



WHITE PAPER



USING A LOW-POWER KIT
TO IMPROVE PRODUCTIVITY,
REDUCE RISK, AND
INCREASE QUALITY

SPEEDING THE ADOPTION OF ADVANCED LOW-POWER TECHNIQUES FOR TEAMS OF
ALL EXPERIENCE LEVELS

INTRODUCTION

In recent years, power consumption has moved to the forefront of ASIC and system-on-chip (SoC) development concerns. The combination of higher clock speeds, greater functional integration, and smaller process geometries has contributed to significant growth in power density. Furthermore, with every new process generation, leakage power consumption increases at an exponential rate.

A wide variety of techniques have been developed to address the power problem and meet evermore-aggressive power specifications. These include the use of clock gating, multi-switching threshold (multi-V_t) transistors, multi-supply multi-voltage (MSMV), substrate biasing, dynamic voltage and frequency scaling (DVFS), and power shut-off (PSO).

Applying any of these techniques requires a substantial amount of effort, introduces risk, and can significantly increase the complexity associated with design, implementation, and verification. In the case of MSMV implementations, it will be necessary to consider low-power cells, level-shifter cells, and so forth. Similarly, in the case of PSO, designers have to choose between “simple power shut-off” (where everything in the block is powered down) and “state-retention power shut-off” (in which the bulk of the logic is powered down but key register elements remain “alive”). This latter technique can significantly reduce the subsequent boot-up time, but state-retention registers consume both power and silicon real estate. The vast majority of designs require a combination of low-power techniques to meet the required timing and power targets—but using multiple techniques concurrently can result in an extremely complex design flow.

“Low power” isn’t just something that can be “bolted on” at the end of the development process. Power, timing, and area are overlapping and potentially conflicting goals that need to be balanced throughout the flow. To meet aggressive design schedules, it is no longer sufficient to consider power only in the implementation phase of the design. The size and complexity of today’s ICs make it imperative to consider power throughout the design process, from the chip/system architectural phase; through the implementation architecture phase; through design (including micro-architecture decisions); and all the way to implementation with power-aware synthesis, placement, and routing. Similarly, to prevent functional issues from surfacing in the final silicon, power-aware verification must be performed throughout the development process.

To address low-power issues, today’s high-end design environments boast a wide variety of sophisticated power-aware tools and methodologies. However, obtaining the full advantage from many of these environments requires members of the design team to be experts in all aspects and stages of low-power design, and this level of expertise is rare.

This paper describes the concept of a “low-power kit” that enables teams with and without low-power expertise to adopt advanced low-power techniques efficiently and effectively. The result is to boost productivity by optimizing engineering and verification resources, reduce risk, and increase the quality of the final product.

HIGH-LEVEL VIEW OF A LOW-POWER KIT

A variety of different considerations must be taken into account when creating a low-power kit. First and foremost, it is necessary to create a representative design that embraces all aspects of a class of typical designs. For example, consider an IEEE 802.11-based wireless access device comprising various interfaces and peripherals. At the core of this device will be an SoC containing a variety of functional blocks. A block diagram of the functions comprising a representative design of this class could be as shown in *Figure 1*.

