Solutions for Mixed-Signal IP, IC, and SoC Implementation

By Peter McCrorie and Mladen Nizic, Cadence Design Systems, Inc.

Traditional mixed-signal design environments, in which analog and digital parts are implemented separately, are no longer sufficient and lead to excess iteration and prolonged design cycle time. Realizing modern mixed-signal designs requires new flows that maximize productivity and facilitate close collaboration among analog and digital designers. This paper outlines mixed-signal implementation challenges and focuses on three advanced, highly integrated flows to meet those challenges: analog-centric schematic-driven, digital-centric netlist-driven, and concurrent mixed-signal. Each flow leverages a common OpenAccess database for both analog and digital data and constraints, ensuring tool interoperability without data translation. Each flow also offers benefits in the area of chip planning and area reduction; full transparency between analog and digital data for fewer iterations and faster design closure; and easier, more automated ECOs, even at late stages of design.

Contents

Introduction

Virtually all systems on chip (SoCs) contain some analog and mixed-signal circuitry in the form of pre-designed blocks commonly referred to as intellectual property (IP). On the other side of spectrum are traditional analog designs that now integrate more and more digital logic to increase functionality or perform calibration. Whether at the IP, integrated circuit (IC), or SoC level, designers have to design, implement, and integrate significant portions of both analog and standard cell digital. The trend is blurring the boundaries between “analog” and “digital.” Consequently, mixed-signal SoC implementation today requires far more than the traditional practice of importing a few black boxes that were designed independently from each other.

This paper focuses on advanced mixed-signal implementation flows and solutions. It describes some of the common challenges that result from today’s disjointed analog and digital design flows. It then discusses different implementation flows and underlining support for them. Two other related Cadence white papers discuss mixed-signal design challenges and mixed-signal verification.

The Hierarchical Nature of Today’s Mixed-Signal Designs

Mixed-signal design has become hierarchical, with analog blocks residing inside digital functions that control analog signals. With the trend toward digitally assisted analog, designs such as phase-locked loops (PLLs) and analog/digital converters (ADCs) contain increasing amounts of digital control logic. Conversely, traditionally digital SoC designs today commonly incorporate high-speed I/Os, RF transceivers, audio/video interfaces, and power management units—each containing substantial analog content.

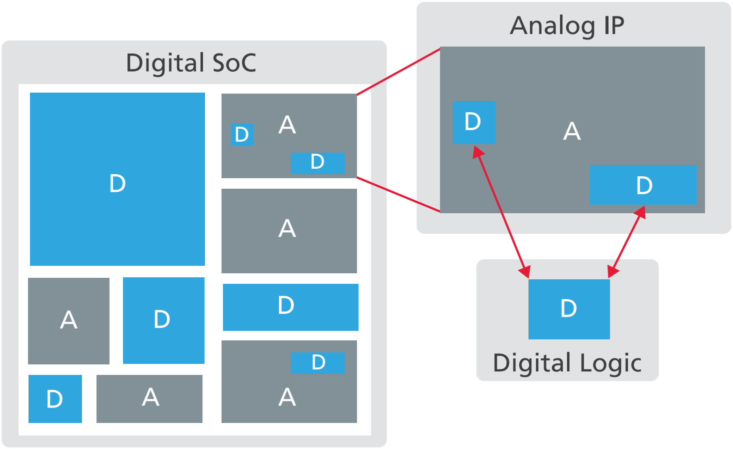

Figure 1 – Digital logic often resides inside analog IP that is integrated into SoCs

SoCs must interact with the outside world through interfaces integrated on-chip, including displays, sound devices, antennas, sensors, and others. A single IP block may represent a complex mixed-signal function that may have been an entire chip in a previous process generation. Due to increased integration and poor scaling of analog compared to digital, analog and mixed-signal blocks may take up half the area of a modern SoC.

Traditional physical implementation methodologies are either "netlist-driven" flows from a digital cockpit or "schematic-driven" flows from a separate analog cockpit. In both cases, these flows were developed to deal with relatively simple integration challenges. While these methodologies will remain important, SoC designers need a new approach as the amount of complex mixed-signal content grows, and analog and digital circuits become more functionally coupled.

