cadence

Building Energy-Efficient ICs from the Ground Up

Considering low-power techniques throughout the development process

Power consumption has moved to the forefront of digital IC development as component sizes shrink and insulating layers on gates become thinner. To enable today's advanced low-power techniques, the design flow must holistically address the architecture, design, verification, and implementation of low-power designs. Cadence offers the design, implementation, and verification tools and flows to address all areas of low-power design throughout the entire SoC development process.

Contents

Introduction.................................. 1
Low-Power Design Techniques........ 3
Creating Low-Power Digital ICs 5
Holistic Power-Aware Design 6
Common Power Format................. 7
Chip Planning—Getting Low-Power Right 7
Power Analysis............................... 8
Top-Down Multi-Objective Optimization.................................. 9
Low-Power Verification Starts Here..................................... 9
Power-Aware Test Considerations..............................10
Moving into the Physical Design Phase 10
Low-Power Formal Verification 11
Physical Implementation of Low-Power Structures...................11
Clock-Gating Challenges................11
Power Grid Implementation and Analysis 11
Power Rail Analysis 12
Signal Integrity Considerations.......12
Proceeding to Signoff 12
Summary 13

Introduction

It's easy to forget that the mobile phones of 10 years ago were just phones. We used to get really upset if we'd used more than an hour or so of talk-time and the battery would not last through the day. Today's smartphone users quickly become disenchanted if their tiny devices don't let them talk for hours, check email, play games, stream videos from the Internet, take video and photos and share them via social media sites—all in a day without recharging. In those 10 years, lithium-ion battery technology did not significantly improve, but as the number of applications increased, the energy efficiency of mobile device technology did.

The need to meet incessant consumer demand for more functionality in handheld devices is just one example of the many market forces driving innovation in ICs. It's not just mobile or battery-powered devices. The industry has also made phenomenal strides in performance and functionality in chips used to power everything from the servers in the data centers powering the "cloud" to digital TV set-top boxes that give viewers hundreds of channels, real-time playback, on-demand streaming, and act as the hub of the home network. Cooling represents a major component of the running costs for data centers, so designers must focus on lowering power density. In consumer electronics, the need to be "green" is not only regulated by specifications such as Energy Star, but energy efficiency has also become a real product differentiator in the "big-box" stores.

Tiny problem

Creating these small wonders requires designers to pack tens of millions, hundreds of millions, or even a billion transistors onto a single sliver of silicon. With every new process generation, leakage power consumption increases dramatically as the size of components shrink and the insulating layers on gates become thinner.

Unfortunately, as process nodes continue to shrink, we are approaching the point where power dissipation is growing exponentially with each new technology node.

Unless designers employ techniques to mitigate leakage power, the current generation of nanometer-scale chips can waste enormous amounts of electricity and dramatically decrease the battery life of portable electronics. At 40nm this wasted power is almost equivalent to the power used to do the useful things the chip was created to do. At 28nm and beyond, the leakage power density is at least equal to the dynamic power density.

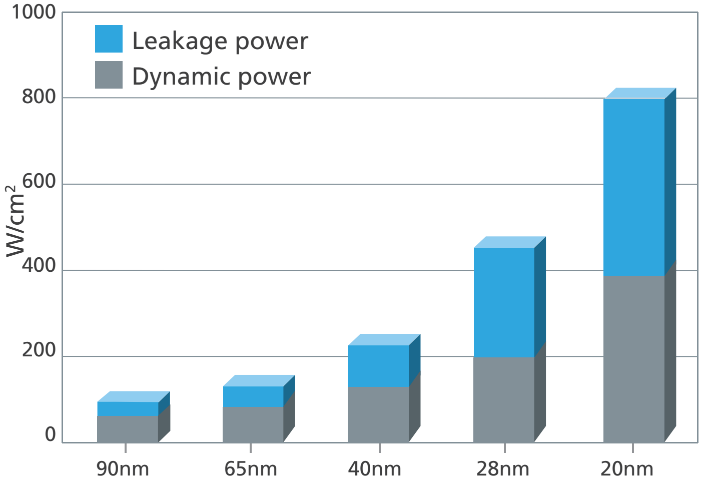

Figure 1: Leakage power becomes a growing problem as demands for more performance and functionality drive chipmakers to nanometer-scale process nodes (Source: IBS).

