Mixed-Signal Design Challenges and Requirements

By Eric Naviasky, Design Services and Mladen Nizic, Solutions Marketing, Cadence Design Systems, Inc

Current market and technology trends have increased the demand for mixed-signal integration and brought many challenges. One challenge is the scalability and performance of analog/mixed-signal intellectual property (IP) at advanced process nodes. Another is system-on-chip (SoC) integration and increasing demands on verification, physical integration, and signoff. Adding to these challenges are aggressive cost targets that affect packaging options. This paper describes these mixed/signal design challenges and introduces advanced solutions, flows, and methodologies from Cadence to address them.

Introduction

Success in today’s electronics marketplace requires highly integrated, low-cost solutions for wireless, consumer, computer, and automotive applications. At the same time, advanced process nodes now make it possible to manufacture analog and RF circuits down to 45 nm and below. Consequently, analog and mixed-signal IP content is significantly increasing in system-on-chip (SoC) devices that in the past contained mostly digital circuitry. This situation creates new challenges for design, integration, and verification.

Most SoCs currently being developed have analog/mixed-signal blocks such as SerDes cores, UARTs, DACs, ADCs, PLLs and other transceivers. Since analog doesn’t scale as well as digital, these blocks may represent a substantial portion of the SoC. Moreover, many so-called analog blocks actually have digital control logic. As such, an increasing amount of “analog” IP is really mixed-signal, and with rapidly increasing SoC capacity, a single IP block may represent an extremely complex mixed-signal function.

The hypothetical SoC in Figure 1 below shows analog blocks in white and digital blocks in orange. Light orange represents mixed-signal blocks with both analog and digital circuitry. In earlier process generations, some of these mixed-signal blocks may have been entire chips.
Figure 1: Many blocks on SoCs are analog or mixed-signal.

The increasing demand for mixed-signal integration results in many challenges. One is the scalability and performance of analog/mixed-signal IP at advanced process nodes. Another challenge is SoC integration and the demands it places on verification, physical integration, and signoff. On top of this, aggressive cost targets require careful consideration of test and packaging alternatives, often making previous packaging options unaffordable luxuries.

This whitepaper describes existing and emerging mixed/signal design challenges and requirements in several key areas, including:

- Mixed/signal IP and SoC verification
- Behavioral modeling
- Design for test
- Chip planning
- Substrate noise analysis
- IP block implementation and signoff
- Chip integration and signoff
- Data management
- Design collaboration
- IC/package co-design

Subsequent whitepapers will present methodologies and tools that can help solve many of these challenges.

**Mixed/Signal IP and SoC Verification**

Due to increasing complexity and shrinking feature sizes, mixed-signal verification is becoming much more difficult. Fast Spice and parallelized simulation engines help push capacity and performance limits, but these technologies alone do not enable verification teams to catch up with growing verification needs. What’s needed is a hierarchical approach that uses behavioral models that make it possible to mix different levels of abstraction.

As a first step, techniques that have been readily adopted in digital design—such as assertions, verification planning, and metric-driven verification—need to make greater inroads into mixed-signal verification. Coverage metrics should be extended to the functional and electrical domains of mixed-signal IP in the context of the SoC they are being integrated into. This will help to drive the integration process with increased designer confidence and a reduced number of ECOs and silicon respins.
Verification strategies also need to reflect an understanding of the two types of errors that challenge the design team—“highly embarrassing” errors such as inverted or misconnected signals, and “technically interesting” errors such as those caused by signal integrity or device physics problems. Of course the second occurrence of a “technically interesting” error changes the classification to “highly embarrassing.” The best way to avoid embarrassing errors is to use the highly structured verification plans, tools and frameworks that have been developed for digital SoCs.

**Behavioral Modeling**

Functional verification relies very heavily on behavioral modeling. For many designers, the move to behavioral modeling requires both a methodology shift and new skills. Models need to be developed and used with a specific purpose in mind. Trying to make behavioral models as accurate as transistor-level models will result in slower simulation. Additionally, generating, validating and maintaining models requires an up-front investment. Models must be synchronized with circuit design and layout, and they need to be re-verified if any changes are made to the IP.

Verifying digitally-controlled analog circuits in mixed-signal blocks for many different modes of operation represents another challenge. Low power techniques—such as multi-voltage domains and power shutoff—are frequently used within mixed-signal blocks, making verification even more challenging. One way to overcome this challenge is to use formal verification methods that can use power specifications across mixed-signal SoC boundaries. The Common Power Format (CPF) should be used to represent digital design intent.

Full-chip static timing analysis (STA) is often not accurate or comprehensive enough for mixed-signal SoCs because design teams generate .lib models for their mixed-signal blocks. These models can be inaccurate as well as insufficient to perform a detailed analysis of the digital logic inside mixed-signal blocks.