In the new methodology, analog and digital blocks are designed concurrently, with flexible pin assignments so they can be placed and optimized within a true mixed-signal floorplan. While analog and digital designers still work in their own familiar design environments, analog and digital tools are integrated on a common database to enable full transparency into analog and digital layout. This simplifies the integration process tremendously. Responsibility for chip assembly and tapeout is shared. In short, there is no hard separation between "analog" and "digital" design.

Mixed-Signal Implementation Challenges

Mixed-signal SoC implementation is a complicated task because it requires collaboration among multiple design teams often using different design environments. For example, analog and digital blocks are designed by separate teams using entirely different tools, with little communication between the teams and little understanding of the other team's environment or challenges.

The analog design flow is typically schematic-driven, with many levels of physical hierarchy. It is largely transistor-based, requiring a level of detail that is mostly hidden on the digital side. While today's analog/custom design environments offer more automation, analog designers still prefer an interactive flow with a high level of user control.

The analog flow typically involves drawing schematics, functional simulation, drawing or editing polygons to implement the design, and then executing layout-versus-schematic (LVS) checking. In more advanced systems such as the Cadence® Virtuoso® Layout Suite, the analog flow often includes the use of parameterized cells (Pcells), automated device placement, and custom routing.

The analog flow makes heavy use of physical and electrical constraints. For example, analog layout may have constraints related to placement, symmetry, current/voltage limits, shielding, and more. To effectively interoperate, digital tools must understand the relevant subsets of these constraints as well, such as placement and routing constraints.

In contrast, the digital design flow is netlist-driven and highly automated. It starts from RTL synthesis or even higher SystemC™ synthesis to generate gate-level logic. Aimed at designing ICs with tens or hundreds of millions of gates, the flow employs cell libraries that hide transistor-level details from the designer. Floorplanning, placement, and routing are timing-driven and highly automated.

When analog blocks are imported into a predominantly digital SoC design, the blocks are typically black boxes giving digital designers no visibility into their layout. The analog blocks typically have fixed guard rings and pinouts. The hardened IP leads to a lack of flexibility in floorplanning, resulting in a less-than-optimal floorplan.

The SoC integration engineer has to assume that the analog designer adequately verified the block. It is not uncommon to discover trivial problems very late in the design cycle. For instance, during final full-chip design-rule checking, a designer finds out that a control signal is inverted because the block was never properly verified in the integration context. Even if all the digital and analog blocks are thoroughly verified separately at the block level, full chip-level verification including analog-digital interfaces must be performed as well.

Analog designers who integrate standard cell digital into their predominantly analog designs face a number of challenges. Digital blocks may have pinouts that are suboptimal in an analog/mixed-signal floorplan. Switching of digital logic can cause noise and signal integrity problems, requiring analog designers to use adequate isolation techniques. Simultaneously switching digital gates can inject noise into the substrate, which can propagate and impact performance of sensitive analog circuits or even cause complete malfunction. Noise can also be transmitted around a chip via the power and signal routes and package.

The mixed-signal implementation flow must accommodate frequent engineering change orders (ECOs) intended to fix problems or to accommodate changes in specification. Analog and digital design teams may have to go back and forth to iteratively change pinouts, floorplans, physical layouts, and other design attributes to satisfy the constraints and performance demands of both analog and digital circuitry. Late-stage ECOs may require designers to re-do chip assembly and chip finishing.

Integrated Solution

Analog and digital design requirements, methodologies, and skill sets are fundamentally different, and design teams are used to their respective design environments. An integrated solution offers a planning-to-signoff methodology for a mixed-signal SoC that enables efficient interaction between the analog and digital teams and supports smooth integration of a large number of analog, digital, and mixed-signal IP blocks.

The integrated solution needs to support chip floorplanning, block-level design, chip assembly, physical verification, and signoff as shown in Figure 2.

