

THE ADVANCED CUSTOM DESIGN METHODOLOGY

TABLE OF CONTENTS

Introduction	1
The overwhelming challenge	2
The meet-in-the-middle approach	5
Platform requirements	7
Task-based methodology	8
Domain-specific block design	14
IP migration, re-use, and leverage	22
Summary	23

TABLE OF FIGURES

Figure 1	The ACD Methodology uses a meet-in-the-middle approach	1
Figure 2	Multidomain integration scope	3
Figure 3	Collateral chain supporting integration from a custom point of view	3
Figure 4	Pre-defined abstraction levels	5
Figure 5	Task-based flow	8
Figure 6	Block-level partitioning supporting top-level simulation needs	11
Figure 7	Custom digital design/verification loops	15
Figure 8	Block-level iterations	16
Figure 9	Device-level layout for custom digital design	16
Figure 10	Analog block creation flow	17
Figure 11	Analog layout options	18
Figure 12	RF IC flow	20
Figure 13	System/IC flow	21
Figure 14	AMS top-level flow	21
Figure 15	Physical chip integration flow	22
Figure 16	Re-use becomes critical in mixed-signal designs	23
Table 1	Sample portion of a simulation strategy	11

1 INTRODUCTION

This document describes a fast, silicon-accurate methodology for advanced custom/mixed-signal design. It directly addresses the primary challenge in creating these designs—predictability—by maximizing speed (pertaining to schedule) and silicon accuracy throughout the design process.

The Advanced Custom Design (ACD) Methodology is targeted to designers of full-custom designs, including those integrating digital standard cells within full-custom designs. The scope of the methodology covers the key design domains of analog, custom digital, and RF, and supports their integration with digital standard cell blocks (performed with a full-custom focus). The ACD Methodology is represented in *Figure 1*.

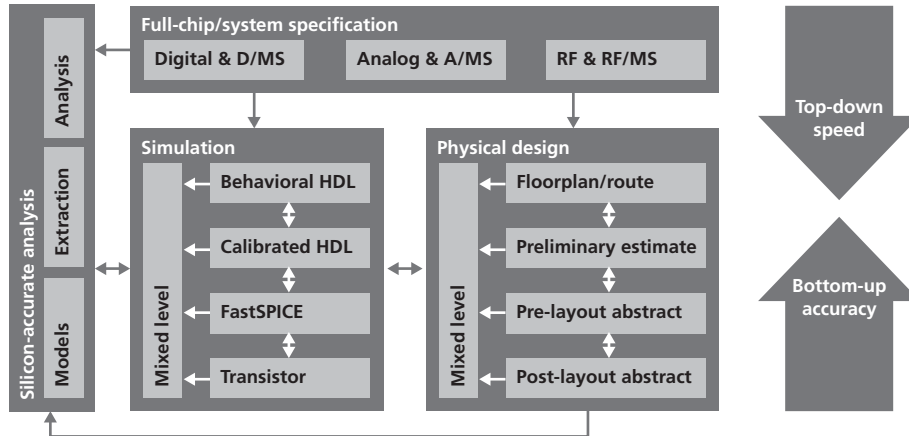


Figure 1: The ACD Methodology uses a meet-in-the-middle approach

Predictability is the driving force behind the ACD Methodology. Predictability is based on two primary concerns: 1) meeting schedule from the beginning of the design process, which necessitates a fast path to tapeout, and 2) meeting performance requirements to achieve first-pass success, which requires silicon accuracy.

Meeting schedule requires a fast design process that supports thorough and complete simulation and physical design. The design process consists of numerous tasks, and many of today's chips contain multiple blocks from multiple design domains. Thus, it is imperative to design in as many of these blocks as possible, perform as many tasks as possible in parallel, and leverage as much top-level IP as possible. This leads to the concept of design evolution, wherein all the design IP is leveraged as it matures through the design process.

This top-down design process, when applied to both simulation and physical design, is the approach that facilitates a fast path to tapeout. Multiple abstraction levels, from high-level design to detailed transistor-level design, are combined to support a mixed-level approach that targets detailed design to only the points needed for a given test. This also allows for leveraging top-level IP, using that information for block design, and subsequently re-verifying the top-level blocks.

At the other end of the design spectrum is the need for silicon accuracy to achieve the required performance. Silicon accuracy relies on base design data such as device models supporting accurate simulation and technology files supporting interconnect, physical verification, and analysis. Test chips, which often consist of critical structures known in the past to be highly sensitive, are also used to verify the feasibility of a process and the accuracy of its corresponding process design kit (PDK). Often, a design group will need to add additional components to the PDK to support a particular design style. Device models may need to be expanded upon to either combine or add corners, statistical modeling, or other approaches that the design team needs.

The silicon accuracy data is driven through the design process through detailed transistor-level analysis, including layout extraction. These steps comprise the lower level of the abstraction chain, which then support the calibration of these results to higher levels of abstraction. This comprises the bottom-up design portion of the ACD Methodology.

The top-down and bottom-up processes work in parallel to produce a “meet-in-the-middle” approach. It is this approach that balances the need for speed and silicon accuracy, ultimately producing a predictable schedule and first-pass success. The ACD Methodology can be applied to a complex integration or a particular domain area. Each domain applies the meet-in-the-middle approach, combining top-down speed with bottom-up silicon accuracy.

2 THE OVERWHELMING CHALLENGE

Taking on a complex SoC or multichip system is a daunting task, yet design teams are expected to achieve a predictable process. Multiple design groups contribute circuit specifications upfront and often use disparate methodologies to design blocks independently from each other—yet these design pieces must integrate seamlessly from both a physical design and a simulation/verification perspective.

Predictability suffers when any glitch in the integration process occurs. But most schedules are created without upfront attention to an encompassing methodology and under the assumption that integration will go smoothly. This almost guarantees problems, causing schedule slips that all but eliminate predictability of schedule or performance.

Problems at the top level can cause a total meltdown—yet this phase of the design process is usually crammed into the last three weeks before tapeout. Schedules slip in a seemingly endless iteration loop at the worst possible time (shortly before scheduled tapeout) on the worst possible database (full-chip level). Often the chip goes to tapeout without proper verification, which inevitably leads to re-spins and further delays in full production and profits.

Leveraging legacy IP from previous generation designs, or producing a derivative from a large SoC, further complicates the process. This can be due to additional market requirements, second sourcing for fab capacity, or migrating to a new process technology to enhance performance or reduce cost. In the custom design world, the term “IP re-use” often generates debate, since IP migrations/modifications require much more thorough design involvement than pure digital designs. Yet, the design itself is highly leverageable (i.e., simulation testbenches, design and layout topologies, and design process) and is a high-value starting point for the migration or modification effort. This issue highlights the integration problem: if a particular block is difficult to integrate the first time through, adding design tweaks for the next derivative can introduce new problems and delay integration. Thus, it is critical to have a design process for these blocks that supports integration and leverage for future designs.

In the end, predictability is the result of a methodology that supports a fast design process based on a silicon-accurate foundation. The top-down, fast design process aids schedule while the bottom-up, silicon-accurate information increases the likelihood of first-time success. Engineers need an encompassing methodology that enables them to focus on their areas of expertise, verify their designs in a chip-level context, and integrate their IP at the top level, which applies to the primary design domains of analog, custom digital, and RF. Chip integration is critical and complicated. It must be considered throughout the entire design process and, as such, is a standalone entity. The design process must be able to support both first-time designs and design derivatives seamlessly.

2.1 ATTACKING THE PROBLEM—KEY CONCEPTS

The ACD Methodology focuses on speed and silicon accuracy throughout the design process. The balance of these seemingly contradictory forces is the linchpin of an effective methodology designed to tackle today's and tomorrow's design challenges. Understanding how this balance is achieved requires more detailed knowledge of the following key concepts: design collateral, top-down design, bottom-up design, mixed-level design, and continuous design evolution.

2.1.1 Design Collateral

An overall design methodology consists of processes that target a particular design class and a particular user base. For any tool to be used effectively, it must be a natural part of the environment that a particular engineer is using. At integration (when analog, digital, and RF designs are brought together), engineers must pay special attention to who will be running the top-level simulations and performing top-level physical design, and where design collateral (netlists, databases) is coming from. It is helpful to take a similar approach to designing this “design system” as an SoC designer would take on a chip.

Figure 2 shows the scope of a complex system comprising several design domains. Each box on this diagram can be considered a “block” of the chip, where requirements exist within the block and block I/O requirements exist to support integration. The end simulation system must support full mixed-signal capability from both a custom point of view and a digital point of view. Each block produces design collateral (netlists, models, simulation setups) that must be 100% compatible for integration from either the custom or digital point of view.

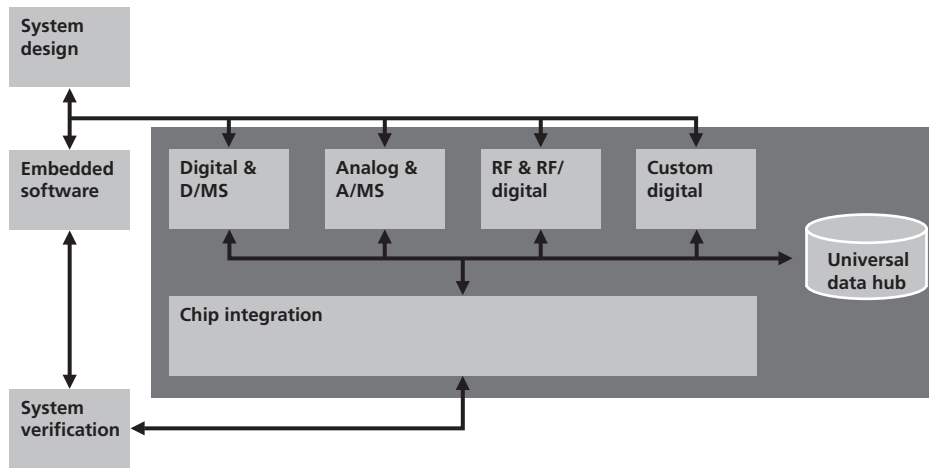


Figure 2: Multidomain integration scope

Therefore, the designer must not only consider the silicon accuracy and detailed process within a particular design area, but also what, and how, design collateral will be used to support both integration and a fast design process—particularly at the top level. Each design domain (analog, RF, digital, etc.) has the ability to produce this collateral as a natural fallout of the design process.