The combination of greater functional integration, higher clock speeds, and smaller process geometries has also contributed to significant growth in power density. Consequently, reducing power consumption has become a major concern not only for makers of battery-operated devices but also for designers of hard-wired AC-powered products.

Burning issue

Inefficient use of energy isn't the only cost associated with growing power densities and shrinking process nodes. Since energy translates into heat, cooling has also become a hot topic for designers. Figure 2 shows how the increasing power density of Intel's microprocessors translates into incredible heat.

Excessive heat presents a number of problems for electronics makers, including:

- Decreased reliability and shorter product lifetimes—every 10 degrees C rise in operating temperature cuts the mean time between failure (MTBF) in half
- Shorter battery life—heat reduces the efficiency of circuits

- Hotter products require cooling fans, increasing energy usage, noise, and product cost
- More costly chip packaging and other cooling techniques, e.g. heat sinks.

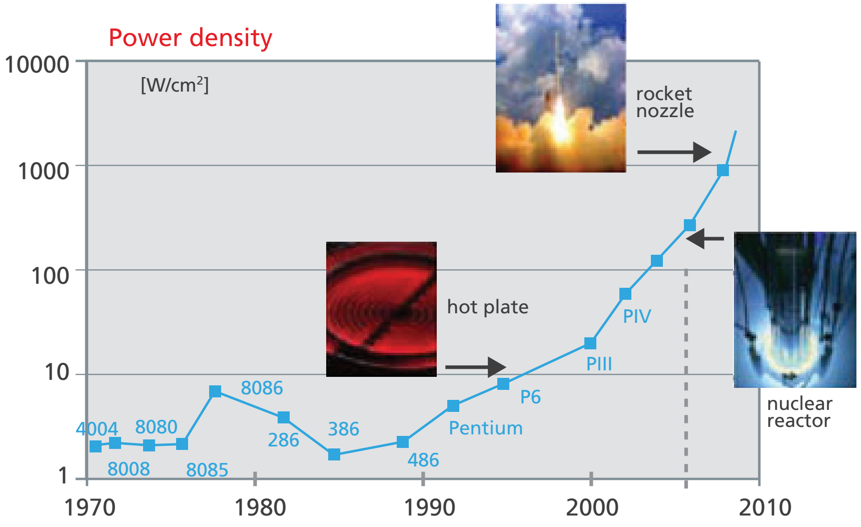

Figure 2: The growing power density (measured in W/cm2) of Intel's microchip processor families. (Source: Intel)

In addition to the business and competitive benefits of creating more energy efficient chips, as consumers and governments increasingly become aware of their respective "carbon footprints," energy efficiency is a growing concern globally. In fact, considering the unacceptable human, economic, and environmental costs of failing to address this problem, creating leaner, greener electronics has moved from an important marketing differentiator to an engineering imperative.

Building energy-efficient designs from the ground up

Not long ago, power considerations were relegated to the down-stream portions of the SoC development flow. In many applications, designers tended to focus on timing closure and signal integrity (SI) issues, rather than the amount of power a chip consumed per se.

By comparison, in the case of today's extremely complex silicon chips "low power" isn't just something that can be simply "bolted" on at the end of the development process. In order 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 SoCs makes it imperative to consider power throughout the entire development process—from the chip/system architectural phase through design (including micro-architecture decisions) 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.

This paper first summarizes the various low-power design techniques commonly used by designers today. It then provides an in-depth look the design, implementation, and verification tools and flows and methodologies that Cadence and other organizations have developed to solve the low-power problem by addressing all areas of power management.