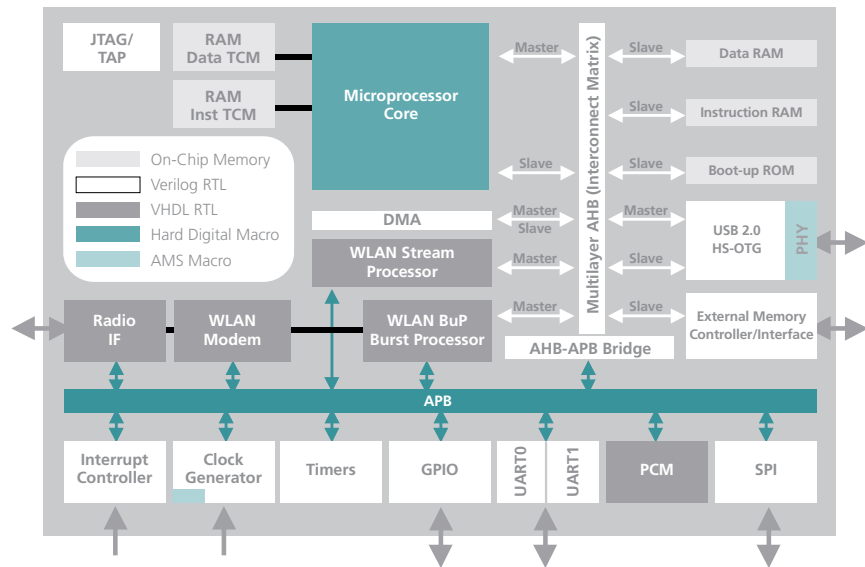


Figure 1: Functional block diagram for a representative design

As we see, a representative design of this class will usually include some hard macros such as a microprocessor core. It will also include a variety of third-party IP, such as blocks of specialized memory. In addition to all of the purely digital logic, there will be a number of analog/mixed-signal (AMS) blocks, such as a phase-locked loop (PLL) in the clock generator and any physical-layer (PHY) interfaces.

Of the other blocks – which may comprise a mixture of internal and third-party IP functions coupled with blocks that are custom-designed for this project – some will be presented in the form of Verilog® RTL, while others will be represented in VHDL.

Now consider a typical power plan associated with this representative design, as illustrated in Figure 2. In this case, the majority of the blocks, which are running at some specific voltage, are targets for multi-Vt and dynamic power optimization. Meanwhile, the USB 2.0 High-Speed On-the-Go function will be implemented using an MSMV approach, and the external memory interface and one of the UART functions will be targeted for a PSO implementation.

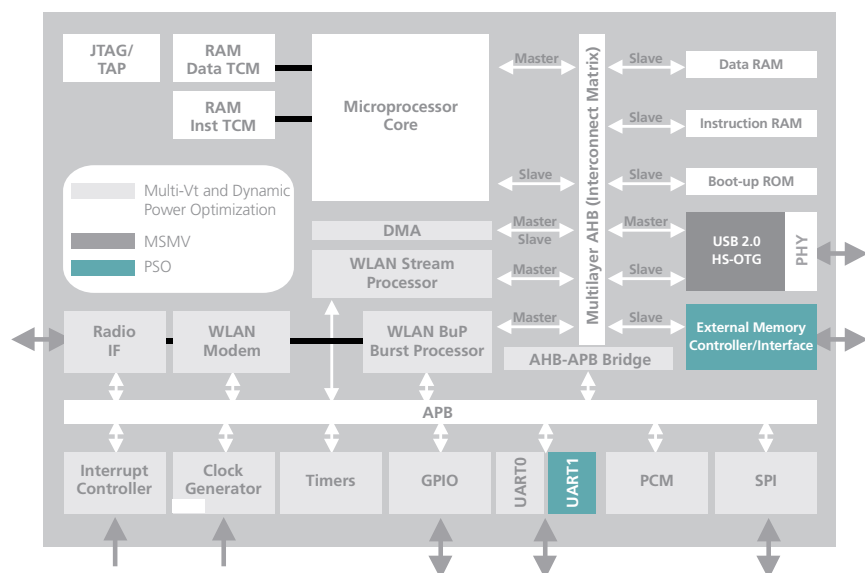


Figure 2: Power plan for the representative design

It's important to note that "low power" is only one aspect of the entire design process. In the case of our representative design, the SoC will be combined with an RF transceiver, and both of these die will be presented in a system-in-package (SiP), as illustrated in *Figure 3*.