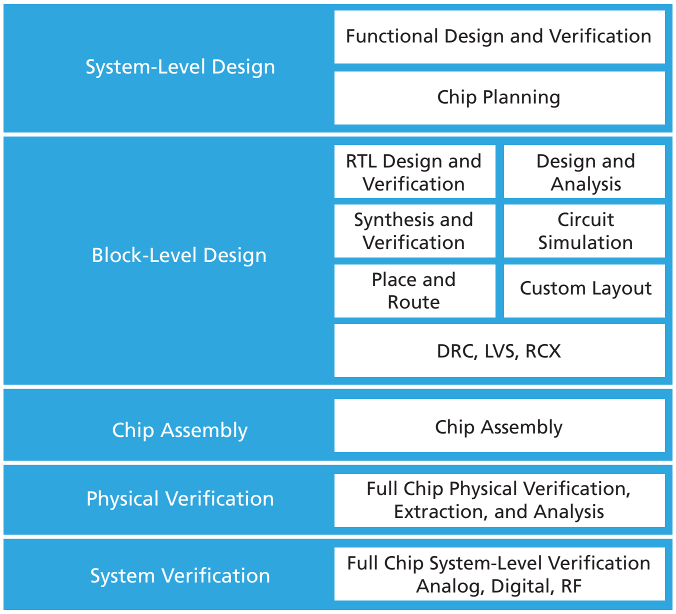
System-Level Design
Functional Design and Verification
Chip Planning
Block-Level Design
RTL Design and Verification
Design and Analysis
Synthesis and Verification
Circuit Simulation
Place and Route
Custom Layout
DRC, LVS, RCX
Chip Assembly
Chip Assembly
Physical Verification
Full Chip Physical Verification, Extraction, and Analysis
System Verification
Full Chip System-Level Verification Analog, Digital, RF

Figure 2 – Integrated mixed-signal design solution

Here are some of the key capabilities that an integrated mixed-signal implementation solution offers:

- The ability for digital designers to "push into" analog blocks and view their layouts
- The ability for analog designers to implement digital functionality within their hierarchical floorplan
- A common design database, such as OpenAccess, instead of a more limited layout representation using LEF/DEF or GDSII
- A chip exploration capability, such as Cadence InCyte Chip Estimator, that allows users to select analog, digital, and mixed/signal IP and obtain early estimates of size, power consumption, and cost
- Common design constraint support for digital and analog, such that constraints developed on each side can be understood by the other side
- A mixed-signal router that can understand analog constraints such as symmetry and differential pair routing
- Routing solutions that can optimize the routes for manufacturing
- Static timing analysis (STA) capability that can analyze timing paths through mixed-signal blocks
- A hierarchical design approach that can manage analog and digital design styles

In addition to meeting specific demands related to mixed-signal integration, a mixed-signal design flow must provide the performance and capacity for today's very large SoC designs, and it must meet all the design-for-manufacturing (DFM) requirements of advanced process nodes. It must also support low-power design techniques for both digital and analog IP.

Mixed-Signal Implementation Flows

As previously noted, there are two main types of mixed-signal implementation flows:

- Schematic-driven flows, in which a custom design environment such as the Virtuoso Layout Suite handles the floorplanning, chip assembly, and block integration. Digital blocks are custom-designed or imported from a digital design environment such as the Cadence Encounter® Digital Implementation System.
- Netlist-driven (Verilog/VHDL) flows, in which the floorplanning, chip assembly, and integration are handled in a digital design system. Hardened analog IP blocks are imported from an analog/custom design environment such as the Virtuoso Layout Suite.

Today's growing and ever-changing mixed-signal requirements demand that these established flows evolve into flows where floorplanning and chip assembly are shared between the analog and digital design teams, and where both analog and digital blocks are developed concurrently, providing additional flexibility to the implementation process.

Schematic-driven flow

This type of flow is driven by schematic entry and makes extensive use of constraints. Custom designers often draw schematics for full-chip, analog, and smaller digital circuits, and may use custom generators for complex datapath and array-based modules. For larger digital blocks, typically a design teams starts with Verilog HDL code to take advantage of digital synthesis, placement, and routing.

A more detailed look at the schematic-driven mixed-signal flow is shown in Figure 3. Here, the orange coloring identifies tasks typically done in an analog/custom environment, while blue coloring indicates tasks typically done by digital tools.