Low-Power Design Techniques

Unless engineers take pains to use appropriate power-saving techniques, today's chips can consume almost as much power while idling as they do operating at full throttle. In order to control both dynamic (switching) and leakage power, engineers can employ various power management techniques.

Over recent years, a wide variety of techniques have been developed to address the various aspects of the power problem and to meet ever-more-aggressive power specifications. These techniques include:

- **Multi-Vt transistors:** Involves the use of transistors with different switching thresholds. In this technique, a cell (logic gate) on a non-critical timing path may be formed from transistors with high switching thresholds (high-Vt) that have lower leakage, consume less power, and switch more slowly. By comparison, a cell on a critical timing path may be formed from transistors with low switching thresholds (low-Vt) that have higher leakage, consume more power, and switch significantly faster.
- **Multi-supply voltage (MSV):** Entails dividing the chip into areas supplied at different voltages and then assigning the various functional blocks forming the design to these different voltage islands. A functional block located in an area supplied by a higher voltage will have higher performance but will consume more power. Locating the same block in an area supplied by a lower voltage will reduce both its performance and power consumption. However, having different voltage islands to which the various functional blocks have to be assigned complicates the floorplanning and placement steps. It is also necessary to insert level shifter cells for the signals passing between different voltage islands—and to account for signal integrity effects between the various voltage domains.
- **Power shutoff (PSO):** As its name suggests, power shutoff refers to completely powering down a functional block that is not currently in use. For example, if a cell-phone contains an MP3 player function but the user is not currently listening to music, then it makes sense to power-down that block. In such a case, the designer must choose between "simple power shutoff" (where everything in the block is powered down) and "state-retention power shutoff" (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 power and also have an impact on silicon real-estate utilization. There are also considerations with regard to powering multiple blocks down and up in a particular sequence (as discussed later in this paper).
- **Standby mode:** Effectively a "halfway house" between MSV and PSO, some blocks may have a standby mode where the supply voltage is reduced to a value that does not allow dynamic operation, but retains existing data. This is commonly used in memories, especially volatile memory. It is also used in "light sleep" modes, since system recovery is quicker than with PSO. Verification tasks usually include testing for correct behavior when circuitry attempts to write data to address ranges in standby.
- **Substrate biasing:** Substrate biasing is typically applied only to portions of the design. The idea here is that a functional block typically doesn't need to run at top speed for the majority of the time. In such a case, biasing can be applied to the area of the substrate associated with one or more functional blocks. This causes the affected blocks to run at a slower speed but with significantly reduced leakage power. Substrate biasing is commonly used in combination with standby mode in memories. Substrate biasing can also be used to control the variability in the design.
- **Dynamic voltage and frequency scaling (DVFS):** This technique is used to optimize the tradeoff between frequency and power by varying the voltage or frequency associated with one or more functional blocks in relatively large discrete steps. For example, the nominal frequency may be doubled to satisfy short bursts of high-performance requirements or halved during times of relatively low activity. Similarly, a nominal voltage of 1.0V may be boosted to 1.2V to improve the performance, or reduced to 0.8V to reduce the power dissipation. Each of these scenarios has to be tested in the context of surrounding blocks, which may themselves switch from one mode to another.
- **Adaptive voltage scaling (AVS):** This technique is similar to the voltage scaling part of DVFS. The difference is that there is on-chip process and temperature monitoring in the control loop that sets the voltage, which is often controlled more precisely in smaller steps.
- **Clock gating:** Refers to disabling the clock inputs to registers in a portion of the design, along with the corresponding portion of the clock tree, when the logic controlled by that portion is not being used. The clock tree can account for a substantial amount of an SoC's total power consumption, so it is becoming common to perform clock gating at multiple levels. This means designers have to decide if clock gating should be performed only at the bottom of the tree (the leaf nodes), at the top of the tree, in the middle of the branches, or as a mixture of all of these cases. There are tools to make this more efficient by moving the clock-gating structures upstream and/or downstream and performing splitting and cloning, but all of this adds substantially to the task of physically implementing the clock tree and also to the verification problem.

