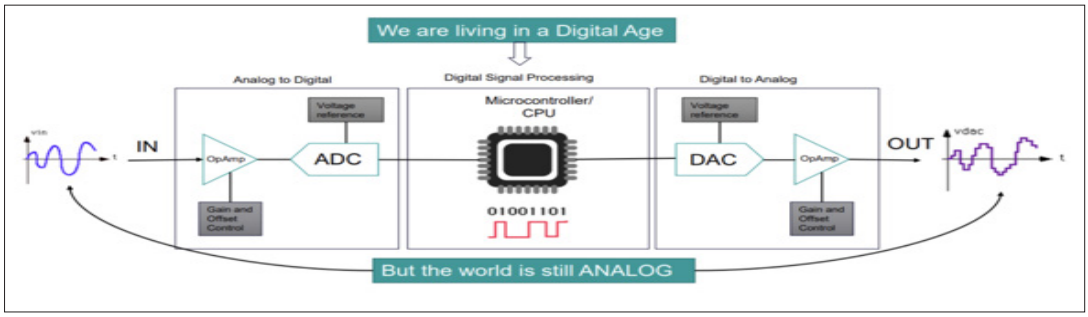

Figure 1: Mixed-Signal Simulation in the 21st Century [13].

Choosing between co-simulation and hybrid simulation depends on the specific needs of the design. Co-simulation might be suitable when the analog and digital parts are relatively independent, while hybrid simulation shines when they are tightly intertwined, like in a high-precision analog-to-digital converter. Ultimately, both approaches aim to overcome the fundamental challenge of simulating the interaction between the continuous world of analog and the discrete world of digital. By carefully choosing the right approach and tools, engineers can ensure that their mixed-signal designs function flawlessly in the real world.

Formal Verification for Mixed-Signal Designs

Imagine you have a design, and you want to be absolutely sure it behaves exactly as intended. Property checking helps this with mathematical certainty. This involves formally specifying desired properties of the design and using automated tools to prove or disprove these properties. While traditionally applied to digital designs, recent advances have extended property checking to mixed-signal systems [14]. You start by clearly stating the desired behaviors of your design. These are your "properties," expressed in a formal language that the verification tool can understand. Think of them as rules your design must always follow. You then use a specialized tool that analyzes your design and checks if it satisfies all the defined properties. This tool uses mathematical algorithms to explore all possible scenarios and ensure your design behaves correctly under every circumstance. The beauty of property checking is its thoroughness. Unlike simulations, which test only a limited set of scenarios, property checking aims to prove or disprove your properties for all possible inputs and operating conditions. This gives you a much higher level of confidence in your design's correctness.

Equivalence Checking, on the other hand verifies that two different representations of the design, such as a high-level model and a detailed implementation, are functionally equivalent. This can be challenging for mixed-signal designs due to the different levels of abstraction [15]. There are two representations of the design, a high-level model and a low-level implementation. An equivalence checking tool meticulously compares these two representations, looking for any discrepancies in their behavior. It uses mathematical techniques to prove that they produce the same outputs for all possible inputs. This is crucial for ensuring that optimizations, refinements, or translations haven't introduced any unintended changes in functionality. It's like having a guarantee that your blueprint and your building are perfectly aligned.

Table 1: Formal Verification of Analog and Mixed Signal Designs, Survey and Comparison [16].

EQUIVALENCE CHECKING				
	[1]	[2]	[3]	[4]
Type of Systems	Linear	Non linear	Linear	Non linear AMS
Models	Transfer function	ODE - DAE	Transfer function	ODE - DAE
Analysis Regions	Transient response	Near operating point transient analysis	Near operating point	Functional analysis
Analysis Domain	Frequency	Time	Frequency	Time
Techniques and Analysis	OBDDs comparisons	Qualitative analysis	Interval analysis	Rewriting, SAT simulation
Tools	N/A	MAPLE	MAPLE	M-CHECK
Case Studies	Low Pass filter	CMOS inverter, Opamp	Band pass filter, opamp	D/A converter

Behavioral Modeling and Abstraction

When it comes to verifying mixed-signal designs, we often need ways to simplify the complexity while still capturing the essence of the analog behavior. That's where behavioral modeling and abstraction come in. Below are the two key techniques used for this purpose:

Real Number Modeling (RNM)

This approach uses real numbers to represent analog signals and variables, allowing for more accurate modeling of analog behavior within a digital simulation environment [17]. Instead of dealing with discrete values like in digital systems, RNM uses real numbers to represent the continuous values of analog signals and variables. This allows for more accurate modeling of analog behavior within a digital simulation environment. For example, instead of representing a signal as a series of discrete steps, RNM can represent it as a smooth curve, capturing its true continuous nature.