Designers will perform a full debug on design collateral native to their own environments, but the imported collateral referenced in most cases does not need full debug capability. Should errors be found in the imported collateral, it may require re-running the simulation in the environment native to where the problem has been identified. For example, if an analog designer is simulating an A/d (big A, little d) block and is convinced that a problem exists in the digital logic, the analog designer would very likely pull the digital designer in to debug the digital section. The digital designer would pull up the same simulation native to his environment, with the analog section referenced, and have full and natural debug capability on the digital section.

Looking at the problem top down, designers must ensure that the proper design collateral exists and is ready for integration. In the design world there is a “design chain,” through which suppliers feed primitives to IP groups who design IP blocks. These blocks are fed to design groups who then integrate and design other blocks to produce the actual chip. Any weak link in this chain causes the system to break down. Similarly, there is a “collateral chain” within the design process, targeted to design collateral that will be used at the next integration level. Figure 3 depicts an example of this collateral chain relating to behavioral models for design blocks.

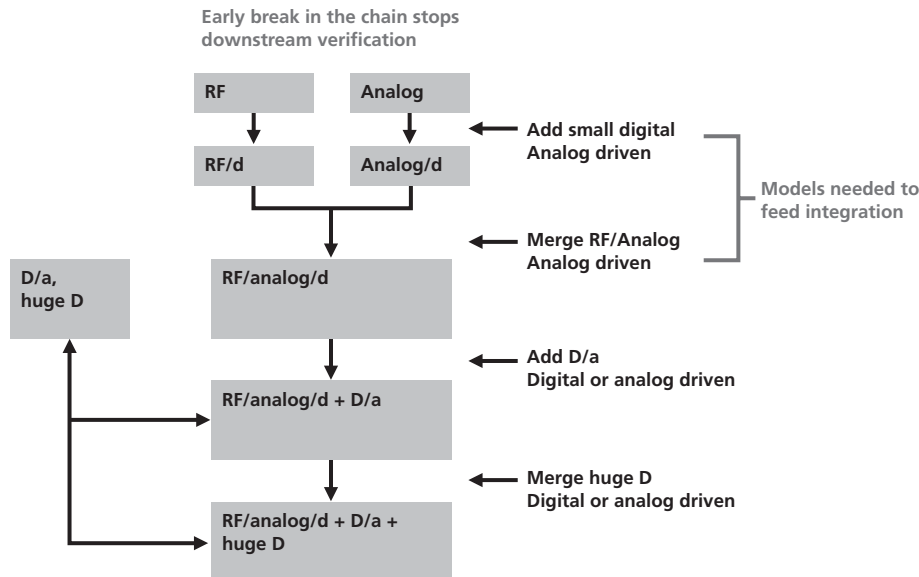


Figure 3: Collateral chain supporting integration from a custom point of view

From the custom point of view, circuit blocks are designed and then blocks are brought in for integration. The focus is either on the circuit block during the design task or on measurements driven from or dependent on the custom blocks during the verification task. To achieve final integration, all previous steps must be successful. Jumping from analog at the top to RF/Analog/d + D/a near the bottom would most likely be unsuccessful. A top-level simulation strategy, which defines what collateral is needed for each integration level, defines the design tasks for each circuit block/designer and the verification needs for that collateral. The incremental approach ensures that each piece of the puzzle will fit together in a phased manner. The same simulation can be run from either the digital or custom point of view where appropriate. A similar picture can be drawn from the digital point of view, where analog collateral is referenced as integration progresses. The effective integration of this design collateral supports a fast design process.

2.1.2 Top-Down Design

A fast design process utilizes the top-down approach extensively during both simulation and physical design. The full suite of simulation can be performed only with multiple abstraction levels starting from the system and behavioral level, with more detailed abstractions evolving over time. Similarly, a top-level physical design must start with early estimations (abstractions) of physical characteristics, which evolve and are updated throughout the design process.

Top-down design promotes speed by facilitating early development of top-level simulations and top-level physical design. For example, a first cut route is performed based on early estimations of block sizes and aspect ratios, followed by top-level extraction. This information is then used as block specifications early on, which saves time in block design.

Simulation and physical design tasks are set up early in the design process and are leveraged consistently throughout for verification, routing, and extraction. Any problems that arise get resolved early, rather than near tapeout, which ensures that versions of the design can be updated in realtime as design data becomes available. This ensures a fast design process since as much work is performed at the earliest stage of the design process as possible.

2.1.3 Bottom-Up Design

A silicon-accurate process utilizes the bottom-up approach extensively. During the design process, all full-custom circuits need to be verified at the transistor level in some form to ensure that silicon-accurate measurements meet specifications. Designers can use detailed parasitic information at their own discretion to make tradeoffs between design specifications and process capability in order to achieve the accuracy level necessary.

Bottom-up design aids silicon accuracy by facilitating all the detailed transistor-level analysis required. During physical design, the ability to bring in full layout data to perform extraction and analysis increases confidence that the design will meet performance criteria.

Often, detailed simulations can only be performed on small sections of the circuit, especially when iterations are common. Nor can it be expected to perform integration quickly with detailed transistor-level data. Thus, the bottom-up design process also supports abstraction capability, or calibration, of these results into faster models that can be run in larger simulations. The designer is responsible for determining how accurate a calibrated model needs to be at the next integration level to maintain accuracy in further simulations.

The bottom-up process is also used to migrate legacy IP into the ACD Methodology. A derivative design can be based largely on IP that already exists, but it is quite possible that this legacy IP does not fully support the ACD Methodology. Therefore, each block is brought into the bottom-up process and upgraded to "ACD compliance" to support the full methodology (see domain-specific block explanations in section 6).

2.1.4 Mixed-Level design

A mixed-level approach is defined as integrating multiple abstraction levels to perform a particular task. The mixed-level capability brings together silicon-accurate information with high-level abstractions to produce a fast design process.

Mixed-level capabilities apply to both simulation and physical design. Often the ability to bring a silicon-accurate representation of a design block together with fast representations of surrounding IP leads to a reliable and predictable method to implement the design under test (DUT) in the larger system. Other techniques discussed later in this document will address applying the mixed-level concept early in the design process to predefine strategies to overcome critical problems.

2.1.5 Continuous Design Evolution

The mixed-level approach supports a continuous design evolution capability, with which the design team can measure how the design is progressing from the top level down at any given time in the design process. While block design schedules do not align perfectly, a fast design process does not rely on a serial task-oriented approach. Instead, all tasks are continuously performed with the most up-to-date data (by definition, mixed-level representations).

Continuous design evolution relies on the simultaneous execution of top-down and bottom-up design styles. The design team relies on a detailed simulation and physical design plan that defines the mixed-level configurations for particular tasks. These are then used to continuously monitor and flag issues as soon as design data changes. This supports a fast design process by both feeding new top-level design requirements to the block level in realtime and identifying new top-level requirements due to block-level design realities.

3 THE MEET-IN-THE-MIDDLE APPROACH

The ACD Methodology relies on a “meet-in-the-middle” approach as the most pragmatic method to achieving predictability on complex designs. This is accomplished by combining the speed of top-down design with the silicon accuracy of bottom-up design. These two primary vectors essentially merge where the majority of the design activity cannot be described either as top down or bottom up, but rather as a combination of the two.

Multiple abstraction levels are used to represent the evolution of each piece of the design. During simulation, behavioral models—which grow more detailed as the design process moves forward—are used initially to bring in measurements and data from post-layout analysis. During physical design, as more design information becomes available, initial size estimates and block abstracts are updated to where the actual layout is used for the top level. Most of the time designers are actually working in the middle, with some blocks at the fast top-down stage and some blocks annotated with additional design data and silicon-accuracy information using the bottom-up process.

This simultaneous use of top-down and bottom-up design, with mixed-level capability at the core, is what constitutes meet-in-the-middle design. Throughout the majority of the design process, the design team is working at this level. The meet-in-the-middle approach supports a continuous evolution of design data to blend the need for both fast design process and silicon accuracy.

Dealing effectively with legacy IP is often the driving force behind a meet-in-the-middle approach. Rarely does a design team start from a clean slate (with which a pure top-down methodology can be employed from the beginning). In the majority of cases, legacy IP blocks must be upgraded to support the ACD Methodology, and this is accomplished through the bottom-up approach. Most blocks may only have the transistor level and layout abstraction levels supported—as a result, the abstraction levels are derived bottom up and then fed to the top-down process.

Pre-defined abstraction levels (see *Figure 4*) serve as the foundation of the meet-in-the-middle approach. Both simulation and physical design have pre-defined abstraction levels that are updated throughout the design process and support mixed-level capability.

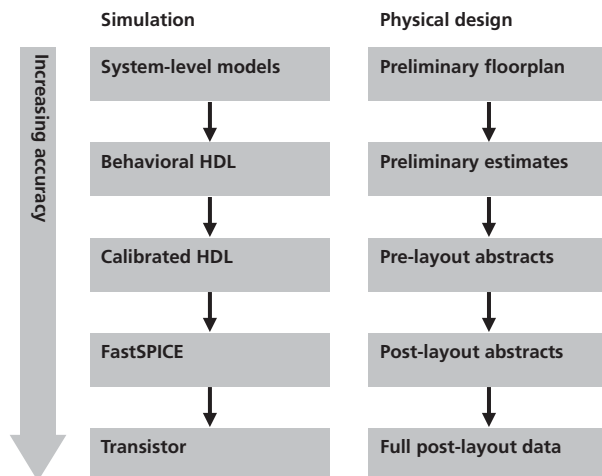


Figure 4: Pre-defined abstraction levels

System models (simulation)

System models are generated from the system-level environment and are the highest level of abstraction represented. They are typically C/C++, SystemC®, proprietary system simulator libraries, etc., which include the testbenches used for system simulations.

Behavioral HDL (simulation)

Behavioral HDL most often refers to Verilog®, Verilog-AMS, Verilog-A, VHDL, VHDL-AMS, or VHDL-A descriptions. At this level, only the circuit functionality is described, and the models are targeted for fast runtime.