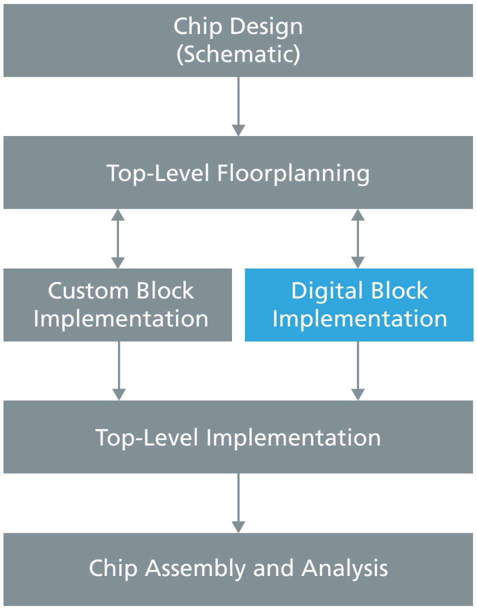
Chip Design
(Schematic)
Top-Level Floorplanning
Custom Block Implementation
Digital Block Implementation
Top-Level Implementation
Chip Assembly and Analysis

Figure 3 – The schematic-driven implementation flow

In the schematic-driven flow, top-level floorplanning is done in an analog environment such as the Virtuoso Floorplanner. This floorplanning takes place before blocks are implemented. Blocks must be placed to avoid signal integrity problems and to allow sufficient routing resources. Special care must be taken to properly place sensitive analog blocks that can be impacted by digital switching noise.

Analog and digital pinout definitions are drawn from the floorplan in a top-down or bottom-up strategy. However, analog floorplanning is interactive and suitable for chips with a modest number of IP blocks. If there are hundreds of blocks in the floorplan for which placement and pin optimization need to be performed, a more automated digital floorplanner may be preferred.

After a floorplan is developed, the implementation of individual blocks is handled in the appropriate design environment. Some blocks may also be pre-designed in-house, or they might be IP acquired from third parties. Implemented analog and digital blocks are then brought back into the custom design system and integrated. Top-level mixed-signal and chip assembly routing can be achieved using the Virtuoso Space-Based Router. Some analysis such as STA, however, may be done in the digital environment.

One requirement of this flow is the ability to effectively handle late-stage ECOs following changes in the schematic and Verilog netlist.

Virtuoso Digital Implementation

To facilitate digital block creation for schematic-driven flows, the Virtuoso Digital Implementation tool can be used. The tool is a capacity-limited version of the Encounter Digital Implementation System. It allows designers to efficiently implement digital blocks using an automated, timing-driven digital implementation flow, and bring the blocks back into the Virtuoso platform using OpenAccess (OA). LEF/DEF and GDS are supported as well, though they do not offer same level of capabilities as an OA-based flow. Virtuoso Digital Implementation includes RTL synthesis, block floorplanning, power planning and routing, placement, clock tree synthesis, signal routing, extraction, and static timing analysis. The solution can be driven by a prepared script, making digital block implementation easier for analog designers.

The best way to use Virtuoso Digital Implementation is with OpenAccess, an open industry database developed by Cadence that provides much faster and more complete data exchange than LEF/DEF. As shown in Figure 4, a top-level schematic is drawn in the Virtuoso environment, and a digital block is passed to the Encounter environment for implementation. The implemented digital block is returned to the Virtuoso environment, where chip finishing takes place.