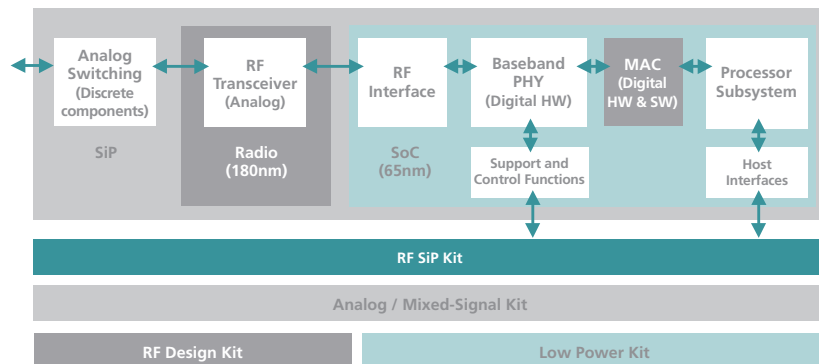


Figure 3: Segment representative design and example methodology kits

This means that, in addition to the SoC itself, such a project will involve designing, implementing, and verifying the RF transceiver and the SiP. Thus, a number of design kits will be required, including:

- A low-power kit
- An analog/mixed-signal kit
- An RF design kit
- An RF SiP kit

The low-power kit should comprise a number of discrete modules, each focused on a particular facet of the low-power design process. For example, one module will concentrate on the MSMV aspects of the design; another module will focus on PSO, and so forth. This means that design and verification teams need use only those modules that are applicable to their particular design problems.

Where applicable, each module forming the low-power kit should include:

- Background information pertaining to this aspect of the design
- Best practices
- Checklists
- Recommended flows
- Training
- Documentation

Furthermore, the fact that the kit will be based on a representative design means that – in addition to the design files themselves – the kit can include all of the scripts required to drive the various design, implementation, and verification tools throughout the entire process, from capture to tapeout.

There's clearly no "one size fits all" when it comes to a design kit. Thus, in addition to the representative design, methodologies and flows, infrastructure, and IP, the low-power kit must also include appropriate consulting services that can adapt it as required to address each customer's specific needs.

COMMON POWER FORMAT

A key enabler of a modern power-aware design flow is the ability to capture and preserve the design intent of the chip throughout the design flow. This requires a common specification format that can be used and shared across the entire design chain, from architectural specification (“this block has three power modes”) to verification (“will the chip recover if these blocks are put to sleep in this order?”).

A low-power kit should take full advantage of such a specification to capture and communicate power-related information between the various modules forming the kit. Among other things, these modules will use the specification information to automatically configure the design (adding appropriate level-shifter and isolation cells, for example) and control tools in the flow (verifying that PSO blocks are powered-down and up in a certain order, for example).

The current state-of-the-art in such a specification is the Common Power Format (CPF), managed under the auspices of the Silicon Integration Initiative (Si2) consortium’s Low-Power Coalition. CPF is a new design language—hardware description language (HDL)-neutral—that addresses limitations in the design automation tool flow. It provides a mechanism to capture architects’ and designers’ intent for power management and it enables the automation of advanced low-power design techniques and “what-if” exploration. CPF provides a holistic specification of power intent that allows all design, implementation, verification, and technology-related power objectives to be captured in a single file. CPF then applies that data across the design flow, providing a consistent reference point for design development and production.

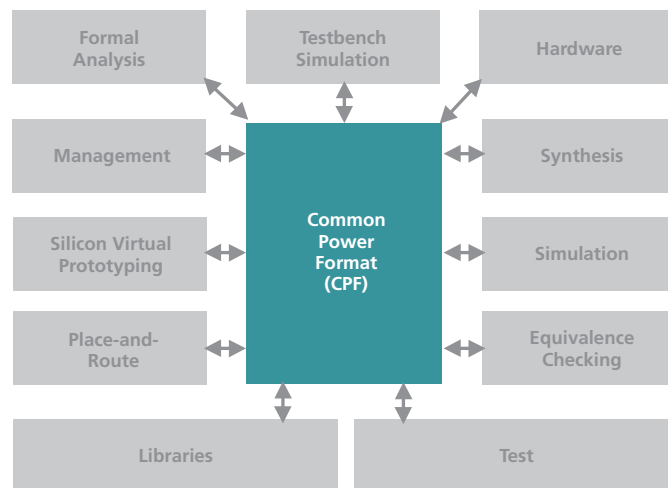


Figure 4: CPF provides a holistic specification of power intent used throughout design, implementation, and verification

Requiring no RTL changes, CPF facilitates automated low-power functional verification, an automated MSMV methodology, and an automated PSO methodology.