Calibrated HDL (simulation)

Calibrated HDL models are calibrated off of transistor-level simulations, making the initial behavioral HDL models more accurately represent circuit behavior. Measurement and performance data are added to the models to balance silicon accuracy vs. fast runtime.

FastSPICE (simulation)

FastSPICE “abstraction” uses the same transistor-level description as for SPICE simulations. Running a FastSPICE simulator allows the designer to choose between silicon accuracy and fast runtime; the FastSPICE option can therefore be considered a separate abstraction level. It can use pre-layout or post-layout data.

Transistor (simulation)

Transistor-level abstraction refers to SPICE-level simulation at the most accurate level, and it can use pre-layout or post-layout data.

Preliminary floorplan (physical design)

The preliminary floorplan is the highest level abstraction for the physical design process. At this stage, relative placement, initial pin optimization, and other floorplanning investigations are supported.

Preliminary size estimates (physical design)

Preliminary block size estimates are based on previous experience, information from derivatives, initial process feasibility studies, or any information on which the designer can base a block size—short of a transistor-level design. Preliminary size estimates will vary through the process, but they are pin accurate (consistent with the simulation models) and support first cut routing and early physical chip integration tasks.

Pre-layout abstracts (physical design)

When a transistor-level description is ready, the preliminary size estimate can be updated to more accurately reflect the layout, prior to completion. The active area and estimated area for routing can be used to further define the abstracts and support the routing task.

Post-layout abstracts (physical design)

When layout is complete, a final abstract matching the physical representation can be provided to the routing process.

Full post-layout data (physical design)

The full physical representation is needed to support final physical verification as well as chip finishing tasks and final tapeout.

The ACD Methodology then uses these abstraction levels as its design components across the design process. How effectively these abstraction levels are put together will determine the level of predictability for the design. The design team must manage a plan upfront to determine where to bias the process for silicon accuracy or speed, define a mixed-level definition to accommodate it, and execute the plan through the meet-in-the-middle approach.

4 PLATFORM REQUIREMENTS

The ACD Methodology is implemented by the tools and processes provided in a unified design platform. The platform provides the infrastructure and support for verifying complex systems in an efficient manner. Using a common platform across projects and throughout the design chain promotes efficiency and speeds overall design development. There are several important requirements of any platform that supports the ACD Methodology.

4.1 SPECIFICATION-DRIVEN ENVIRONMENT

Support of the top-down design approach, and especially continuous evolution, is greatly aided when the design team focuses on measurements and specifications. These turn into testbenches and simulation setups, all of which need to be managed from a specification-driven perspective. This allows for easy setup of continuous regressions, supporting the characterization of silicon accuracy and the fast, optimized execution of the simulation strategy.

4.2 UNIVERSAL DATA HUB

Seamless integration of design data across multiple design domains is central to mixed-domain design. As such, a universal data hub is needed for data to move across design tasks and between completely different processes. For example, digital standard cell design is not covered within the ACD Methodology; however, design data coming from the digital standard cell design process must enter the platform easily and completely to achieve a fast design process and maintain silicon accuracy. Execution time and capacity are also key features of the data hub because large databases often need to be manipulated at various times throughout the design process.

The universal data hub must extend through the front-end and back-end tool infrastructure because tasks like RC extraction and analysis of IR drop, electromigration (EM), and substrate noise requires a melding of front-end and back-end data.

4.3 MULTI-MODE SIMULATION

Simulation capability must extend through multiple design domains (analog, RF, custom digital, digital standard cell) and support the mixed-level capability and meet-in-the-middle approaches. Simulation must not only support silicon accuracy at the lowest abstraction levels, but also enable high-level descriptions for fast execution.

Mixed-language support (for example, where digital logic could be described in VHDL and analog representations in Verilog-AMS) needs to be available because design teams cannot always control the representations in which imported IP can be described.

4.4 ACCELERATED LAYOUT

Since designers must balance silicon accuracy with a fast design process, accelerated layout capability is a platform necessity. This should allow the designer to perform device-level layout quickly and feed that layout to the top level. Accelerated layout can be used in conjunction with manual approaches, where an initial cut at the block uses as much automation as possible, and then manual tweaking occurs if necessary. Polygon pushing across all custom blocks does not support a fast design process.

4.5 ADVANCED DEVICE MODELING

Silicon accuracy is based on the reliability of simulation results within a predictable design margin and the ability of silicon results to validate the approach. As such, device modeling needs to be part of the platform, with the underlying simulators tied closely to the device modeling technology to ensure the most accurate simulation results. Device modeling capability is also needed because design teams make custom additions to PDKs, and it ensures model consistency across those additions.

4.6 PROCESS DESIGN KITS

Process design kits (PDKs) are at the core of how the platform achieves silicon accuracy and, as such, need to be complete and verified across the entire toolset supporting the ACD Methodology. The more a single PDK can support the complete platform, the more a fast design process can be executed as all tasks can be reliably performed to silicon-accurate levels. In contrast, when multiple PDKs are used and a design team relies on stitching capability, the likelihood of error increases, compromising both speed and silicon accuracy.

4.7 SILICON ANALYSIS

The ACD Methodology performs multiple analyses (IR, EM, substrate noise, RC extraction) throughout the design process. These analyses rely on silicon-accurate data from the PDKs and simulation results to highlight potential problems often not detected through traditional simulation. Predictability can only be achieved through a parasitic-aware methodology supporting continuous evolution of the full design through a fast, silicon-accurate process.

5 TASK-BASED METHODOLOGY

The implementation of the ACD Methodology can be described through a task-based flow, as shown in *Figure 5*.

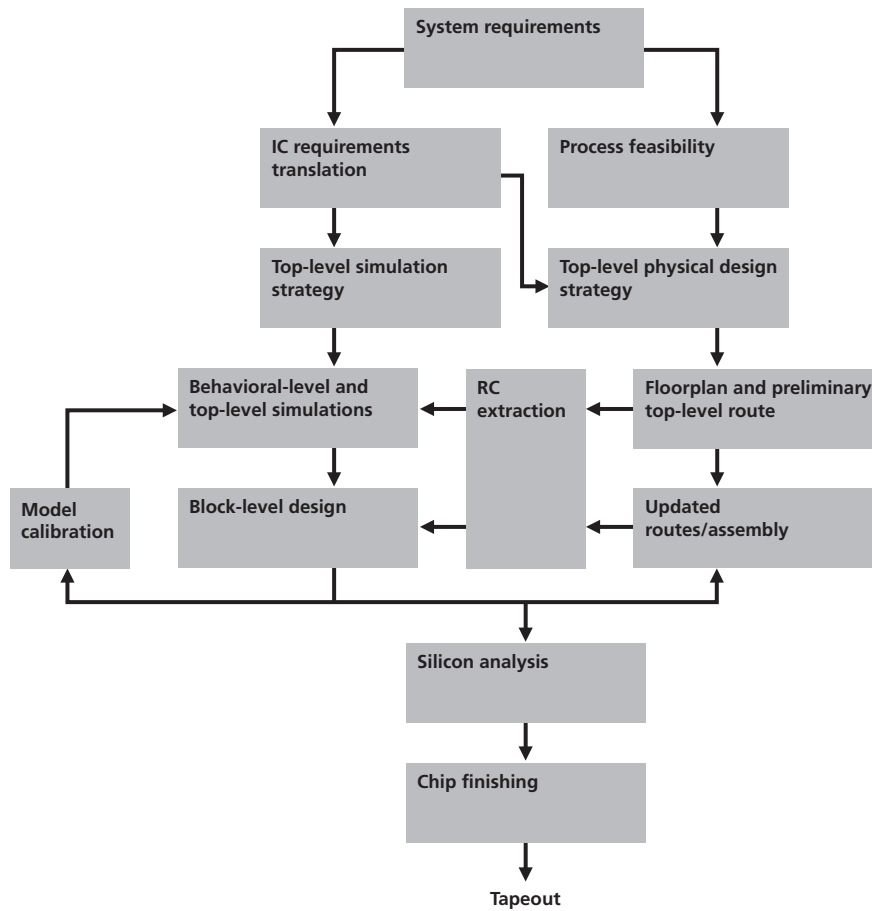


Figure 5: Task-based flow

As described in the previous sections, a fast, silicon-accurate design process is achieved by attacking the final design early in the design process. Since the top-level tasks pose the most risk, and because saving them until the end of the design process invariably produces delays and iterations, it is imperative to move top-level tasks to the front of the design process. As the process moves along, blocks get further defined and fed into a top-level evolution process—where top-level tasks, simulation, and physical design are continuously verified with updated design collateral as these pieces mature. Supporting and maintaining this evolution is what ensures predictability.

The primary advantage of enforcing a methodology that allows for the top level to continuously evolve is that difficult tasks—silicon analysis, RC extraction, routing, and physical verification—can be performed early on. While this is not the final version of these tasks, interim data is used from these tasks initially to drive design tasks through the hierarchy and support a fast design process. Also, the mere fact that these tasks are performed early on ensures that they can be repeated downstream as the design matures. Silicon accuracy information is provided through abstraction levels as the design matures, assuring the design team that the chip will come together at the end. Knowing how long it takes to route the top level through verification per design iteration allows the design team to more accurately predict the time each stage of the design process will take, because any tool or data issues are resolved early on.

5.1 SYSTEM REQUIREMENTS

By definition the system design process itself is beyond the scope of the ACD Methodology; however, the methodology relies on the system design task being performed. For more information on the system design and verification methodology, please refer to the Cadence® Unified Verification Methodology white paper. In the ACD Methodology, the design collateral from the system design process is used as the first, and highest, abstraction level. System-level descriptions are also highly useful as top-level chip tests. Models of the surrounding system can be combined with a high-level model of the chip, producing a “chip under test” scenario within the system. System requirements serve as

the first specifications to drive the chip-level requirements and ultimately turn into repeatable testbenches and regression simulations.

Leveraging system-level IP greatly enhances a fast design process, as the most reliable testbenches are those used in defining the system and chip specification.

Deliverables of this task are:

- System-level models
- System-level testbenches
- IC performance specifications

5.2 PROCESS FEASIBILITY — ADDRESSING SILICON ACCURACY INFRASTRUCTURE

With IC requirements generated from system specifications, process technology must be selected and evaluations of silicon accuracy capabilities and various integration strategies must be performed to verify the feasibility of the proposed integration approach. Issues such as performance, noise characteristics, cost, circuit type, and risk are all considered.