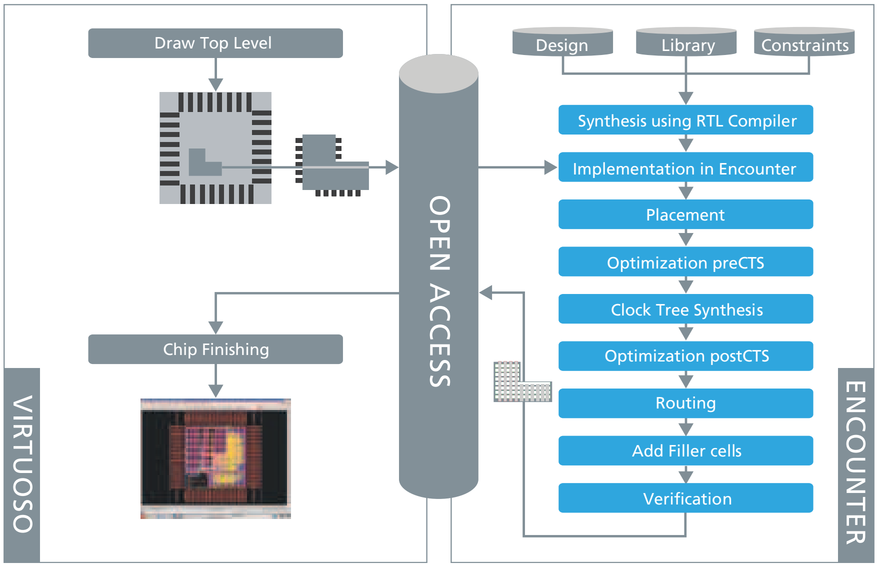

Figure 4 – By sharing the OpenAccess database, Virtuoso and Encounter technology enables users to work on unified design data for analog and digital blocks

Netlist-driven flow

The netlist-driven flow is used for predominantly digital designs and is outlined in Figure 5. It typically uses hardened analog IP designed by an analog design group or a third-party provider. The layout for the analog IP is completed, the pinouts are typically fixed, and guard rings are locked into place using the Virtuoso Layout Suite.

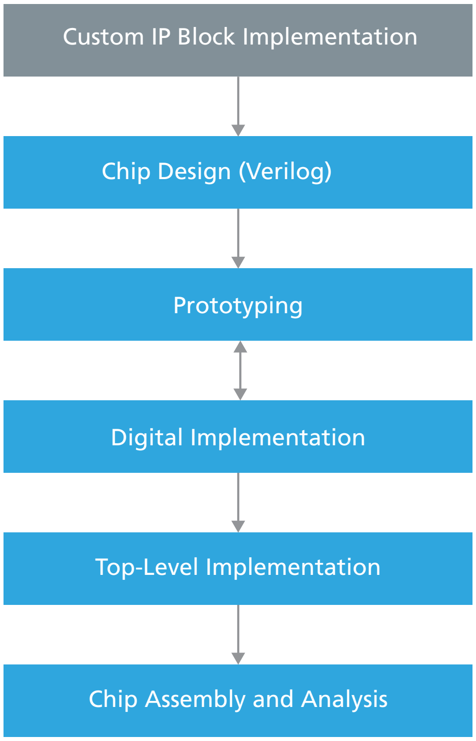

Figure 5 – Netlist-driven implementation flow

Floorplanning and prototyping precede detailed digital implementation. In the Encounter Digital Implementation System, chip prototyping provides a very rapid full-chip exploration that allows users to improve their floorplan for timing, power, and congestion. The prototype provides a high level of confidence that the design and floorplan will not cause integration challenges and timing closure problems later on.

After analog blocks are imported and digital blocks are either imported or created, top-level implementation takes place in the digital environment. Analog blocks may have specific area or aspect ratio requirements. Chip assembly and analysis mostly take place in the digital environment, although some of the verification, extraction, and analysis may take place on the analog side. SoC signoff requires static timing, signal integrity, and IR drop analyses, and the integrator must ensure that any additions or fixes don't negatively impact analog/digital interfaces.

Cadence Virtuoso and Encounter platforms support the OpenAccess database, greatly simplifying the transfer of analog blocks to and from the Virtuoso environment. Connectivity, physical design data, and routing constraints can all be saved by either tool environment in OpenAccess and then communicated among engineers in charge of block implementation, floorplanning, and chip integration.

In the Cadence flow, Encounter users have full visibility into layouts for analog IP blocks (Figure 6), including any Pcells they may contain. Analog layout objects, like guard rings, are frozen for editing until designers decide to override the restriction. Routing of sensitive analog nets (such as differential and matched pairs and shielded nets) can be done using Virtuoso Space-Based Router in the Virtuoso platform ran directly from the Encounter cockpit.