There are three major benefits of using CPF to drive the design, verification, and implementation steps of the development flow. First, it helps designers achieve the required chip specs by driving the implementation tools to achieve superior tradeoff among timing, power, and area. Second, by integrating and automating the design flow, it increases designer productivity and improves cycle time. Third, by eliminating the need for manual intervention and replacing ad hoc verification methodologies, it reduces the risk of silicon failure due to inadequate functional or structural verification.

Regarding CPF, the low-power kit should include features such as:

- Guidelines for CPF file creation
- Checks to ensure CPF files are complete
- A migration methodology from a non-CPF flow to a CPF flow
- A generic CPF file template
- A specific CPF associated with the kit's representative design
- Product-specific scripts that illustrate how CPF is used

MODULARIZATION AND FLOWS EASE ADOPTION

To facilitate understanding by an end user, the low-power kit should be partitioned into a number of main categories; for example, Design Environment, Low-Power Techniques, Design Creation, Physical Implementation, Verification, and Management (see *Figure 5*). In turn, each of these categories will comprise a number of discrete modules – design and verification teams need use only those modules that are applicable to their particular problems. Such a modular approach greatly facilitates the incremental adoption of low-power design flows and techniques.

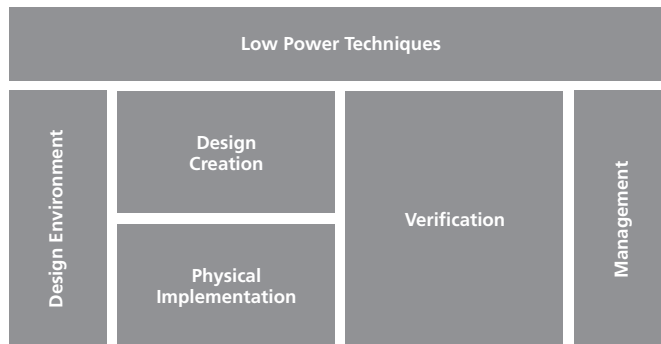


Figure 5: The low-power kit may be partitioned into a number of categories

Furthermore, the low-power kit should feature a number of “flows,” where each flow embraces a methodology, tool use, scripts, and so forth. For example, there might be low-power flows for logic synthesis, design-for-test (DFT) and automatic test pattern generation (ATPG), physical design, functional verification, formal implementation verification, and power-grid signoff, as illustrated in *Figure 6*.

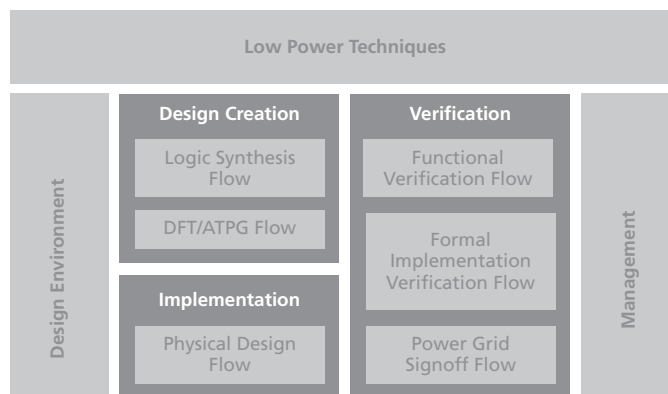


Figure 6: The low-power kit may feature a number of flows

Observe that some categories – such as Design Environment, Low-Power Techniques, and Management – won't have any flows associated with them. This is because the concept of a flow is tied to the ways in which the various design tools are used and the ways in which design data propagates through the environment.