When a process technology is selected, the designer must consider the entire design process to be used. As a result, the infrastructure associated with it must support each required task in the design process. Much of the circuit will require silicon-accurate data. As such, the designer must consider the accuracy of device models, extraction decks, and so on, because the task is not simply to run the tools and get a number, but to verify data that it is ultimately correlated to silicon.

The PDK must be validated for its completeness and potential accuracy. The design team needs to ensure that the PDK is complete for all tasks through the design process, ideally to a baseline PDK specification for the platform. Should the foundry not support all tasks within the methodology, the design team must evaluate how to enhance the PDK to bring it into compliance and ensure full silicon accuracy. The time to address this issue is before design begins, because risk areas must be identified upfront and either flagged or addressed before final chip design steps are performed. Often, the PDK is taken for granted, leading to problems at each step. A well-characterized, complete, and high-confidence PDK is absolutely essential in ensuring a predictable design process based on silicon accuracy.

For higher speed designs, it is quite possible that additional devices may be needed that are not provided initially by the foundry. In this case, these devices will need to be added to the PDK, undergo device model extraction, and then augmented to the foundry models. Layout views, schematic views, and all such supporting collateral also need to be added consistent with other devices contained in the baseline PDK specification.

Highly sensitive IP should also be considered at this point and may require a test chip to verify. Should a test chip be developed, correlating silicon results to simulation results will verify a design margin that is incorporated throughout the design process. In addition, using any pre-designed IP that is silicon verified before the design process adds a high degree of confidence to the design. Depending on the sensitivity of the design, the design group may have to be involved with this process, with device modeling engineers ensuring the integrity of model data and design engineers ensuring critical circuit structures and devices are verified. With the importance of parasitic data, the accuracy and correlation of the extraction process is paramount to successfully verifying a design and achieving silicon accuracy.

The end deliverable of this feasibility process must be a verified and correlated PDK that supports each step of the design process. Design engineers are given guidelines for the design margins necessary to achieve success and thus avoid excessive overdesign—this is central to silicon accuracy. Not only does silicon accuracy prevent re-spins by ensuring that simulation results are accurately predicting real data, but it also prevents costly overdesign (resulting in tighter area), because the design team can confidently rely on a design margin based on silicon accuracy data.

Deliverables of this task are:

- Process selected and validated to IC specifications
- PDK verified for completeness
- PDK correlation validated
- Design margin determined

5.3 IC REQUIREMENTS TRANSLATION

The system design process produces specifications that the IC must meet. As mentioned above, the IC design process leverages system-level design collateral through the use of system-level models, testbenches, and measurements. The testbenches may be further enhanced to match IC specifications, where the specification-driven environment can be set up. The specification-driven environment then drives all the chip-level and, subsequently, block-level tests in a manner consistent with the original requirements given to the design team.

Deliverables of this task are:

- Electronic specification of top-level IC specifications
- Setup for specification-driven environment for above specifications
- IC-level testbenches derived from system-level environment

5.4 SIMULATION STRATEGY

With a process selected and its feasibility and silicon accuracy ensured, the simulation strategy by which the design will be built can be defined. At this point, the design team has made primary decisions as to the integration strategy of the design (is this single chip or multichip?), and has identified constraints upfront to insert throughout the design process.

Successful execution of a complex design is contingent on the thoroughness of the planning upfront. No design can come together smoothly by accident. With a strong initial plan that specifies top-level requirements, block-level requirements, and mixed-level strategies to use, a meet-in-the-middle approach can drive each block design to ensure full coverage of important design specifications and allow for blocks to have different schedule constraints. By using the most up-to-date information available at any given time (the continuous evolution approach), blocks that are finished earlier can be verified in the top-level context and be ready to go. This allows more time and resources (if necessary) to focus on the more complex blocks, which can also be using the most up-to-date information.

At this stage, the high-risk points for targeted verification are flagged and examined. These could be areas such as analog/digital interfaces, timing constraints, or signal/data paths. Extremely important at this stage is to look at a simulation and physical design approach that can support verifying these risk points. In the mixed-level approach, the abstraction level at which these risk points are described needs to be examined. For example, a key analog/digital interface may need both the digital interface section and the analog interface section described at the transistor level, with detailed parasitic information in between to ensure bit errors do not occur. If this is the case, the designer must examine how the design will be partitioned to allow this simulation to occur in an efficient and repeatable manner. Often, this interface can only be tested at chip level over a variety of simulation setups. Predictability will be based on the assumption that all critical items are part of a simulation and verification strategy, and that they are repeatable and reliably executed throughout the design process.

With critical circuit issues identified, the next step is to tackle design partitioning as part of the simulation and physical design plans. It is important to consider design partitioning from not only a functional perspective, but also from a verification perspective, because the design tools must verify the identified critical circuit issues effectively. The designer must consider the ability of the tools to handle certain types of analysis and design the circuit hierarchy to isolate each issue and efficiently tackle the problems associated with it.

Design partitioning is nearly always viewed from a functional perspective. It is natural to partition in this way because it leads to block specifications and layout partitions, which in turn leads to a natural top-level simulation strategy. It is important to keep this functional partition intact. However, as in the case of the above example of the analog/digital interface, designers must also consider how — at the top level — the mixed-level capability can be employed to verify this interface. One approach is to ensure that the block partitioning on the analog side has an interface piece that can be swapped at the transistor section, and that the digital side also has its interface piece able to swap at the transistor section. Parasitics can be added inside these transistor sections and interconnect parasitics can be backannotated in between these blocks. The rest of the chip level can then be described at the HDL level of choice for increased simulation speed. *Figure 6* represents this concept.

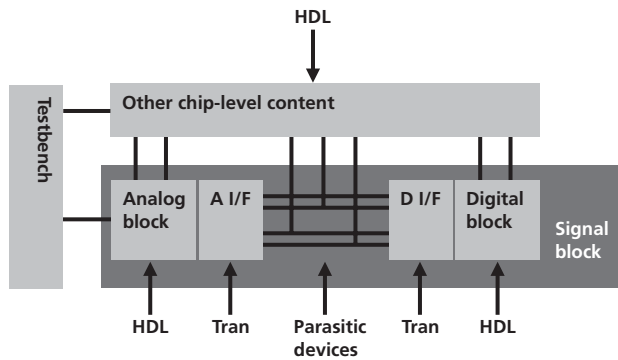


Figure 6: Block-level partitioning supports top-level simulation needs

If the design partitioning does not take this situation into account, then the next option is to bring all the analog and digital blocks into the transistor level (assuming this interface is critical and needs to be simulated at the transistor level). While this would achieve the objectives, it is quite possible this simulation would be slow, regardless of the simulator used. This would also require that the transistor level be complete. In contrast, the partitioning approach allows for the analog and digital sections to be completed at their own pace. If the interface sections are completed first, the interface itself can be tested before the analog and digital core pieces are complete, aiding a fast design process. The ability to simulate the interface of concern at the transistor level satisfies the silicon accuracy requirement. As the design evolves, it may be desirable to bring more pieces into the transistor level, or to simulate the analog block with the analog interface at the transistor level, etc. This adds to the predictability of the design process by enabling evolution and resolving critical design issues early on.

Thus, the simulation strategy must be comprehensive to account for all tests that must be performed and ensure that the design database is partitioned conducive to that strategy. The simulation strategy should also take into account the completion estimates of each block and specify the mixed level for that simulation. Table 1 lists some sample sections of a simulation strategy.

Top-level test	Testbench	ADC	DAC	DSP	Codec
Codec verif	Functional A	Verilog-AMS	Verilog-AMS	Verilog	SPICE transistor
BER function	System BER	Verilog-AMS	Verilog-AMS	Verilog	Verilog-AMS
DSP verif	Functional A	Accurate Verilog-AMS	Verilog-AMS	FastSPICE transistor	Verilog-AMS

Table 1: Sample portion of a simulation strategy

For large SoCs, separate tables may be necessary for the major blocks. In many cases, the first level of hierarchy for each block is much like a large chip and may have all the issues associated with a chip. In these cases, separate add-on tables like Table 1 may exist for each block at the top level and, subsequently, throughout the hierarchy where applicable.

As the design evolves, analog HDL descriptions can become more accurate as transistor-level simulation results are backannotated into the models. There is some sacrifice in simulation speed for this, however. As such, the simulation strategy is amended where the accurate models are needed. In block cases, it is likely that accurate HDL will be used across the board in conjunction with FastSPICE capability and SPICE-level capability for the most sensitive circuits.

For complex blocks that require some silicon accuracy at the top level, it is possible that a particular mixed-level configuration will be specified by the block designer for use when simulating at the top level. At the top level, this block-specific configuration would be used to exercise the simulation strategy. One view may be a non-hierarchical behavioral view for the block; another may contain the internal sub-blocks at accurate HDL or transistor level. This hierarchy and configuration needs to be managed to match the simulation strategy.

Deliverables of this task are:

- Documented simulation strategy
- Design partitioning consistent with executing to the simulation strategy

5.5 BEHAVIORAL-LEVEL/TOP-LEVEL SIMULATIONS

The fast, top-down design process necessitates a top-level HDL description of the design. This description is consistent with the partitioning specified through the simulation strategy and follows the declared hierarchy.

The simulations performed are consistent with the specification-driven environment, where individual tests are documented in the simulation strategy. These simulations are then used as test beds for blocks under test. Block-level testbenches are derived from the chip-level simulations capturing block-level stimulus.

Deliverables of this task are:

- HDL descriptions consistent with documented partitioning
- Simulation results consistent with system-level results
- Block-level testbenches derived from and/or defined based on top-level simulation results

5.6 BLOCK-LEVEL DESIGN

Block-level design is based initially on the top-level simulations that verify the block specifications. Block-level design then encompasses the detailed, silicon-accurate, transistor-level design. This also includes incorporating parasitic data and performing silicon analysis.

Regardless of the individual block's design domain (analog, RF, custom digital), the deliverables from the design process support integration by supplying the necessary collateral and abstractions. Some blocks may follow the chip-level process from the start as they are complex and multidomain in their own right.

Detailed, silicon-accurate, transistor-level design is performed at this stage to ensure full block-level performance and functionality.

