Journal of Research in VLSI Design Tools and Technology
Volume 1 Issue 3, September-December 2016

Design Automation for Mixed-Signal, Analog, and Multi-Domain Chips

Raghuveer Sriwastav[1], Kapil kant Tiwari[2]
Associate Professor, Assistant Professor
Department of Electronics and Communication Engineering
Vardhan Engineering College, India
Email: *raghuveersriwastav21@gmail.com[1], tiwari13kapilkant@yahoo.com[2]*

Abstract

The growing demand for high-performance, low-power, and compact electronic systems has driven the rapid integration of digital, analog, radio-frequency (RF), power, and sensor interfaces onto a single chip. Such systems, commonly referred to as mixed-signal and multi-domain chips, are fundamental to applications ranging from mobile devices and automotive electronics to biomedical instrumentation and industrial automation. However, the design of these chips poses significant challenges due to the coexistence of continuous-time analog behavior and discrete-time digital logic, coupled with strong interactions across voltage, power, thermal, and timing domains. Traditional electronic design automation (EDA) tools have been largely optimized for digital design, leaving analog and mixed-signal (AMS) design heavily reliant on manual expertise. This paper presents a comprehensive review of design automation techniques for mixed-signal, analog, and multi-domain chips. It discusses the evolution of AMS design flows, modeling and abstraction strategies, synthesis and optimization methods, verification and validation challenges, and emerging trends such as machine learning–assisted automation and multi-physics co-design. By highlighting both academic research and industrial practices, the paper aims to provide a consolidated perspective on the current state and future directions of AMS design automation.

Keywords: *Mixed-signal design, analog automation, multi-domain chips, EDA tools, AMS verification, design optimization*

Introduction

The semiconductor industry has witnessed an unprecedented level of integration over the past two decades. While early integrated circuits were predominantly digital, modern systems-on-chip (SoCs) increasingly incorporate analog front-ends, RF transceivers, power management units, sensors, and high-speed interfaces alongside complex digital logic. This convergence has given rise to mixed-signal and multi-domain chips, where multiple signal types and physical domains interact closely.

Despite the success of digital design automation, analog and mixed-signal design remains a major bottleneck in terms of productivity, time-to-market, and design cost. Analog circuits are sensitive to device-level variations, layout parasitics, and environmental factors, making them difficult to capture using high-level abstractions. As a result, analog designers still rely heavily on manual sizing, simulation-based optimization, and experience-driven heuristics.

Design automation for mixed-signal and multi-domain chips seeks to bridge this gap by introducing systematic methodologies, reusable models, and intelligent tools that can handle the inherent complexity of such systems. This paper reviews the key concepts, challenges, and solutions in this area, with a focus on practical design flows and emerging automation techniques.

Characteristics of Mixed-Signal and Multi-Domain Chips

Mixed-signal and multi-domain chips represent a class of integrated circuits where different signal types, operating principles, and physical effects coexist on the same silicon die. Unlike purely digital systems, these chips must correctly process both continuous-time and discrete-time signals while operating reliably across multiple interacting domains. Understanding their characteristics is essential for developing effective design automation strategies.

Mixed-Signal Integration

Mixed-signal integration refers to the combination of analog and digital circuitry on a single semiconductor substrate. Digital blocks such as processors, controllers, and memory operate on binary logic levels and are typically optimized for speed, area, and power efficiency. In contrast, analog blocks process continuous-valued signals and are designed to meet stringent performance requirements such as gain, linearity, noise, bandwidth, and stability.

Common mixed-signal building blocks include analog-to-digital converters (ADCs), digital-to-analog converters (DACs), phase-locked loops (PLLs), voltage regulators, sensor readout circuits, and high-speed serial transceivers. These components form critical interfaces between the physical world and digital processing units.

One of the main challenges in mixed-signal integration is the strong coupling between analog and digital sections. Digital switching activity generates substrate noise, supply voltage fluctuations, and electromagnetic interference, which can severely degrade the performance of sensitive analog circuits. Clock jitter originating from digital logic can directly impact timing accuracy in PLLs and data converters. Additionally, differences in layout styles, device sizing, and routing requirements further complicate integration.