**Design for Test (DFT)**

It is not uncommon for the test cost of a modern SoC to greatly exceed the silicon cost. A common requirement is that the device be testable on a 200 MHz digital tester, regardless of the SoC I/O or internal speed and regardless of the functions and mixed-signal content. The device must basically test itself, and in many cases implement redundancy so it can repair itself.

The economics of mixed-signal test are daunting, with test time on a cheap tester running at $0.05 per second and with the increased content of the chip driving up the number of vectors that must be run. Built-in self test (BIST), which typically increases area by 10 percent, is often used for analog functional tests. To self-test the mixed-signal sections, the chip must include signal generators and measuring circuits. Consequently, the number of modes that must be verified increases considerably. To have a SoC that can basically test itself, it is necessary to plan from the very beginning. Self-testing can not be added as an afterthought.

BIST engines are often custom developed for specific analog circuits. For example, SerDes blocks often come with a BIST capability that can run applicable tests, such as bit error rate (BERT) tests. When BIST engines are placed in multiple blocks, or shared among identical blocks, the designer must hook up all of the various BIST blocks to a chip-level interface such as an IEEE 1149.1 test access port (TAP) controller. Applying BIST tests for different types of analog blocks requires translating the BIST setup and application sequences to the chip test interface. Design for Test (DFT) automation can hook the BIST engines up to the chip test interface, translate the BIST setup, and run sequences from the BIST engine ports to the chip test interface ports.

**Chip Planning**

Mixed-signal SoCs require very careful consideration of the floorplan. To make the right decisions regarding placement of I/Os, the designer needs to take into account analog, mixed-signal and RF blocks, as well as routing, power, timing, noise and other constraints. This is necessary to minimize area and avoid problems that could cause silicon respins or create difficulties throughout implementation.

For example, in designs requiring bus routing or high-speed digital and sensitive analog nets, components must be placed to enable sufficient routing resources and avoid congestion or signal integrity problems. Similarly, special care needs to be given to placement of sensitive analog blocks, because noise from digital switching propagated through package, substrate, power or signal net coupling can impact the performance of these blocks. Without tools that are capable of preventing these kinds of problems by analyzing different floorplan configurations, designers often increase margins and waste chip area by using excessive isolation.
This means the floorplan needs to be done before blocks are created—even though the block sizes are unknown. Using estimates of block sizes with an understanding of the signal flow through the system is the first step in optimally placing the blocks to avoid signal integrity problems. If problems are unavoidable, then this is the time to design the shielded channels for critical signals. The channels are laid out, extracted and used to provide models for the designer to use in the design process.

Additionally, power domains and routing need to be overlaid on the preliminary floorplan to allow proper forecasting of supply noise (self and external). This forecasting is also used to establish conditions for the block design, and allows users to plan the location and number of decoupling cells that need to be placed in the floorplan.

Timing-driven floorplanning requires models for mixed-signal blocks. If the models are not provided by mixed-signal designers, the SoC designer will need to create them manually, using the original specification. This might be sufficient for the floorplanning stage, but more accurate models are required for physical implementation and signoff. In general, well-characterized mixed-signal IP with models for timing, power, and noise, as well as constraints for its integration into an SoC, are required and should be provided by the same mixed-signal designer that designed the IP.

**Substrate Noise Analysis**

In addition to crosstalk, mixed-signal SoCs pose an increased risk of chip malfunction due to coupling through the substrate. Substrate noise injection and propagation need to be modeled to identify the magnitude and frequency of aggressors for noise-sensitive circuits. Designers need to use sensitivity analysis to determine if floorplan modification or isolation is required.

A conservative approach, which typically involves heavy use of guard rings or a triple-well process, costs area and could introduce unwanted coupling paths, thereby making it counterproductive. A better approach is to model aggressors and analyze victim sensitivity in order to make good tradeoffs and mitigate risk.

**IP Block Implementation and Signoff**

Once the testability is assured, the floorplanning is done and the noise and supply conditions are set, it is time to do the block design. Implementing analog or digital blocks is usually straightforward. IP blocks containing substantial amounts of standard cell logic together with analog or RF circuits are more challenging, particularly where it is not possible to meet design targets by partitioning analog/RF and digital functionality into sub-blocks and implementing them separately.

Analog and digital IP blocks are often treated as “black boxes” with fixed layouts and pinouts. Integrating these blocks into SoCs is becoming an inefficient methodology.

Instead, designers need a unified design environment with a common database, constraint sharing, and the ability to apply semi-automatic design refinement for analog blocks and full automation for digital blocks.

Additionally, IP blocks must be verified against specifications and modeled accurately for timing, power, and noise. Some signoff tasks are increasingly being pushed to the block level, since the huge size of today’s SoCs makes full chip signoff time consuming. Therefore it is important that IP blocks not just meet timing and power specs, but also that they are signal integrity, DRC, litho and CMP clean using full-chip signoff criteria. Block designers must also consider the boundary conditions that are expected in SoC integration. Integration-ready IP with all necessary models and views will help ensure smooth SoC integration, and ideally make full-chip signoff a one-step task.
Chip Integration and Signoff

In a traditional SoC, analog and mixed-signal circuitry used about 20% of chip area. Due to strong demand for higher integration, this has been steadily increasing, and in SoCs designed at advanced nodes it is common to find half of the chip area occupied by analog and mixed-signal blocks. Poor analog scaling, larger design margins, and more willingness to trade area to reduce the risk of respins also contribute to increases in the analog portion of the chip.