Deliverables common across domains of this task are:

- HDL descriptions consistent with block-level hierarchy
- Block-level simulation plan
- Block-level physical database
- Block-level specification-driven environment setup
- Physical verification setups (completed physical verification outputs)
- Block-level post-layout simulation results

5.7 MODEL CALIBRATION

The silicon accuracy process, enabled through the bottom-up flow, requires higher level abstractions to maintain as much of the silicon-accurate information of the individual blocks as possible. This requirement is met by calibrating functionally-correct behavioral models with the silicon-accurate design data derived through post-layout transistor-level simulation.

This supports the mixed-level capability, allowing the design team to achieve a fast design process by targeting those blocks at the HDL level to be updated with silicon-accurate information. In some cases, calibrated models can run with transistor-level models to more effectively perform various block tests.

The repeatable specification-driven environment is leveraged to generate the simulation results over the necessary corners, temperature, voltages, and so on. These calibrated models then become the next abstraction level above transistor-level analysis options.

Deliverable of this task is:

- Pin-compatible HDL models performing within a specified margin of transistor-level results

5.8 PHYSICAL DESIGN STRATEGY

The top-level physical design strategy is specified in parallel with the simulation strategy, although to some extent the simulation strategy depends on the specification of the hierarchy and partitioning. The purpose of the physical design strategy is to look at routing constraints, floorplanning constraints, and initial placement based on block characteristics. These constraints determine what level top-level routing is performed (will some blocks be reduced to the second or third level of hierarchy?) and flag any areas of concern.

At the most upfront stages of a design, transistor-level descriptions and layouts likely do not exist. In some cases, such as derivatives or process migrations, previous layouts can exist to estimate size accurately. A similar block in a different process can also be used to derive a first cut size estimation. If no guidelines exist, then previous experience is used.

It is important to try to keep the layout and simulation hierarchies matched as much as possible. Most layout tools allow for different hierarchies from the schematic, and at times it may seem like chip area can be saved through flattening or re-partitioning. While this is possible in many cases, issues arise when post-layout simulations occur. Managing silicon-accurate parasitic data is of paramount concern. It is unlikely that analysis can be performed flat with full parasitics, many of which are distributed, at the top level. This means that the only way to manage these parasitics when related to the chip level is through hierarchically simulating, abstracting, and folding in parasitic data and inter-block routing information in a controlled and manageable process. As a result, great care must be taken when the physical design process specifies changes to this and the implications on post-layout simulation must be considered.

Deliverables of this task are:

- Documented physical design hierarchy, flagging inconsistencies with the simulation partitioning
- Top-level routing constraints (matching lines, over-the-cell routing constraints)

5.9 FLOORPLAN AND PRELIMINARY TOP-LEVEL ROUTE

The floorplan and preliminary top-level route are critical in supporting a fast design process. When an initial route has been completed, a repeatable process exists to support continuous design evolution. The setups and constraints are re-used and modified as the design evolves. These setups identify issues at the top level early—when both design and tool issues can be resolved before tapeout. Predictability can only be achieved if these steps are completed early and repeated throughout the design process.

Top-level chip assembly often poses the highest risk in scheduling predictability—especially when starting at the end of the design process. Therefore, it is critical to flush out as many issues as early as possible to ensure a clean and repeatable process moving forward. Silicon accuracy is also enhanced when parasitic effects are considered early, because block-level design can factor in these effects from the beginning. Although the blocks may only be based on size estimates that will certainly change, a number of issues can be dealt with at this stage that affect the design in short order.

Top-level parasitics are a major concern, where critical nets can be identified and analyzed early, and loading constraints can be passed on to the block designer for consideration before transistor-level design is complete. Issues such as critical analog/digital interfaces (see section 5.4) can be identified and the physical design database can be established upfront to support parasitic extraction of interface lines and block parasitics where needed, which can feed the simulation setups for that particular test. Often these setups are complex and contain a number of tool issues; the correct time to identify and solve these issues is early on. Doing so can have a significant effect on partitioning for analysis, and in many cases the partitioning will need to be changed to support the tests identified at this level.

Deliverables of this task are:

- Initial floorplan, producing pin location specifications for block design
- Initial top-level route (at abstract level) for parasitic information to block design
- Critical nets identified for further RC extraction and analysis

5.10 UPDATED ROUTES/ASSEMBLY

As the design evolves, the initial setups are used to route updated physical abstracts that represent more accurate size estimations, ultimately through accurate block abstracts generated from the completed block layout process. The mixed-level capability and meet-in-the-middle approach are used in a similar manner as the simulation plan, aiding the fast design process. Block designs are not complete at the same time, yet as each block matures to pass on an updated or complete physical representation, a top-level route is performed to ensure consistency and up-to-date design evolution, catching problems as early as possible.

Physical verification is performed as the blocks mature and each iteration is routed. At the end, a final physical verification (with all design data read in) is run to resolve any unforeseen errors.

Deliverables of this task are:

- Incremental top-level routes for input to RC extraction
- Final physical database
- Physical verification setups (completed physical verification outputs)

5.11 RC EXTRACTION

Wherever possible, post-layout analysis on the first cut database should be setup as well, even if the results are not yet totally meaningful. Predictability is achieved when this analysis is performed downstream. Dealing with the top level for any function is cumbersome at best, but smoothing the process upfront facilitates downstream analysis. This would likely include top-level net extraction and analysis of substrate noise, IR drop, and electromigration. Thus, the physical design strategy must address if, how, and where this analysis should be conducted.

Silicon-accurate RC information is fed to block-level design early on as block specifications. As the design matures, simulations are performed with extraction data from updated routes to continuously re-verify the design. The simulation strategy, which determines how the meet-in-the-middle approach will be applied, specifies how and with what abstraction levels the simulations will use the extraction data.

Deliverables of this task are:

- RC data for critical nets, fed as loading requirements for block design
- RC data for nets to support top-level simulation

5.12 SILICON ANALYSIS

At the top level, important silicon-accurate analysis is performed that functional-based simulation does not catch. This includes IR drop, EM, and substrate noise analysis. Some silicon analysis can be performed at the block level, and some analysis can be performed during the updated routing tasks.

IR drop analysis is necessary on all designs, both at the block level and chip level, because it is possible that power lines are not sized appropriately, causing voltage drops and resulting in circuitry not performing to specification.

EM analysis should be performed on all designs, even when the design team is sure the problem doesn't exist—because an EM problem can manifest itself over the lifetime of the design in the field.

Substrate noise is most acute when RF circuitry (high speed and sensitive) combines with digital logic. This is also true when high-speed or sensitive analog circuitry combines with digital logic—digital logic is inherently noisy.

Deliverables of this task are:

- IR drop verification and repeatable setups
- EM verification and repeatable setups
- Substrate noise analysis and repeatable setups

5.13 CHIP FINISHING

Chip finishing includes tapeout preparation tasks such as adding PG text, layer editing, adding copywrites/logos, and metalfill. It may also be necessary to make final edits at this point based on last-minute design needs. While it may be necessary to “break the loop” to meet schedule, after tapeout the design team must backannotate the change into a repeatable process rather than rely on undocumented last-minute editing for each design derivative.

To execute chip finishing, the complete design database must be read in and a final “full guts” DRC/LVS must be performed. The chip finishing tasks are then performed on this data, and physical verification is repeated for the final check. As a result, the chip finishing tasks are often performed on a large database.

Deliverable of this task is:

- Final physical design database for tapeout

6 DOMAIN-SPECIFIC BLOCK DESIGN

It is this point in the design process that comprises the traditional full-custom design approach, which consists of transistor-level focus and detailed design. Section 5.6 described the general block design process as common across design domains. This section will describe the block design process specific to each design domain.

With information at this point ensuring top-level integration, block designers can design to these specifications and meet their individual block-level specifications. These blocks fall into the various domain categories of custom

digital, analog, and RF. Similarities exist between the domains; however, each domain has unique requirements. The design process for each domain produces silicon-accurate design collateral consistent with the fast integration approach, and this design collateral is natural fallout of the design process. For this integration collateral to be realistically produced, the task of creating it is useful at the block design level, meaning that the block designer performs these tasks because they are useful, not solely for integration tasks to be performed.

As stated above, a well-defined simulation and physical design strategy is imperative to achieving design predictability. Each domain process needs to employ this top-level strategy by producing the silicon-accurate collateral necessary to support a fast design process through the block and chip integration levels.

The block-level design process can be separated into several domain-specific flows that produce the appropriate design collateral for integration. To support continuous evolution, these flows produce collateral useful for the integration task throughout the design process. Continuous evolution relies on the ability to use design collateral at all stages in the design process to catch problems early.

These flows break down into: AMS block creation, RF IC, and custom digital. The blocks then are used at integration, which can be categorized as an AMS top-level flow (for simulation), a system/IC flow (for verification within a system-level context), and a physical chip integration flow (for physical design and chip finishing).

6.1 CUSTOM DIGITAL

The custom digital flow comprises digital logic designed in the full-custom process at the transistor level. Most often these custom digital blocks are then integrated with predominantly standard cell blocks, where the custom process was used to decrease area/cost or increase performance. The custom digital domain also includes cell library generation (cells generated at the transistor level using the custom process) and memories/arrays.

Integration outputs for the custom digital process include digital HDL behavioral descriptions, a transistor-level database for simulation, and a layout database supporting abstracts and GDSII. In some cases, custom blocks and cells are characterized by producing timing files for static timing analysis, which is collateral used in the digital standard cell-based environment. Digital HDL descriptions can provide the basis for mixed-signal simulation, and in some cases transistor-level descriptions can support mixed-signal simulation.

All design domains require fast simulation, especially when a particular piece of the design is likely to undergo repeated iterations. Simulators are usually of the FastSPICE class or SPICE-level class. FastSPICE simulators are very conducive to transistor-level digital designs and can be used as the simulator of choice with little accuracy degradation. The fast runtimes for FastSPICE simulators allow iterations on large pieces of the design to be simulated. The high capacity of FastSPICE simulators supports larger parasitic backannotation. SPICE simulators can augment this process. When critical performance issues arise, it may be desirable to simulate a portion in SPICE to achieve full accuracy. SPICE simulation can also be valuable as a final check. Although the simulation runtime is very slow, when taken out of the iteration bottleneck this “second set of eyes” can be very useful for performance-critical designs. SPICE simulation is also desirable to use in the case of memories, because it characterizes memory/array cells in the most accurate fashion. When these array cells are tiled together properly, the huge transistor-level database is most effective and can then be verified through the FastSPICE simulator (see *Figure 7*).