DESIGN ENVIRONMENT

This category covers everything required to establish a low-power design environment, including infrastructure, tools, IP components, and detailed checklists and dependencies.

The modules included in this category might be as follows:

- **Library Qualification** Ensure the necessary view and databases are available for successful low-power flows
- **Process Selection** Understand how low power affects process selection and what is needed based on your design requirements
- **Infrastructure** Define data management and system infrastructure to ensure a smooth low-power implementation

Ensuring that the design environment is established correctly provides the basis for a smooth passage through the rest of the processes and flows. Defining the necessary components for low-power design "up front" allows users to avoid common problems. And any decisions made here will subsequently feed into all of the other components of the kit.

LOW-POWER TECHNIQUES

This category covers the various low-power techniques that may be employed in a design, including best practices, tradeoffs, and re-creatable MSMV / PSO design databases and scripts.

The modules included in this category might be as follows:

- **Multi-Threshold Voltages** When and how to use multi-Vt effectively in a design
- **Low-Power Clocking** As a major contributor to power consumption, managing clocks effectively is key to a low-power implementation
- **Multi-Supply Multi-Voltage** Defines the benefits and pitfalls of MSMV and what is needed to implement it in a design
- **Power Shut-Off** PSO has substantial power benefits, but it needs to be applied appropriately as it affects all areas of the design process
- **Dynamic Voltage and Frequency Scaling** DVFS techniques may also have substantial power benefits, but ensuring that the performance-voltage-frequency feedback mechanism works correctly can be a challenge

The modules in this category define the different low-power techniques, including everything that needs to be considered when implementing these techniques in a design; how the adoption of the various techniques will affect the different modules and flows; the complexities that will be introduced by the various techniques; and the ways by which these complexities may be overcome.

DESIGN CREATION

This category covers all design creation aspects of the flow, including performing architectural tradeoffs among power, timing, and area, along with best practices for success with designs employing MSMV and/or PSO techniques.

The modules included in this category might be as follows:

- **Architecture Tradeoff** Compare different techniques and how they affect total power for a design
- **RTL Design** What can be done at the RTL level to affect power consumption as well as how the different techniques are codified
- **CPF Creation** Effective power intent representation using the Common Power Format
- **Low-Power Synthesis** How synthesis affect the power process and how to implement the most effective synthesis for a low-power design
- **Power-Aware DFT** Power affects both normal and test modes of operation and, as such, must be considered throughout

One of the flows associated with this category will be the low-power logic synthesis flow. Neglecting any feedback cycles, a high-level view of this flow (including appropriate elements from the DFT flow) is illustrated in *Figure 7*.