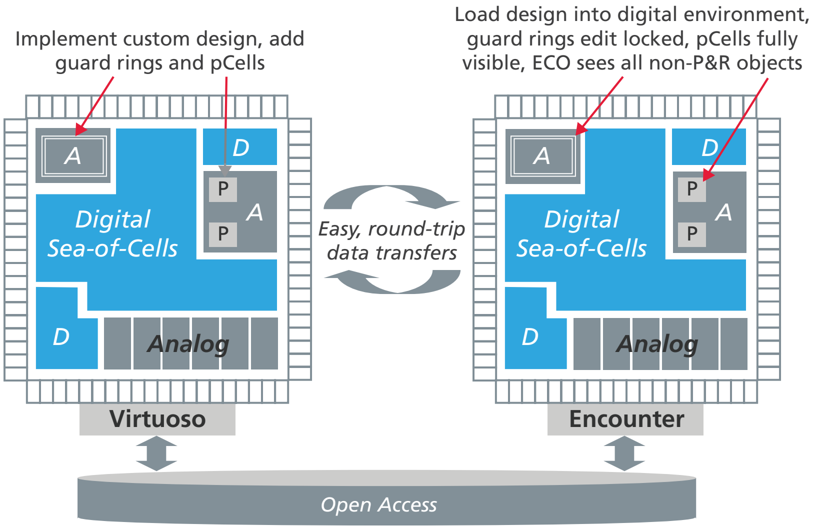
Implement custom design, add guard rings and pCells
Load design into digital environment, guard rings edit locked, pCells fully visible, ECO sees all non-P&R objects
A
D
P
A
P
Digital Sea-of-Cells
D
Analog
Virtuoso
Easy, round-trip data transfers
A
D
P
A
P
Digital Sea-of-Cells
D
Analog
Encounter
Open Access

Figure 6 – OpenAccess eases the integration of analog IP into SoC designs

Static timing analysis

The signoff of digital timing paths through mixed-signal blocks is problematic in traditional netlist-driven flows. Typically, digital implementation tools extract path parasitics only to the analog IP block's instance pin, and any loading inside the block is accounted for by a timing model (Figure 7, left side).

The Encounter Digital Implementation System has the capability to analyze paths through mixed-signal blocks all the way to the last digital instance (Figure 7, right side). This is enabled by the OpenAccess database carrying all necessary connectivity and layout information for the mixed-signal block to the Encounter environment to stitch it with the top and other blocks, and to perform RC extraction and timing analysis. This is the most accurate method for performing full-chip static timing analysis on a mixed-signal SoC.

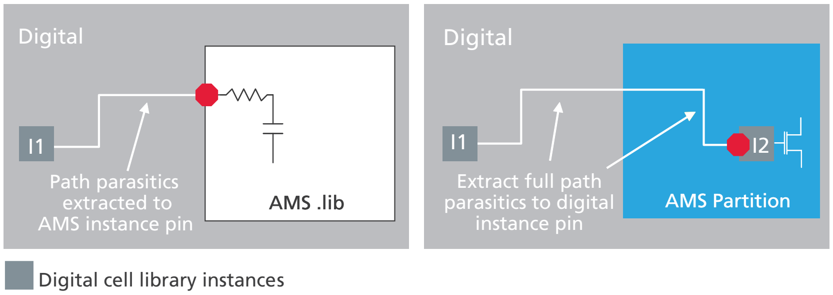
Digital
I1
Path parasitics extracted to AMS instance pin
AMS .lib
Digital
I1
Extract full path parasitics to digital instance pin
I2
AMS Partition
Digital cell library instances

Figure 7 – Extracting full-path parasitics allows timing closure on mixed-signal blocks without requiring generation of a .lib file

Concurrent mixed-signal implementation flow

Both schematic-driven and netlist-driven flows are block-based approaches that involve the import of a block designed in one domain (analog or digital) into the full chip that's done in the other domain. The concurrent mixed-signal flow represents a much more integrated approach. While block implementation remains the same, floorplanning, chip integration, and ECOs in the concurrent mixed-signal flow are done in a collaborative approach among analog and digital designers.