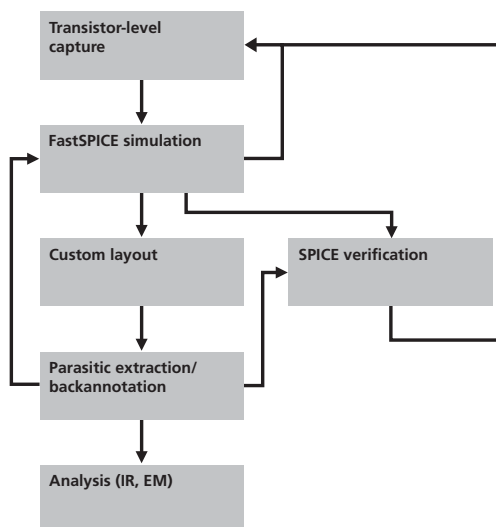


Figure 7: Custom digital design/verification loops

Simulation management is also a key factor in achieving both fast design process and silicon accuracy, especially when characterizing blocks for use in other flows. All parameters must be analyzed, test setups must be defined in a repeatable way, and datasheet documentation should be produced for review. Designers must ensure proper corners are run at the transistor level. This process allows for continuous regressions to be performed, since it is assumed that the design is going through a number of iterations as issues are found. Not only are block-level simulations performed, but block-level verification is performed in a chip-level context consistent with the top-level simulation plan. At each iteration, the regression suite should verify any minor changes to the blocks at the block level in parallel with verification at the chip level, as shown in *Figure 8*.

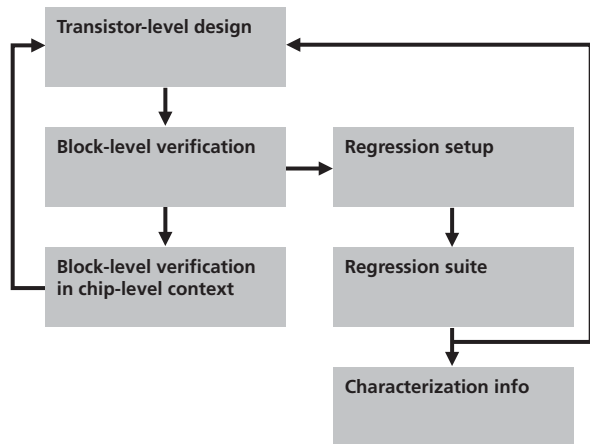


Figure 8: Block-level iterations

Custom digital designs typically have a large device count when compared to analog and RF designs. This makes layout productivity a key issue to address. Layout productivity is most enhanced through leveraging connectivity information and producing a repeatable process. Repeatability needs to be the focus, since custom layout designers typically finish a block and then get last-minute design changes requiring re-layout. In a pure handcrafted layout, it can be a huge task to insert just one transistor because hours or days went in to optimizing the layout for area reduction. Therefore, a tradeoff must be made between schedule concerns that rely on a repeatable and changeable process and a multishot approach at pure handcrafted layouts. This tradeoff is also dependent on design types. For example, a cell library or memory cell, whose cells have a high usage rate, would most likely rely on total handcrafted attention, as the leverage from this activity would be the highest. For a DSP section, such as an arithmetic operation block, the large transistor counts and lower instance counts of the block itself would make repeatability the key issue because designers would expect a number of design changes as the iteration process proceeds.

To support a repeatable process, designers use layout automation that can support more rapid change as the design progresses. A methodology that supports more thorough verification of each design piece will find errors that require fixing, making efficient layout turnaround even more important. This means leveraging connectivity information (contained in the schematic), device information from the schematic (allowing for automated device generation), device placement, and automated device-level routing, as shown in *Figure 9*.

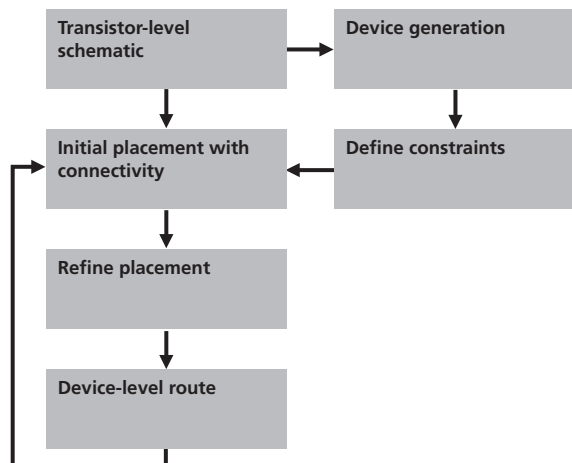


Figure 9: Device-level layout for custom digital design

Silicon analysis, primarily of IR drop and EM, is performed after block layout is complete. Parasitic extraction is performed post-layout and is used as part of the simulation environment described above.

At this point, the custom digital methodology outputs the following collateral for integration:

- Digital HDL descriptions
- Regression suites
- Characterization information for use within other flows
- Layout database for the top level
- Repeatable layout process for further design changes
- Transistor-level database, possibly with parasitics (if called for by the top-level simulation plan)

6.2 AMS BLOCK CREATION

Analog design consists of a wide range of circuit types from low to high frequency. Analog blocks such as ADCs, DACs, and PLLs contain a significant amount of custom digital components but are still considered part of the analog domain. However, in high speed PLLs found in optical networks, it is hard to distinguish a digital component exhibiting analog behavior due to its high frequency and needing a full-fledged analog design process. Other analog blocks such as amplifiers, bandgap references, and active filters can be considered pure analog.

The analog process borrows a number of concepts from the custom digital process, although the nature of analog design can place additional requirements on the design process. The flows differ most in their approach to simulation, because analog designers often need to design to more corners and are likely to be concerned with process effects due to higher circuit sensitivity. *Figure 10* depicts the analog block creation flow.

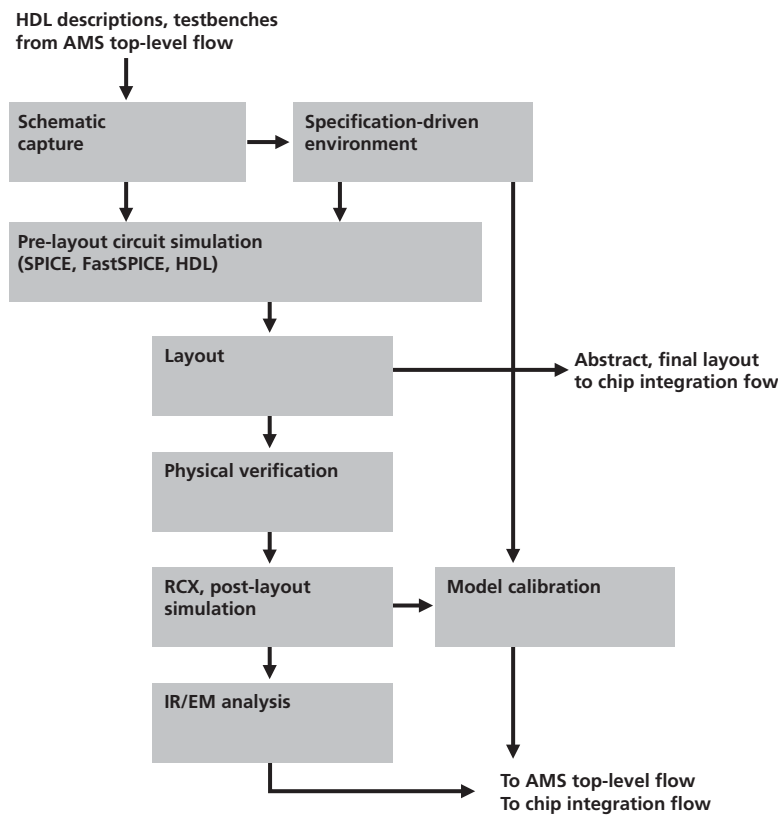


Figure 10: Analog block creation flow

In large analog blocks consisting of many analog components, simulation is long and tedious. Often a full simulation of an analog front end does not reach completion or doesn't even get run. To facilitate full simulation, the analog HDL models from the top level and the testbench setups can be used as a starting point. The first step is to take the HDL model and partition the analog section into the next level of hierarchy at the HDL level. A mixed-level plan is then developed to simulate the entire analog front end. Transistor-level design can then occur for each sub-block, which are verified within the analog front end and at the full-chip level. It is often the case that multiple analog

blocks need to be simulated at the device level at the same time that closed loop interaction between those blocks occurs—and the analog section's simulation plan must account for this.

Traditionally, SPICE-class simulators have been the simulators of choice for analog designs. Because of the extreme sensitivity of many analog circuits, analog designers believe high silicon accuracy is paramount, even at the expense of simulation speed, and they find it difficult to trust other simulators. Yet, the speed and capacity of FastSPICE simulators, and especially their ability to backannotate parasitics, is alluring. This leads to a slight modification to the custom digital approach to simulation, for which SPICE is used at the leaf-cell level where the number of devices is small. Since it is impossible to run a huge number of devices in SPICE, it becomes a worthy tradeoff to run a FastSPICE simulator on the leaf cells and ensure proper functionality, with perhaps some compromise on accuracy. Furthermore, when running large simulations, FastSPICE simulators can often be used as a verification tool for the analog section.

In parallel, transistor-level results can be backannotated into analog HDL models to produce calibrated HDL models. These models will not capture all the silicon accuracy effects found at the transistor level, but post-layout results (either from SPICE or FastSPICE) can be captured to produce a much more accurate HDL simulation at very fast simulation speeds. This also enhances the mixed-level options, because no matter what transistor-level simulator is being used, these simulations are hard to set up and take a significant amount of time to run. There is great advantage to capturing the results in HDL at the top level when simulating with large amounts of digital logic. It is important to have an HDL option to the mixed-level arsenal, and it benefits the analog designer natively.