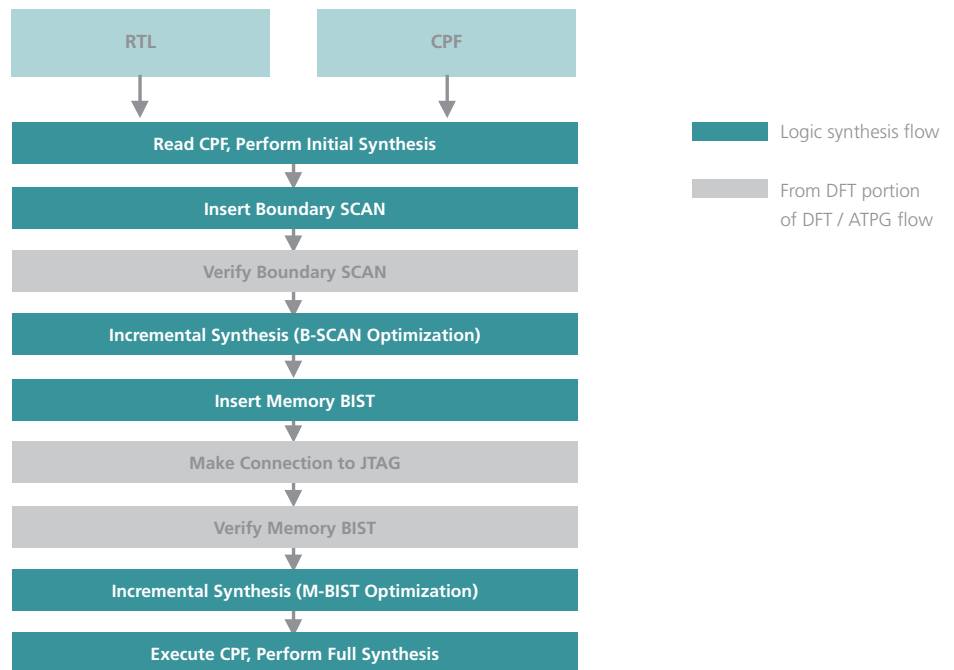


Figure 7: High-level view of the low-power logic synthesis flow including DFT

Just as flows can cross multiple modules, some aspects of different flows may be intertwined. As illustrated in *Figure 7*, the insertion of boundary SCAN and memory BIST elements may be considered part of the low-power logic synthesis flow, while the verification of these elements may fall into the low-power DFT/ATPG flow, as illustrated in *Figure 8*.

PHYSICAL IMPLEMENTATION

This category covers best practices and complete coverage, from netlist through design signoff, including power-aware design-for-test (DFT) and low-power automatic test pattern generation (ATPG).

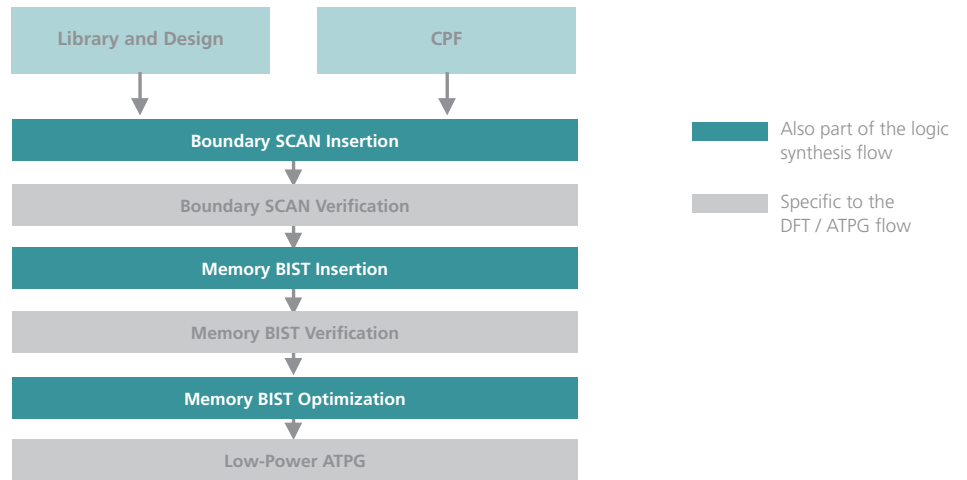


Figure 8: High-level view of the low-power DFT/ATPG flow

The modules included in this category might be as follows:

- **Prototyping and Parasitic Correlation** Many iterations of floorplan optimization may be needed on the road to timing closure. Accurate parasitic scaling between detailed extraction and prototyping extraction speeds this up
- **Power Planning** Proper selection of a power distribution scheme (grid or ring) must be taken into account for low- and non-low-power blocks
- **Low-Power Floorplanning** Determine the best location for sensitive blocks. Rules may govern that blocks with the same voltage are located in close proximity or closer to pad I/Os, for example
- **Timing and Signal Integrity** Even if power intent is considered to be the most important, timing closure and SI specifications must be met to obtain a functional design