One big advantage of this new concurrent approach is that floorplanning can be truly mixed-signal—optimizing floorplan and pin location with full visibility into analog and digital objects that could be at the same level of hierarchy. After an optimal floorplan is derived, analog and digital blocks can be implemented concurrently. Any changes required to the floorplan through the process of block implementation are taken into account and re-optimized at the full chip level, thanks to the common, unified OA design database. This methodology results in better area optimization, less iteration, and fewer surprises during chip integration.

The concurrent mixed-signal implementation flow is shown on Figure 8. Notice how the top-level floorplan becomes a joint exercise between analog and digital design groups, usually facilitated by the lead floorplanning engineer or chip architect. These teams concurrently optimize the floorplan, changing pinouts and locations and routing nets as needed, until they can both sign off on the floorplan. Each team must be aware of the constraints on the other side. Once the floorplan is completed, custom and digital block implementations are done using their optimal tools. Chip assembly and analysis are collaborative tasks as well.

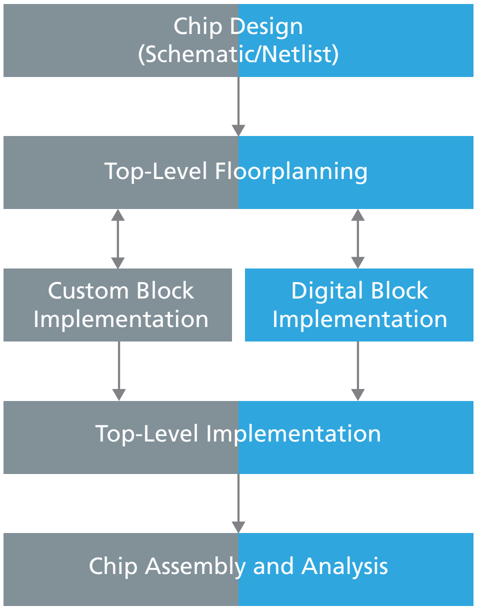
Chip Design (Schematic/Netlist)
Top-Level Floorplanning
Custom Block Implementation
Digital Block Implementation
Top-Level Implementation
Chip Assembly and Analysis

Figure 8 – Concurrent mixed-signal implementation flow

Cadence provides an integrated, concurrent mixed-signal flow built using industry-proven Virtuoso and Encounter platforms. The unified and shared OpenAccess database makes this type of flow possible. Instead of LEF/DEF transfers of black boxes, analog and digital designers can quickly view and operate on the analog and digital layout stored in a single OA design database.

Other important capabilities of the concurrent mixed-signal flow from Cadence include:

- The ability for the mixed-signal router to understand analog routing constraints
- The ability to view the complete design layout in digital and analog environments
- Improved pin optimization for analog and digital blocks
- Use of manual/interactive and automated hierarchical floorplanning
- No data translation between analog and digital domains
- The ability to mix digital and analog objects at the same level of hierarchy
- Easier manual or automated ECOs on a unified design database, possible at any stage of the design
- Static timing analysis covering parts through mixed-signal blocks
- Chip assembly routing with support for complex analog requirements as well as high-capacity digital needs

Conclusion

Today, applications at both mature and advanced process nodes require a higher level of integration of analog and digital functionality, increasing the challenges of mixed-signal implementation.

Traditionally, mixed-signal implementation was done in a silo approach among analog and digital designers with rigid handoffs between them. This is no longer effective for implementing modern mixed-signal designs—integrated flows enabling collaboration among analog and digital designers are required.

The mixed-signal implementation flows supported by Cadence help designers implement complex mixed-signal designs more productively. Schematic-driven and netlist-driven implementation flows remain broadly used and have been enhanced. The concurrent mixed-signal implementation flow, built by integrating Virtuoso custom and Encounter digital design platforms on a common OpenAccess database, enables the most advanced floorplanning, integration, ECO, and signoff capabilities. Moreover, this flow supports a collaborative approach among analog and digital designers to optimize designs for smaller area and higher performance.

cādence®

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