Top-level implementation and SoC integration are typically done using digital automation, and the SoC is then signed off using cell-based static timing, signal integrity and IR drop analyses. The increased analog content requires special care of top-level routing and analog/digital interfaces.

The SoC integrator must make sure that top-level optimization, buffer insertion, signal-integrity improvements and DRC fixes do not negatively impact sensitive analog routes and analog/digital interfaces. The quality of signoff for timing, signal integrity and power depends on the accuracy of models for the mixed-signal blocks. Even the most accurate black-box model might not be sufficient. Critical timing paths, for example, might span logic in multiple mixed-signal blocks. The most accurate way to verify and debug critical timing paths is to flatten all logic from the hierarchy of mixed-signal blocks, and to then run extraction and timing analysis. Transistor-level analysis on critical paths might be necessary for high-speed signals.

Once the analog and digital blocks are integrated, performing ECOs is much harder, and often requires going back to individual blocks to make necessary changes and re-do part of the integration work. Sometimes, small analog or digital ECOs are required after chip finishing steps like metal fill insertion. The ability to make such ECOs on the full chip without having to redo integration or chip finishing tasks would help keep many design projects on schedule.

Data Management

Analog and digital are increasingly being designed in parallel by teams located all around the world. Managing and maintaining the integrity of the huge amounts of data required for a mixed-signal SoC is not a simple task. To avoid unintended changes or corruption, a data management system must cover all tools and data involved in the process.

Furthermore, for mixed-signal blocks that are co-designed, keeping different views of the blocks synchronized often represents a challenge. If a block abstract, its layout, timing or power model are not synchronized when delivered to chip integrator, it will likely prolong turnaround time or result in a silicon respin.

Using a data management system, following agreed guidelines and naming conventions, and using checking utilities to verify completeness and data integrity are all important steps in enabling global teams to work on the same SoC productively.
Design Collaboration

In a traditional mixed-signal design, analog and digital groups worked well apart with clearly separated responsibilities. The analog designer would determine the chip’s architecture and floorplan, decide on locations for analog objects, carve out an area for the digital part of the chip including its pin location, and pass only this information to digital designer. After designing the logic in the given chip area, the digital designer would return layout data with a logic netlist and parasitics for integration and signoff. The handoff was well defined, and this “over the wall” approach worked for many mainstream mixed-signal applications on older process nodes.

Increased analog and digital integration at advanced process nodes requires a much closer collaboration between analog and digital designers throughout the project. For example, at the chip planning stage, tradeoffs have to be made with respect to package and I/O design, die size and area allocation, verification strategy, timing and power budgets, noise and other physical and electrical constraints. Designers need to be able to explore alternatives and arrive at an optimal chip architecture in order to ensure minimal die size and smooth verification and implementation. Similarly, at the chip integration stage, designers need to collaborate on achieving design closure and processing ECOs.

Figure 3: “Over the wall” approach versus a collaborative approach.

An effective design flow and methodology must provide flexibility and support to global design teams in their collaborative approach throughout the design process. A unified design database and design constraints, and well integrated flows for verification and implementation, are key to meeting designers’ needs.
IC/Package Co-Design

The integration of the package and the silicon into a single device is orders of magnitude more complex than it was in the past. Pin counts in excess of 1,000, pin speeds (for hundreds of pins) in excess of 3GHz, and power supply currents of tens of amps have made package and chip integration extremely challenging.

Increased integration and diverse applications call for different packaging models. Integrating multiple dies in the same package (multi-chip module, MCM) continues to offer an attractive solution. Integrating RF and digital dies into an MCM requires considerable care about the electrical and physical aspects of the integration due to higher frequencies and greater interaction between RF and digital parts, as well as the passive components on the die and in the package.

Another promising approach employs stacked die with through-silicon vias (TSVs). A technology called Silicon Interposer, which uses multiple dies on a silicon substrate connected using through-silicon vias, offers more flexibility compared to MCMs and is a likely intermediate step toward stacking multiple die on top of each other.

In either an MCM or TSV implementation, it is critical to model mutual inductive coupling across different technologies. Since IC and package designs are done in two or even three different design environments, it is important to be able to fully exchange design data and constraints across environments. This helps to minimize iterations between package and IC designers.

Summary

Market and technology trends have raised the bar for mixed-signal designs, causing some old challenges to reappear and new ones to emerge. Cadence® sees this as an opportunity to serve the market with the advanced tools, flows and methodologies that are needed to overcome these challenges.