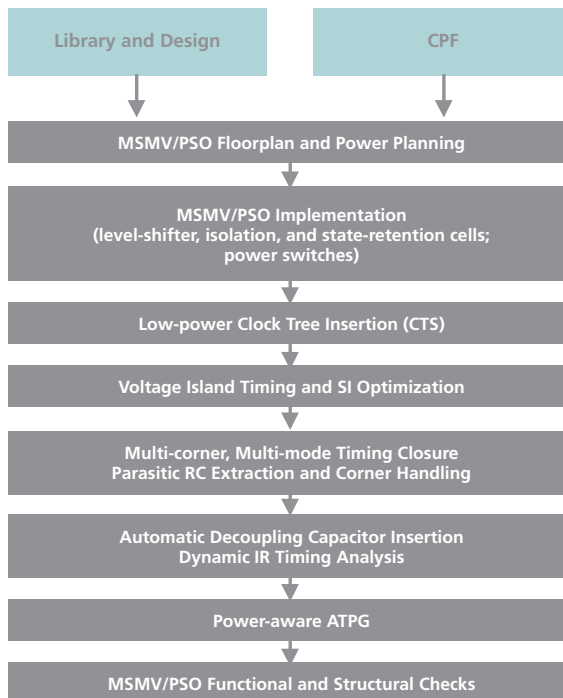


Figure 9: High-level view of the low-power physical implementation flow

Neglecting any feedback cycles, a high-level view of the low-power physical implementation flow is illustrated in *Figure 9*.

VERIFICATION

This category covers all aspects of verifying a low-power design. In the case of a design featuring the PSO technique, it is necessary to verify that the various blocks are powered-down and powered-up in the correct sequence.

The modules included in this category might be as follows:

- **Low-Power Functional Verification** Power intent can directly affect functional intent, so functionality must be verified in the context of the final power implementation using techniques such as simulation and assertion-based verification
- **Low-Power Formal Implementation Verification** Defines a process to ensure that the low-power implementation chosen is correct and consistent through all design transformations
- **Power-Grid Signoff** Prior to signoff, verify that the static and dynamic behavior of the power grid meets all requirements for correct operation

The ability to check the design's functionality based on power intent is paramount from the RTL stage. Starting with power control management, the designer should be able to create assertions easily to check for proper power control sequencing. Often, complex relationships exist between how different chip power modes interact and sequence in the design. Creating assertions is the easiest way to verify this type of behavior in a design. Once specified, either formal or dynamic techniques can be used to verify these power control structures. But PSO, state-retention PSO, and DVFS also present a problem for functional verification. Through a combination of assertions, transactions, and specialized simulator intelligence, the verification automation of these structures and debug of functionality can be greatly simplified (see *Figure 10*).

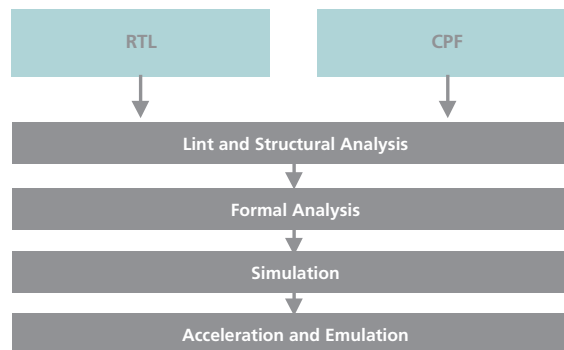


Figure 10: High-level view of the low-power functional verification flow

Throughout the implementation process, checks must exist to constantly monitor not only simple functionality but power-aware functionality. Thus, additional checks must exist to ensure that low-power structures are inserted and connected correctly (see *Figure 11*).

Verifying the integrity of the power grid is critical to a working design. Static rail analysis provides a quick turnaround and is a good start toward ensuring robust power structures. Adding dynamic rail analysis allows the power grids to be verified with more realistic worst-case conditions. If power rails are marginal under static conditions, they may fail under various dynamic conditions such as simultaneous switching. Additionally, correct power up/down sequencing becomes critical for reliable silicon.