From a design automation perspective, mixed-signal integration is difficult because analog behavior is highly dependent on physical implementation details. Small changes in layout parasitics or supply noise can cause noticeable performance shifts, making it hard to rely solely on abstract models. As a result, automated tools must carefully balance abstraction with accuracy while enabling effective co-design of analog and digital blocks.

Multi-Domain Nature

Beyond the coexistence of analog and digital signals, modern chips operate across multiple interacting domains, often referred to as multi-domain systems. Each domain introduces its own set of constraints, design objectives, and sources of variability, and their interactions significantly influence overall chip behavior.

The **power domain** is characterized by the use of multiple supply voltages, voltage islands, and power gating techniques to reduce energy consumption. Analog circuits may require stable, low-noise supplies, while digital logic often operates under aggressive voltage scaling. Fluctuations or coupling between power domains can lead to functional failures or performance degradation, especially in mixed-signal blocks.

The **timing domain** includes synchronous clocked logic, asynchronous interfaces, and multiple clock frequencies within the same chip. Clock domain crossings, jitter accumulation, and skew management are critical concerns. Analog circuits such as PLLs and clock generators directly interact with digital timing structures, making timing analysis a cross-domain problem rather than a purely digital task.

The **thermal domain** has become increasingly important with technology scaling and higher integration density. Localized heating caused by high switching activity or power-hungry blocks can alter transistor characteristics, affecting analog parameters such as offset, gain, and noise. Temperature gradients across the die also impact reliability and long-term performance, requiring thermal-aware design and verification.

The **process domain** accounts for manufacturing variations arising from fabrication imperfections. Variability in transistor dimensions, threshold voltages, and interconnect parameters leads to performance spread across process corners. Analog circuits are particularly sensitive to such variations, making robust design and extensive statistical verification essential.

These domains do not operate independently. For example, power consumption influences temperature, which in turn affects timing and analog performance. Similarly, process variations can exacerbate noise sensitivity or timing uncertainty. Because of these complex interactions, mixed-signal and multi-domain chips demand holistic design, simulation, and verification approaches. Design automation tools must therefore support cross-domain analysis and optimization rather than treating each domain in isolation.

Evolution of Design Automation for AMS Systems

Design automation for analog and mixed-signal (AMS) systems has evolved much more slowly compared to its digital counterpart. Early electronic design automation (EDA) tools were primarily developed to support digital integrated circuits, where behavior can be represented using discrete logic levels and timing models. Digital synthesis, place-and-route, and static timing analysis benefited from well-defined abstractions and scalable algorithms, enabling a high degree of automation and repeatability.

In contrast, analog circuit behavior is continuous in nature and strongly dependent on device physics, layout parasitics, and operating conditions. As a result, early analog design relied heavily on manual calculations, hand-crafted schematics, and iterative SPICE simulations. The absence of standardized representations for analog intent and performance specifications further limited the applicability of automation techniques. Analog designers traditionally optimized circuits using experience-driven heuristics, which were difficult to formalize within EDA tools.

A significant milestone in AMS automation was the **introduction of hardware description languages with analog extensions**, such as Verilog-A and VHDL-AMS. These languages enabled designers to describe analog behavior using mathematical equations and continuous-time constructs, while still supporting interaction with digital logic. Behavioral modeling using these languages made it possible to perform early functional verification and system-level simulations without relying entirely on transistor-level descriptions. However, ensuring consistency between behavioral models and final implementations remained a challenge.

The next major advancement came with the **development of mixed-signal simulators** capable of co-simulating digital event-driven models and analog continuous-time waveforms. These simulators allowed designers to verify interactions between analog and digital blocks within a unified environment. For example, the impact of digital switching noise on an analog front-end or the effect of clock jitter on data converter performance could be analyzed more realistically. While powerful, mixed-signal simulation introduced new challenges related to simulation speed, convergence, and scalability, especially for large system-on-chip designs.