Related to this technique is operand isolation. Designs that do not fully utilize their arithmetic datapath components typically exhibit significant overhead in terms of power consumption. Whenever a module performs an operation where the result is not used in the downstream circuit, power is consumed for an otherwise redundant computation. Operand isolation minimizes the power overhead incurred by redundant operations by selectively blocking the propagation of switching activity through the circuit.

Although the use of advanced techniques such as MSV and PSO (and to a lesser extent substrate biasing) can dramatically reduce power consumption, they also increase the complexity associated with design, verification, and implementation tools and methodologies.

Furthermore, while using a single technique in isolation could be relatively simple, often a combination of these techniques must be used to meet the required timing and power targets. Using multiple techniques concurrently increases the complexity of the development flow, thereby mandating a development environment and tools that can adequately address all of these issues. It is also important to match the technique to the type of design. For example, PSO is of no use if the logic is always running but with different loads. In this case, DVFS would be a better option.

Figure 3 illustrates the power, timing, and area tradeoffs among the various power management techniques.

Power-reduction Technique	Power Benefit	Timing Penalty	Area Penalty	Methodology Impact			
				Architecture	Design	Verification	Implementation
Multi-Vt optimization	Medium	Little	Little	Low	Low	Low	Low
Clock gating	Medium	Little	Little	Low	Low	None	Low
Multi-supply voltage	Large	Some	Little	High	Medium	Low	Medium
Power shut-off	HUGE	Some	Some	High	High	High	High
Dynamic and adaptive voltage frequency scaling	Large	Some	Some	High	High	High	High
Substrate biasing	Large	Some	Some	Medium	None	None	High

Figure 3: Tradeoffs associated with the various power management techniques.

Creating Low-Power Digital ICs

The various portions of the power-aware development process may be summarized as follows:

- **Chip/system architectural specification:** Power-aware design starts with the architectural specification of the chip/system, including partitioning the system into its hardware and software components. The system architecture component of a low-power design typically looks at the efficiency of different algorithms and the efficiency of hardware vs. software implementations of a specific algorithm. The choice of IP as well as modeling and estimation are critical in this phase, and this is also where the biggest impact on power is made.
- **Power architecture:** Following the definition of the chip/system architectural specification, the next step in the development process is to refine the power architecture. At this stage, evaluations should be made as to which blocks are not performance-critical, which means they can potentially be run at a lower voltage and/or frequency to conserve power. Similarly, certain blocks may be suitable candidates for “sleep mode” or to be completely shut down to conserve power when they are inactive. For example, the architectural specification may specify that a certain block should be implemented in such a way that it is capable of being completely powered down. In the power architecture portion of the process, the team will determine just how often this block is to be shut down, and also any interdependencies among this block and other blocks and modes.

- **Power-aware design:** In this context, "design" refers to the portion of the flow where—taking the results from the chip/system architecture and power architecture phase—design engineers capture the RTL descriptions for the various blocks forming the design. The designers associated with each block are responsible for ensuring that block will meet its functional, timing, and power requirements while using the minimum silicon real estate. (Note that Cadence® C-to-Silicon Compiler technology can streamline this process by directly generating the RTL from the system specification, if a system-level modeling flow is in use.)
- **Power-aware implementation:** The implementation phase is where all of the work performed during the chip/system architecture, power architecture, and power-aware design phases comes to fruition with the aid of power-aware engines for logic synthesis, clock gating, design-for-test (DFT), placement, clock-tree synthesis, routing, and so forth. In addition, during the implementation phase, the logical and physical structures needed for the various power techniques are created. These include power grid synthesis, power plane implementation, and insertion of level shifters, switch cells, isolation cells, and state-retention cells.
- **Power-aware verification:** Verification commences with the planning process. In this process, every hardware and software element comprising the design that is to be tested is detailed, the way in which each element will be verified is defined, and the required coverage metrics for each element are specified. As functional verification proceeds, coverage metrics are captured and used to guide the verification process until coverage goals are met. In addition to functional verification, the physical verification that occurs during and after placement and routing must ensure that power intent is met before silicon signoff takes place. Throughout the verification process, the questions to be asked are: Did I build what I meant to build? Did the tools do what I told them to do? Does the final result meet all my performance goals?