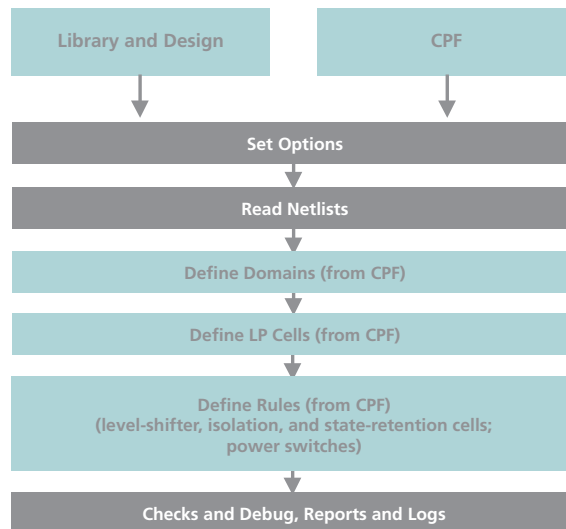


Figure 11: High-level view of the low-power formal implementation verification flow

Last but not least, the ability to capture power grid information about where failures may occur and being able to feed this information back into the flow to drive changes in the power grid design and layout is highly valuable.

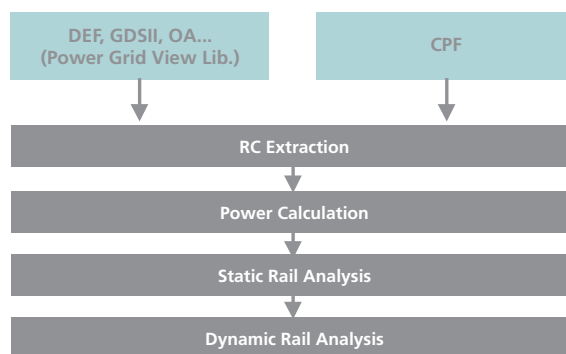


Figure 12: High-level view of the low-power power-grid signoff flow

MANAGEMENT

This category covers the various facets of managing a low-power design, from planning a verification strategy to a low-power engineering change order (ECO) methodology.

The modules included in this category might be as follows:

- **Low-Power ECO Methodology** Track all data and flow dependencies to ensure that late-stage changes do not cause problems
- **Planning, Metrics, and Analysis** Predictability is ensured through proper metrics and planning, and must be considered throughout the design process

This category embraces defining all the data and flow dependencies, both for up-front design management and for changes that occur later in the flow. Good management ensures that all the flows execute smoothly and that appropriate data can be extracted by key project members and managers to gauge the progress of the design.

SUMMARY

Power consumption has moved to the forefront of ASIC and SoC development concerns. A wide variety of low-power design techniques have been developed to address the various aspects of the power problem, but applying these techniques requires a significant amount of effort, involves a certain amount of risk, and can increase the complexity associated with design, implementation, and verification.

To fully benefit from the most advanced low-power techniques while also reducing risk, all members of the design team need efficient access to low-power design expertise. The solution is to use a low-power kit, which combines a real-world segment representative design with methodology, infrastructure, IP, and applicability consulting services that can adapt the kit to address each customer's specific needs.

A low-power kit gathers expert knowledge and best practices to eliminate common problems. Using established flows ensures that the various tools and low-power technologies are applied to achieve the best results; using established processes leads to predictable timelines. With a low-power kit, design teams can reduce packaging and system cost, enhance productivity and predictability, reduce project risk, and improve design quality. The end result of using a low-power kit: an optimized design that meets aggressive power requirements and market windows.

For more information about
this and other products contact:

info@cadence.com

or log on to:

www.cadence.com

cadence™

Cadence Design Systems, Inc.

CORPORATE HEADQUARTERS

2655 Seely Avenue

San Jose, CA 95134

P: +1.800.746.6223 (*within US*)

+1.408.943.1234 (*outside US*)

F: +1.408.943.5001

www.cadence.com