Another important step in the evolution of AMS automation was the **emergence of constraint-driven analog synthesis and layout tools**. Instead of manually sizing transistors, designers could specify performance constraints such as gain, bandwidth, noise, and power consumption. Optimization engines then explored the design space to identify suitable device parameters and, in some cases, generate preliminary layouts. These tools improved productivity and helped reduce design iteration cycles, particularly for well-understood circuit topologies. Nevertheless, their effectiveness was often limited by the quality of constraints and the availability of accurate models.

Despite these advances, **full automation of analog and mixed-signal design remains an open research problem**. Unlike digital flows, analog design lacks universally accepted abstraction levels and reusable IP standards. Human intuition and domain expertise are still critical in topology selection, trade-off analysis, and layout refinement. Current AMS automation approaches are therefore best viewed as assistive tools that augment designer productivity rather than completely replacing manual design efforts. Ongoing research continues to explore higher-level abstractions, learning-based optimization, and multi-domain co-design frameworks to further close the automation gap between digital and analog systems.

Modeling and Abstraction Techniques

Modeling and abstraction play a crucial role in the design automation of mixed-signal and multi-domain systems. Due to the inherent complexity of analog behavior and cross-domain interactions, it is impractical to perform all design tasks at the transistor level. Abstraction techniques allow designers to represent circuit functionality at different levels of detail, enabling faster simulation, early validation, and efficient design space exploration. However, selecting the appropriate abstraction level remains a key challenge, particularly for analog and mixed-signal blocks.

Behavioral Modeling

Behavioral modeling focuses on describing the functional behavior of a circuit without explicitly modeling its internal transistor-level structure. These models capture input–output relationships using mathematical equations, state machines, or signal flow descriptions. Hardware description languages such as Verilog-A and VHDL-AMS are commonly used to implement behavioral models for analog and mixed-signal components.

Behavioral models are extensively used during early design stages for system-level validation and architectural exploration. For example, an ADC may be modeled using ideal quantization and noise equations to evaluate system performance before the actual circuit topology is finalized. Similarly, phase-locked loops and voltage regulators are often modeled behaviorally to study stability, jitter, and transient response in the presence of digital control logic.

The main advantage of behavioral modeling is simulation efficiency. Since these models avoid detailed device-level computations, they significantly reduce simulation time and allow designers to analyze large systems that would otherwise be infeasible to simulate. Behavioral models also enable rapid design iteration and early detection of functional errors.

However, maintaining accuracy across different operating conditions is a major limitation. Behavioral models may not fully capture non-linearities, parasitic effects, or sensitivity to process, voltage, and temperature variations. As a result, discrepancies can arise between behavioral simulation results and transistor-level performance. Careful model calibration and validation are therefore necessary to ensure meaningful results.

Hierarchical Abstraction

Hierarchical abstraction is a widely adopted strategy in mixed-signal design that decomposes a complex system into smaller, manageable blocks. Each block is modeled at an abstraction level appropriate to its function and design maturity. This approach helps control complexity while allowing designers to focus on critical system interactions.

In a typical mixed-signal system, digital blocks are described at the register-transfer level (RTL) using languages such as Verilog or VHDL. These blocks benefit from mature digital design automation flows, including synthesis and static timing analysis. Analog blocks, on the other hand, are often represented using a combination of behavioral models and schematic-level descriptions, depending on the required accuracy.

Hierarchical abstraction enables mixed-level simulation, where high-level models coexist with detailed circuit implementations. For example, a complete system may be simulated using behavioral models for most analog blocks, while one critical amplifier is represented at the transistor level for detailed performance evaluation. This flexibility significantly reduces simulation cost while preserving accuracy where it matters most.

Despite its advantages, hierarchical abstraction introduces challenges related to model consistency and interface definition. Ensuring that signals, timing assumptions, and operating conditions are aligned across abstraction levels is non-trivial. Inconsistent assumptions can lead to misleading simulation results and integration issues later in the design cycle.

Multi-Domain Modeling

Modern mixed-signal systems are increasingly influenced by interactions across multiple physical domains, including electrical, thermal, and sometimes mechanical domains. Multi-domain modeling frameworks aim to capture these interactions within a unified representation, enabling more accurate analysis and optimization.