Analog Behavioral Languages

You need a language that can express both the continuous flow and the actions of these components. Languages like Verilog-AMS and VHDL-AMS extend traditional hardware description languages with special constructs for describing analog behavior. They allow you to create abstract models that capture the essence of analog components without getting bogged down in the intricate details of their internal workings [18]. These languages provide features for Describing Continuous-Time Behavior, Modeling Complex Relationships and Creating Hierarchical Models.

By using these behavioral modeling and abstraction techniques, engineers can simplify the verification of mixed-signal designs without sacrificing accuracy. It's like having the right map and the right language to navigate the complex terrain of mixed-signal systems.

Metric-Driven Verification (MDV)

MDV involves defining specific metrics and coverage goals to guide the verification process. This helps ensure that the verification effort is focused and comprehensive. Let’s take an example to understand how MDV helps with mixed signal verification. Assuming there is a Phase-Locked Loop (PLL) in the design and this needs to be verified. A circuit that generates a stable output clock signal. Here's how MDV can help [19].

Let’s first define the goal of verification, we want to ensure the PLL locks quickly to the input frequency, maintains a stable output frequency, and has low jitter (timing variations). Second is to establish metrics, for eg: Lock time, Output Frequency Stability, Jitter analysis these need to be defined in the first place. Lastly, monitor the previously mentioned metrics throughout the verification process. If you find that the jitter is excessive under certain conditions, you investigate further and potentially adjust the PLL design or parameters.

The Challenge of Verifying the "Perfect Eye"

What is the "Perfect Eye"? Imagine a signal traveling down a high-speed communication channel, like a stream of data flowing through a fiber optic cable. This signal is a series of pulses representing digital ones and zeros. If you visualize these pulses on an oscilloscope, they form a pattern that resembles an eye – hence the term “eye diagram”.

A "perfect eye" is an eye diagram that is wide open and clearly defined, with no distortions or overlaps between the pulses. This indicates a clean, strong signal with minimal noise and jitter (timing variations). It's essential for reliable data transmission at high speeds. This has always been challenging to verify, several factors make achieving this "perfect eye" a significant challenge:

High Frequencies

High-speed serial links operate at incredibly high frequencies, often in the gigahertz range. This makes it difficult to accurately simulate and analyze the signals, pushing the limits of available tools and techniques [20].

Signal Integrity

Maintaining signal integrity across the entire communication channel is crucial. Any reflections, crosstalk, or distortions can close the eye and lead to data errors [21].

Jitter and Noise

Jitter, or variations in the timing of the signal edges, and noise, which can distort the signal shape, are major enemies of the "perfect eye." Accurately modeling and mitigating these effects is essential [22].

Complex Interactions

High-speed serial links often involve complex interactions between analog and digital components, such as transmitters, receivers, equalizers, and clock recovery circuits. Verifying these interactions requires sophisticated simulation and analysis techniques [23].

Verifying the "perfect eye" requires a multi-pronged approach, advanced simulation where the tools are used to model the behavior of the high-speed link, including the effects of signal integrity, jitter, and noise. Precise measurements are taken on the simulated or prototyped hardware to analyze the eye diagram and extract key parameters like jitter and eye opening. The design of the high-speed link is carefully optimized to minimize signal degradation and ensure a clean eye diagram. This might involve adjusting the physical layout, selecting appropriate components, and using equalization techniques [24].

Emerging Trends

As mixed-signal designs grow increasingly complex, traditional simulation methods can sometimes hit their limits. That's where these emerging trends come in, offering exciting new possibilities for faster and more comprehensive verification. Prominent areas are captured below.

Emulation and FPGA Prototyping

Specialized hardware platforms called emulators can mimic the functionality of a chip at a high level. While not as fast as the final chip, they are significantly faster than software simulations, allowing for more extensive testing [25]. Field-Programmable Gate Arrays (FPGAs) are reconfigurable chips that can be programmed to behave like your design. This allows you to run your design at speeds closer to its intended operation and interact with real-world hardware [26]. The advantage of using either one of these prototypes is that they allow you to run much longer and more complex tests, which is crucial for mixed-signal designs with their intricate interactions between analog and digital components. They provide a more realistic environment for testing, capturing hardware-specific behavior and interactions with external components. They enable early integration of hardware and software, allowing you to test the entire system before the final chip is available.