Holistic Power-Aware Design

A major consideration in low-power design is to ensure that each power-centric implementation technique is executed in a holistic manner, taking into account of all of the other techniques being employed on a particular design. Each technique has to be supportive of the other techniques, and any power savings achieved from one technique need to be maintained and preserved by the other techniques. For example, one of the most basic savings in power is to not use the biggest drivers. A slight reduction in margins will result in a smaller design occupying less area and consuming less power. But if the design environment does not support power-aware timing closure, it may negate any power savings achieved earlier in the process by adding too many cells for delay fixing.

Similarly, the practice of adding extra timing margin to account for arbitrarily set amounts of IR drop will negatively impact power consumption. Thus, once again, it is important that all optimization techniques simultaneously consider all of the implementation objectives to achieve the best possible solution.

While the use of advanced low-power design techniques such as MSV, PSO, DVFS, and substrate biasing can dramatically reduce power consumption, they also increase the complexity associated with design, verification, and implementation methodologies. In the past, this led some project teams to disregard the various low-power techniques and trade-offs because they considered the risks to be too high. Today's design teams require effective methods and technologies to reduce the risks of implementing these techniques.

Use of a standard power intent format such as IEEE-1801 Unified Power Format (UPF) or the Si2's Common Power Format (CPF) is the key to effective power-aware implementation. Cadence provides interoperability with UPF but, for the purposes of this document, we will use CPF to illustrate the flow. While the syntax and semantics differ between the two formats, all the concepts covered here are similar. As described in the next section, CPF captures the intent of chip architects and designers and enables automatic optimizations throughout the design flow. Of particular interest is the fact that having this specification separate from RTL enables re-use of a block in different power profiles..

By employing automation tools that understand the designer's power intent through CPF for optimization, design planning, and layout steps, design teams can achieve superior timing and power tradeoff while improving the productivity and cycle time during the implementation.

Common Power Format

A key enabler of a modern power-aware design flow is the ability to capture and preserve the intent of the chip architects and designers 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?").

The current state-of-the-art with regard to such a specification is the Common Power Format, which is managed by the Silicon Integration Initiative (Si2) consortium's Low Power Coalition. First released in 2007, CPF enables design teams to capture the intent of a design from a power perspective. It provides a mechanism to capture architects' and designers' concepts and constraints for power management, and it enables the automation of advanced low-power design techniques. CPF allows all design, implementation, verification, and technology-related power objectives to be captured in a single file and then applies that data across the design flow, thereby providing a consistent reference point for design verification and implementation.

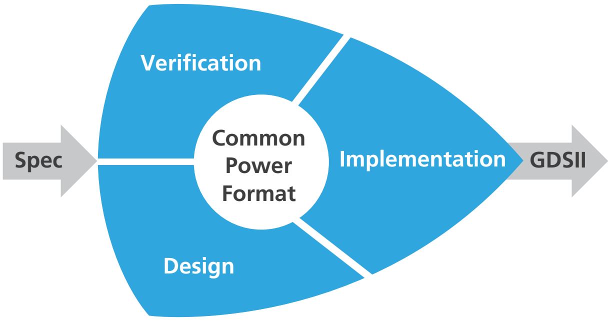

Figure 4: CPF drives design, verification, and implementation.