Electrical-thermal modeling is particularly important for power management integrated circuits (PMICs) and high-density SoCs. Power dissipation in digital blocks leads to localized temperature rise, which in turn affects transistor parameters such as mobility and threshold voltage. These thermal effects can degrade analog performance, alter timing behavior, and reduce long-term reliability.

In sensor-based systems and micro-electro-mechanical systems (MEMS), mechanical and electrical domains are tightly coupled. For example, capacitive sensors rely on mechanical movement that directly influences electrical characteristics. Multi-domain models are essential for predicting system behavior under real-world operating conditions.

From a design automation perspective, multi-domain modeling increases computational complexity and modeling effort. Accurate cross-domain models are often difficult to develop and validate. Nevertheless, they are critical for early detection of performance issues that cannot be identified through single-domain analysis. As mixed-signal systems continue to grow in complexity, multi-domain modeling is expected to become an integral part of future design automation flows.

Analog and Mixed-Signal Synthesis

Constraint-Driven Design

Analog synthesis tools rely on designer-specified constraints such as gain, bandwidth, noise, and power. Optimization algorithms then search the design space to determine suitable device sizes and topologies.

Topology Selection

Selecting an appropriate circuit topology is one of the hardest tasks to automate. Research approaches include template-based synthesis and knowledge-driven methods derived from expert rules.

Optimization Algorithms

Common optimization techniques include:

- Gradient-based methods
- Genetic algorithms
- Simulated annealing

These methods balance conflicting objectives such as power, area, and performance.

Physical Design and Layout Automation

Layout plays a critical role in analog performance due to parasitic effects and device matching requirements. Unlike digital layouts, analog layouts must consider symmetry, common-centroid placement, and routing constraints.

Recent automation efforts focus on:

- Parameterized layout generators
- Constraint-aware placement
- Automated parasitic extraction and back-annotation

Verification and Validation

Mixed-Signal Simulation

Mixed-signal simulation integrates event-driven digital simulation with continuous-time analog solvers. While accurate, such simulations can be computationally expensive for large designs.

Formal and Semi-Formal Methods

Formal verification techniques are being explored to complement simulation by proving certain properties of analog behavior within bounded conditions.

Corner and Monte Carlo Analysis

Process, voltage, and temperature variations are evaluated using extensive corner and Monte Carlo simulations. Automation tools aim to reduce the number of required simulations through intelligent sampling.

Multi-Domain Co-Design and Optimization

Multi-domain co-design considers interactions across power, thermal, and timing domains during early design stages. For example, power-aware analog design must account for voltage scaling effects on gain and noise.

Table 1: Domains in Mixed-Signal and Multi-Domain Chips

Domain	**Key Parameters**	**Design Challenges**
Electrical	Voltage, current, noise	Signal integrity
Power	Supply levels, gating	Energy efficiency
Thermal	Temperature gradients	Reliability
Timing	Clocks, delays	Synchronization

Machine Learning in AMS Design Automation

Machine learning (ML) has recently gained attention as a promising approach to analog automation. ML models can learn complex relationships between design parameters and performance metrics, enabling faster optimization.

Applications include:

- Performance prediction
- Design space exploration
- Yield optimization

While promising, ML-based methods require high-quality training data and careful validation.

Case Study: Automated Design Flow for a Mixed-Signal SoC

A typical automated flow for a mixed-signal SoC includes:

1. System-level modeling and partitioning
2. Behavioral verification
3. Constraint-driven analog synthesis
4. Digital synthesis and physical design
5. Mixed-signal verification and sign-off