Simulation management is a huge issue to the analog designer, who has many corners to manage and multiple design results to verify. As in the custom digital domain, full repeatable regression suites are necessary to produce and document results for use in reviews or for annotation to HDL. Supported abstraction levels are fed continuously to the top level, because top-level simulations are run with more and more annotated design data.

The analog layout process is similar to the custom digital process. Automation is used judiciously, although the same tradeoff between repeatability and full handcrafted layout needs to be considered. This is especially true in analog layout, where many constraints such as matching and signal pairs must be managed. The advantage to capturing these constraints and maintaining a repeatable process is preserving “designer IP” in which the integrity of the design is preserved—even if the designers change. This is most important in complex and larger circuits. When combined with layout templates, a rapid analog layout flow can be employed to greatly reduce the time it takes to generate an analog layout.

The other tradeoff is in the quality of results from a rapid analog layout flow vs. a handcrafted analog layout flow. A rapid analog layout flow allows more realistic abstracts to be fed to the top-level route and continuously hones that process. And, the sooner a layout surfaces, the sooner parasitic information can be generated and tested. Should a design team be convinced that the results of this process can be improved through handcrafted layout (perhaps some parasitics cause the design to not meet specification), the option exists to use the “handcrafted flow” with automation assistance in spots. The layout produced from the rapid flow can either be tweaked if it is close, or re-done if large scale changes are necessary (see *Figure 11*).

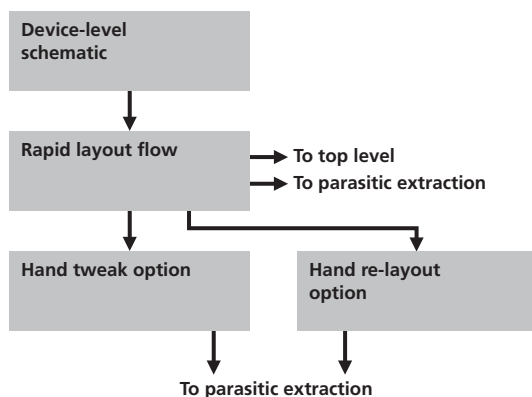


Figure 11: Analog layout options

Parasitic effects are of great concern to analog designers, who rely greatly upon RC extraction. Designers must use their judgment to determine how distributed the RC network is; highly detailed analysis should only be run on selected nets.

Silicon analysis, especially IR and EM, is performed at the block level to identify problems early and aid in chip-level analysis. Substrate noise analysis is performed where high-speed design is affected by noisy digital content.

At this point, the analog methodology produces the following collateral for integration:

- Analog HDL behavioral descriptions
- Calibrated analog HDL descriptions
- Device-level database, possibly with parasitics
- Layout database
- Regression suites

6.3 RF PROCESS

The RF process adds yet more complexity due to designs at very high frequencies. The RF area is most prevalent in wireless designs and consists of receiver and transmitter chains. A wide range of physical implementation options exists in RF, where boards, systems in package, and ICs all play a role. The ACD Methodology is primarily focused on IC design, but is applicable to a variety of integration strategies.

Sample block types in RF consist of LNAs, mixers, VCOs, power amps, filters, and so on. Both time domain and frequency domain analysis can be performed on these circuits, often at the designer's discretion on which to use—as such, the methodology needs to support both types of analysis. The device counts for RF blocks are typically low, yet the simulation times are very long due to the detailed analysis performed.

A particular challenge in RF design is capturing silicon-accurate second- and third-order effects, which can affect circuit behavior. This leads to a modeling challenge and an integration issue. Analog HDL languages do not support detailed RF effects, which can lead to overly optimistic models. Yet, HDL is still useful as long as the designer is aware of this caveat. Only counting on the HDL as a testbench setup, to help run simulations at the block level in a chip-level context, can lead to fuller coverage and faster simulation time. Relying on these models to capture detailed effects will most likely be unsuccessful. As a result, for example, a receiver IC designer really has no choice but to simulate the entire circuit at the device level, but can pass off an HDL description to integration in order to support transient analysis.

In contrast with analog and custom digital simulators, FastSPICE simulators are not as practical when simulating RF effects across the board, nor do they operate in the frequency domain. To speed simulation, designers may use envelope-based analysis, which accelerates simulation of circuits that have a combination of very high and low frequency signals. Envelope simulation can be performed in both the time and frequency domains.

Additional effects of physical design also have a greater impact on RF circuits than those often found in analog. This necessitates that the RF designer include silicon-accurate inductance to verify RC during parasitic extraction when simulating in the time domain, or to consider substrate coupling and transmission lines during frequency domain analysis. Spiral inductors are common in RF circuits, where the designer may need to use a field solver and electromagnetic simulation to capture S-Parameters, producing a special model for simulation. Electromagnetic simulation is also used with pre-declared nets for highly accurate simulation.

Silicon analysis is a big concern for RF designers. Substrate coupling must be considered during block design and when integrating noisy circuitry. IR/EM analysis must also be performed.

Due to the lower device count, layout automation is less a concern in RF design, although connectivity-based layout and device generation enhances productivity. The RF domain is highly sensitive to physical topologies; as a result, there are a huge number of iterations between layout, extraction, and circuit design, and optimization of an RF circuit is tedious (see *Figure 12*). During each iteration, RF designers must consider all physical design effects to ensure silicon accuracy.

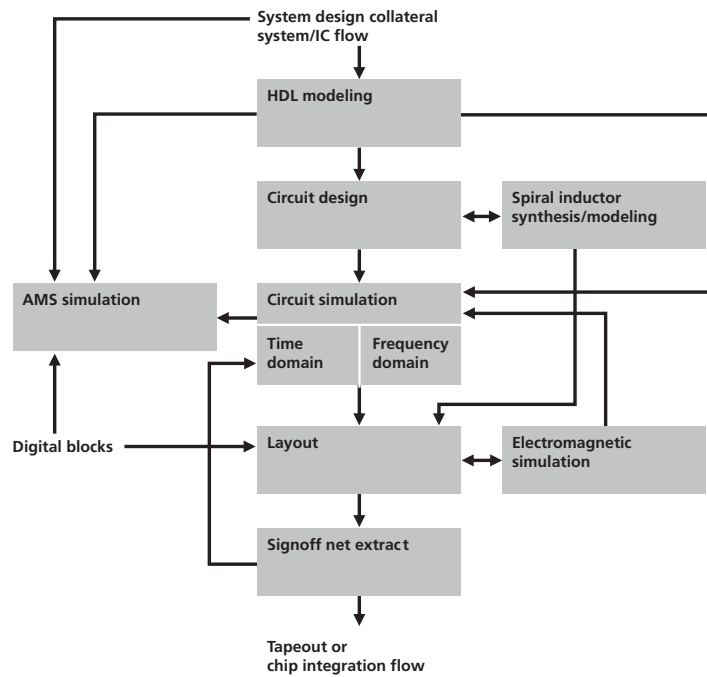


Figure 12: RF IC flow

At this point, the RF methodology produces the following collateral for integration:

- HDL behavioral models
- Device-level database
- Layout database

6.4 SIMULATION AND PHYSICAL INTEGRATION FLOWS

The chip integration process relies on the individual design domain processes to produce the collateral necessary to be successful. It is assumed that chip-level activity started from the very beginning, problems were ironed out, and block data has been passed to the chip environment continuously throughout the design process. By executing this strategy, when it is time for the final chip integration tasks, there are no unknowns and the process is well defined and predictable. The result is a fast design process built on silicon-accurate design collateral.

Designers can leverage system-level collateral by using the system/IC flow. Here, system models, testbenches, or perhaps co-simulation with system-level simulators allow for the IC to be verified within a system-level context. System models provide the highest level of abstraction and thus allow for fast simulation times, while IC data can be inserted through mixed-level simulation to achieve appropriate accuracy levels. Calibrated models prove especially valuable in this mode, as the HDL code contains information from transistor-level simulation (even post-layout if required). This allows IC and systems designers to exchange design collateral and verify performance budgets, or negotiate actual performance budgets based on simulation data. Figure 13 illustrates the system/IC flow.

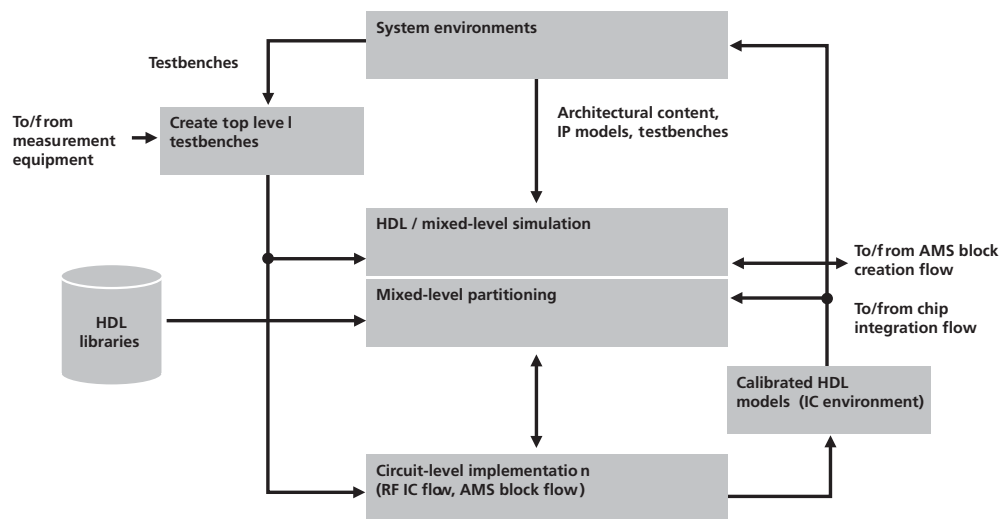


Figure 13: System/IC flow

At this point in the design process, the simulation plan should already be in effect, and final block-level data is delivered for final top-level regression runs. The block designers should have verified their blocks at lower abstraction levels in a chip-level context and verified that the HDL models still match their results. Should transistor-level abstractions be necessary, block designers should have specified this constraint to the top-level simulation plan by this time, and they should have specified the simulator to use for the transistor-level abstractions in the mixed-level specification.