AI/ML for Verification

Machine learning algorithms can be used to analyze simulation data, identify anomalies, and even generate test cases, potentially accelerating the verification process [27]. Some ways AI/ML can be used are Anomaly Detection, where AI/ML algorithms can analyze simulation results to identify unusual behavior or potential bugs that might be missed by traditional methods. It's like having a detective that can spot subtle clues and inconsistencies. Another scenario is for test

Case Generation where AI/ML can be used to automatically generate test cases that target specific areas of the design or cover corner cases that are difficult to identify manually. This can significantly accelerate the verification process. Lastly, another important factor is performance optimization, where AI/ML can help with analyzing simulation data to identify areas where the design can be optimized for better performance or lower power consumption.

Conclusion

This paper has provided a comprehensive exploration of the state-of-the-art in functional verification methodologies for mixed-signal designs. We delved into the unique challenges posed by these intricate circuits, where the analog and digital worlds intersect, and examined a range of powerful techniques used to ensure their correct operation. From advanced simulation methods that bridge the gap between continuous and discrete signals to formal verification techniques that provide mathematical certainty, we've covered a wide spectrum of approaches. We've also highlighted the importance of metric-driven verification in guiding this complex process and explored emerging trends like emulation, FPGA prototyping, and the application of AI/ML. While significant progress has been made in mixed-signal verification, ongoing research is essential to address the evolving complexities of these designs. By continuing to push the boundaries of mixed-signal verification, we can ensure that the next generation of integrated circuits will meet the demands of an increasingly interconnected and technology-driven world.

Reference

1. Roy A, Das D (2018) Mixed-signal design and simulation challenges: A survey. Journal of Semiconductor Technology and Science 18: 485-500.
2. Montoye RK (2001) Verification of mixed-signal integrated circuits. Proceedings of the IEEE 89: 537-553.
3. Salem A (1996) Challenges in mixed-signal verification. Mentor Graphics White Paper 2017.
4. Man HD (1996) Synchronization problems and solutions in mixed-signal simulation. Proceedings of the Design Automation Conference 264-269.
5. Kundert K (2006) Simulation of oscillators and other feedback loops. Designer's Guide Community.
6. Vladimirescu A (1994) The SPICE book. John Wiley & Sons.
7. Razavi B (2001) Design of analog CMOS integrated circuits. McGraw-Hill Education.
8. Terman CJ (1984) Simulation of VLSI circuits. Advances in CAD for VLSI 3: 167-227.
9. Gullapalli K (2003) Accuracy issues in mixed-signal simulation. Cadence Design Systems White Paper.
10. Chua LO, Lin PM (1975) Computer-aided analysis of electronic circuits: algorithms and computational techniques. Prentice-Hall.
11. Hodges AD, Jackson HG (2004) Analysis and design of digital integrated circuits. McGraw-Hill.
12. 12. Kundert K (1999) Introduction to RF simulation and its application. IEEE Journal of Solid-State Circuits 34: 1298-1319.
13. 13. (2022) Mixed-Signal Simulation in the 21st Century. EE Journal.
14. 14. Walter D (1007) Formal verification in a mixed-signal design flow. Proceedings of the International Symposium on Quality Electronic Design 182-187.
15. 15. Fujita M (2007) Equivalence checking for mixed-signal designs. Proceedings of the Asia and South Pacific Design Automation Conference 641-646.
16. 16. Hamad HA, Salem A (2006) Formal Verification of Analog and Mixed Signal Designs: Survey and Comparison. Proc. IEEE Northeast Workshop on Circuits and Syst (NEWCAS) 29-32.
17. 17. Michael C, Rhodes DL (1992) Real number modeling for mixed-signal simulation. Proceedings of the Custom Integrated Circuits Conference 497-500.
18. 18. Frey P (2000) Analog behavioral modeling languages: Verilog-AMS and VHDL-AMS. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition 976-981.
19. 19. Haedicke F (2005) Metric-driven verification for analog and mixed-signal designs," Proceedings of the Design Automation Conference 19-24.
20. 20. Sanders A (2003) High-speed signal propagation: advanced black magic. Prentice Hall PTR 2003.
21. 21. Bogatin E (2004) Signal integrity simplified. Prentice Hall Professional.
22. 22. Li MP (2007) Jitter, noise, and signal integrity at high-speed. Prentice Hall PTR.
23. 23. Palmer R (2003) Understanding high-speed serial links. EDN Magazine.
24. 24. Hall SH, Hall GW, McAllister JA (2000) High-speed digital system design: a handbook of interconnect theory and design practices. John Wiley & Sons.
25. 25. Bergeron J (2013) Writing testbenches: functional verification of HDL models. Springer Science & Business Media.
26. 26. Brown S, Vranesic Z (2008) Fundamentals of digital logic with Verilog design. McGraw-Hill Education.
27. 27. Mishra V (2018) Machine learning for functional verification. Proceedings of the International Conference on Computer-Aided Design 1-8.