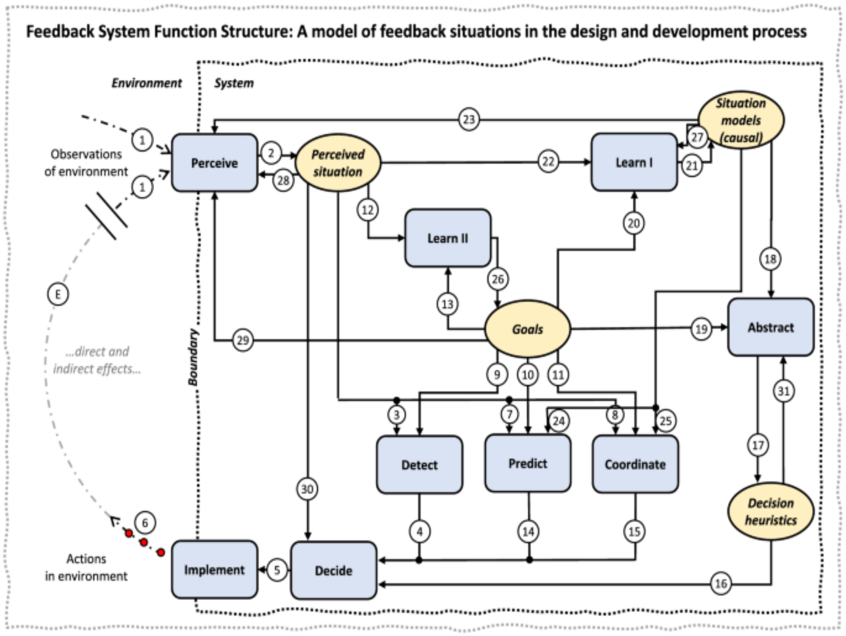

Figure 1 (conceptual) shows an overview of such a design flow, highlighting feedback loops between synthesis and verification stages.

Challenges and Open Research Issues

Despite progress, several challenges remain:

- Limited robustness of analog abstractions
- Scalability of mixed-signal simulation
- Integration of multi-physics effects
- Trust and acceptance of automated analog tools by designers

Addressing these issues requires collaboration between academia and industry.

Future Trends

Future AMS design automation is expected to emphasize:

- Higher-level system abstractions
- Greater use of AI-driven tools
- Cloud-based simulation and optimization
- Standardized analog IP reuse

These trends aim to significantly reduce design cycles and improve first-pass silicon success.

Discussion

Design automation for mixed-signal and multi-domain chips is not merely a technical challenge but also a cultural shift in design practices. Analog designers must adapt to more automated flows, while tool developers must capture decades of design intuition into usable algorithms. Incremental automation, rather than full replacement of human expertise, appears to be the most realistic path forward.

Conclusion

Mixed-signal, analog, and multi-domain chips form the backbone of modern electronic systems, yet their design remains complex and labor-intensive. Design automation offers a pathway to manage this complexity by introducing structured methodologies, intelligent optimization, and comprehensive verification strategies. This paper reviewed the key aspects of AMS design automation, including modeling, synthesis, layout, verification, and emerging ML-based approaches. While significant challenges remain, ongoing research and industrial innovation continue to push the boundaries of what can be automated. Ultimately, effective design automation will be essential for sustaining innovation in next-generation semiconductor systems.

References

1. Rutenbar, R. A., Gielen, G., & Roychowdhury, J., “Hierarchical modeling, optimization, and synthesis for analog and mixed-signal design,” *IEEE Transactions on CAD*, 2009.
2. Kundert, K., “Designing analog circuits in the deep submicron era,” *IEEE Design & Test*, 2010.
3. Gielen, G., & Sansen, W., *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Springer, 1991.
4. Vlach, J., & Singhal, K., *Computer Methods for Circuit Analysis and Design*, Springer, 2003.
5. Nagaraj, K., et al., “Analog circuit design automation: Past, present, and future,” *Integration, the VLSI Journal*, 2014.
6. Martins, R., & Mak, P., “Analog IC design automation in nanoscale CMOS,” *Microelectronics Journal*, 2016.

7. Huang, Q., et al., “Machine learning for analog circuit design: A survey,” *IEEE Access*, 2020.
8. Kundert, K., “Predicting the phase noise and jitter of PLL-based frequency synthesizers,” *IEEE Transactions on CAD*, 2018.
9. Goyal, S., & Gupta, S., “Multi-domain modeling and co-design for mixed-signal systems,” *Journal of Low Power Electronics*, 2019.
10. Chen, Y., et al., “Challenges and opportunities in analog and mixed-signal EDA,” *IEEE Design & Test*, 2021.