Next, a final regression suite at the top level is run to the simulation plan. To ensure consistency, it's a good idea to re-run each block's regressions in parallel with their block-level simulation plans at the transistor level. Top-level parasitics, which initially were examined and evolved throughout the process, are re-examined at this point with the final data in—how and where the re-examination occurs has already been determined.

The AMS top-level flow pulls the collateral together and provides a simulation and verification environment (see Figure 14). It supports continuous design evolution by allowing regression suites to be run continuously as the design progresses. Block-level flows input collateral, be it transistor, HDL, or calibrated HDL descriptions. Physical flows input block-level interconnect parasitics.

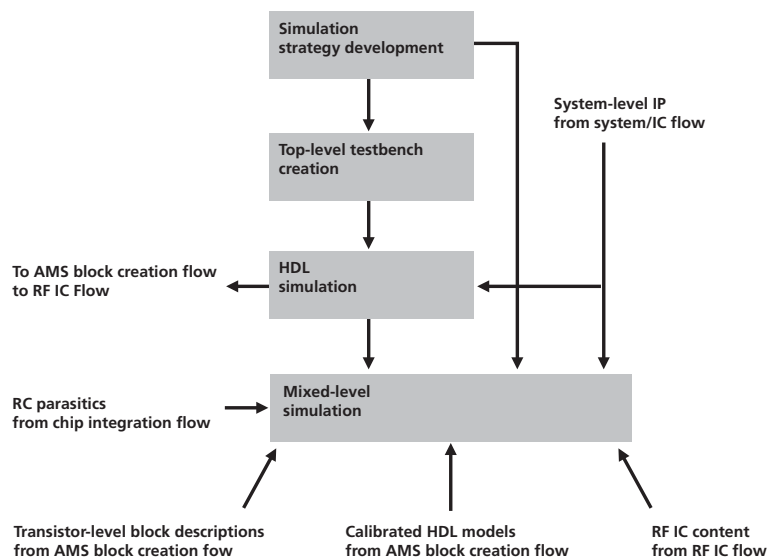


Figure 14: AMS top-level flow

Top-level physical integration, already smoothed through the continuous evolution process, takes in the final physical design databases. Here, standard cell blocks can be incorporated for integration into the process and

treated as another top-level block. Final routing (including global routing) is performed, and final parasitics are analyzed. Final top-level layout verification is necessary across the entire database. Depending on confidence level, final layout verification may need to be performed flat at the GDS level. The physical chip integration flow is shown in *Figure 15*.

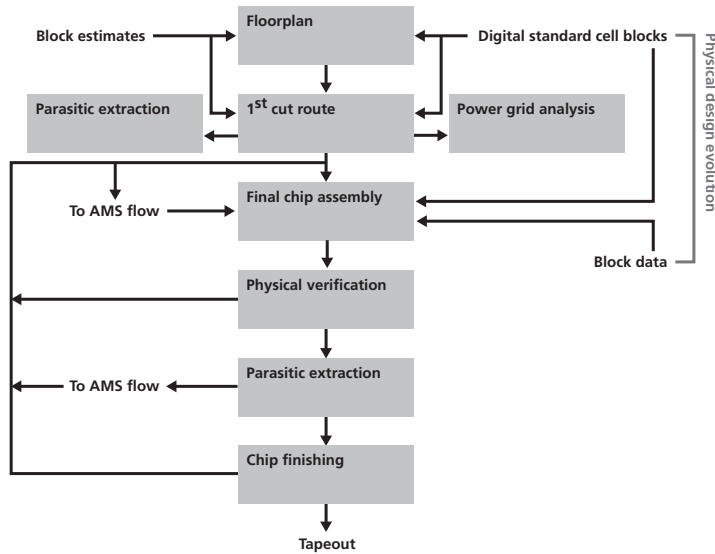


Figure 15: Physical chip integration flow

Because the design team took the time to ensure the proper evolution of the design data, this integration step runs like a well-oiled machine as tapeout draws near. The final integration steps primarily consist of updating small changes to block layouts, performing any necessary re-routing, and folding in the detailed design data to run full-chip DRC/LVS. With this process pre-rehearsed, the resulting integration is a predictable and complete process consistent with the top-level physical and simulation design plans.

It is important to note that top-level integration from an analog perspective assumes that digital portions of the design are black boxes and will not be routed flat. If the design team concludes that digital content should be handled at the top level in a different manner (i.e., flat), then it is likely that the digitally dominated physical design platform will be used to perform the top-level integration.

With a complete top-level database now in hand, designer can perform chip-level post-layout analysis, consisting of RC extraction and analysis of EM, substrate noise, and IR drop. Assuming a high degree of confidence exists that no major design changes are forthcoming, the chip database goes into its final iterations for tapeout. It is quite possible that manual edits and tweaks at the chip level may be necessary, especially if based on the results of final post-layout analysis, which might require spot editing. Other tasks to prepare for tapeout include adding PG text, layer editing, adding copywrites/logos, and metalfill.

7 IP MIGRATION, RE-USE, AND LEVERAGE

With the advent of SoC design during the 1990s, much was made of IP re-use. The hope was that IC design would be largely an assembly process of pre-designed IP. This was partially realized in digital standard cell areas, where IP migration and re-use methodologies were best applied.

In analog (and especially in RF) design, however, the term “re-use” can be controversial. As such, most analog and RF design teams consider the concept of re-use to be applicable to them, but not as cleanly as in digital design. However, while the concept of re-use and subsequent “pain-free IP migration” across process nodes for high-performance analog and RF design is dubious, the concept of leverage and repeatability by incorporating designer IP is not only valid, but critical for a predictable design.

Direct IP re-use in the analog and RF domains is possible when a derivative is based on the same process node technology and the block functionally meets the new requirements and does not need re-design. In this case, the design can be used as is, and the appropriate design IP can be directly accessed for the integration. Often, for second sourcing or area reduction for cost purposes, the design is migrated to new process technologies. In this case, high-performance analog and RF IP quite likely will not scale and perform correctly, as is possible in automated scaling approaches in the digital design area. Leveraging design IP—circuit topology, testbenches, and specifications—is

the key to speeding up the design tweaks necessary to complete the migration effort and produce a faster design process.

As design teams move to smaller process geometries, they move from developing multichip systems to single-chip systems. As a result, much of the circuitry on each new chip can actually be re-used. The ACD Methodology accommodates this trend in the same fashion as IP moving from derivative to derivative (see *Figure 16*).

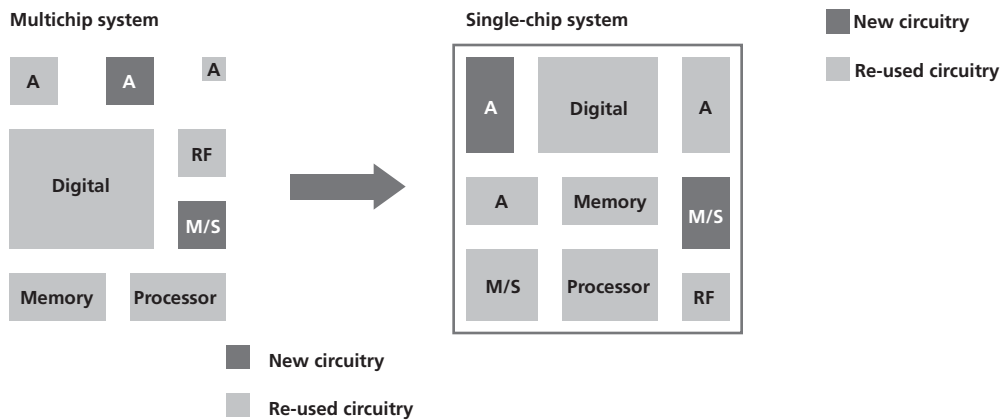


Figure 16: Re-use becomes critical in mixed-signal designs

Adherence to the ACD Methodology not only assures a predictable and successful first pass, but also ensures that the “designer IP” associated with design pieces is maintained. While it is true that analog and RF migration processes can result in significant re-design work, many of the assumptions and decisions a designer made at one process node are likely to apply at another. As such, information like critical nets, matching characteristics, partitioning strategies, base topologies, and noise strategies are useful for the next derivative. Designer IP also includes testbenches and measurements driven from a specification-driven environment.

Simulation test setups and simulation regression suites are likely to be re-usable as well, even though the specifications themselves are likely to change. This ensures that the same full simulation coverage exists for the target design as for the original.

Leveraging designer IP and maximizing re-use throughout the design process greatly increases predictability. Even in cases where a straight migration is not performed, designers can expedite design tweaks or minor changes by relying on previous experience.

8 SUMMARY

Thorough upfront planning, continuously updating the top-level design, and taking advantage of block-level data as it becomes available are key factors in successfully pulling together a complex IC. The overarching goal is predictability, which relies on a fast design process based on silicon-accurate design collateral. Performing tasks in parallel and identifying and addressing complex issues early in the design process lead to a predictable and thorough approach that results in meeting schedule and a good chance for first-pass success. Silicon-accurate design data, verified at the beginning of the design process, ensures that the chip will correlate to silicon and time-to-production can remain on track. All of these characteristics together constitute the fastest road to predictable design.

Diligent adherence to the ACD Methodology also introduces repeatability into the process. In the majority of cases the design will produce a number of derivatives or be migrated to a new process technology where leveraging the process will result in tighter schedules and similar first-pass success rates. Any derivative should add to or modify the original top-level simulation and layout plans where necessary, and strictly follow them throughout the design process just as on the original. For new designs, block plans, databases, and regressions, can be re-used in different configurations on new designs.

In the end, the simulation and layout strategies, critical net parasitic analysis, regression suites across the hierarchy, and layout constraints comprise designer IP that must be preserved to be repeated—especially when it is likely that different designers will be working the database over time. Well-controlled and documented design databases—natural fallout of the design process—strengthen legacy IP and reduce risk. The end result is a predictable design process and a successful design team.



Cadence Design Systems, Inc.

Corporate Headquarters

2655 Seely Avenue

San Jose, CA 95134

800.746.6223

408.943.1234

www.cadence.com

© 2005 Cadence Design Systems, Inc. All rights reserved. Cadence, the Cadence logo, Palladium, and Verilog are registered trademarks and Incisive is a trademark of Cadence Design Systems, Inc. System C is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. All others are properties of their respective holders.

5883B 02/05