There are three major benefits of using CPF to drive the design, verification, and implementation steps of the development flow:

- Achieve the required chip specs by driving the implementation to match the required design architecture
- Integrate and automate the design flow, which increases designer productivity and improves the cycle time
- Eliminate the need for manual intervention and replaces ad hoc verification methodologies, thereby reducing the risk of silicon failure due to inadequate functional or structural verification

Chip Planning—Getting Low-Power Right

Low-power design starts with the architectural specification of the chip/system. This is where the most significant tradeoffs can be made. One critical task is to partition the system into its hardware and software components. Hardware implementations are fast and consume relatively little power, but they are "frozen in silicon" and cannot be easily modified to address changes in algorithms, standards, or protocols. By comparison, software implementations are slow and consume a relatively large amount of power, but they are extremely versatile and can be modified long after the chip has gone into production.

Some design techniques have an impact on the architecture of the design and need to be considered early. Some examples of this are DVFS and PSO. Evaluations should be made at the architectural stage as to which blocks are not performance-critical, so that they could potentially be run at a lower voltage and/or frequency to conserve power. In some designs, certain blocks may be suitable candidates to utilize voltage-frequency scaling techniques, in which case it is necessary to determine (and document) how the performance-voltage-frequency feedback mechanism will function. Similarly, in some designs, certain blocks may be suitable candidates for "sleep mode" or to be completely shut down to conserve power when they are inactive. This means that the architects will define different "modes" and then specify which blocks will be on or off (or asleep) in each mode. In some cases, blocks may have different power/performance requirements associated with different modes.

Another key consideration is IP selection, which relates to both internal and third-party IP. In the case of one real-world design that exceeded its power budget, an unexpected amount of leakage was tracked down to the memory IP obtained from a third party. Thus, in addition to its core functionality, an IP block should be evaluated in the context of the device architecture's low-power requirements. Is it required for this block to be capable of being placed in sleep mode and/or completely shut down, for example? And if so, does this IP block support the required functionality?

The real problem was that, until recently, there was no way for system architects and designers to accurately evaluate different design and implementation scenarios and perform "what if" analysis at the architectural level. This led some teams to "over specify" the design at the system-level and to over-constrain (guard-band) the design to be "safe." Unfortunately, this approach can lead to unnecessary increases in costs, schedule delays, and time-to-market (and time-to-profit) delays. Even worse, it leaves unrealized performance "on-the-table," which is unacceptable in today's extremely competitive marketplace.

Alternatively, in the case of teams that decide to use low-power design techniques without access to appropriate planning and analysis tools, there is a tendency to underestimate the overall effort involved with accurately designing and verifying low-power intent, which can lead to device failures and—in the worst case—product recalls.

The end result is that some design teams are too conservative while others are too aggressive—but both approaches are wrong. To address these issues, project teams must have the ability to make educated power trade-offs early in the development process. Cadence InCyte Chip Estimator solves this problem by enabling accurate estimation of a chip's size, dynamic and leakage power consumption, performance achievability, and cost. In addition to these fundamental capabilities, it also provides integrated low-power planning capabilities that allow users to define different power domains and to assign the various functional blocks forming the design to these domains.

For each block, the IP provider or the user can represent the power consumption of that block by defining specific power values associated with performing certain functions or by using simple models, such as "power-per-megahertz." These models can also be defined in terms of the voltage associated with a block and any substrate biasing being applied to that block. The design team can then associate specific parameters—such as clock frequencies with the various functional blocks—and receive accurate power estimations for the alternative architectural scenarios.

This technology also helps the user to define different functional modes and to explore the cost benefits associated with using advanced low-power techniques such as power shutoff and multiple supply voltages. By facilitating rapid exploration of different "what-if" power scenarios, design teams have a much better awareness of what they are up against regarding the various power trade-offs. This allows the teams to better quantify overall "costs" and to address return on investment (ROI) early in the process when it matters most.

Of particular interest is that, once the SoC's high-level power architecture has been defined, it can be exported as a CPF file, which drives the desired power intent into downstream tools and solutions. In some cases, it may be necessary to generate multiple CPF files where each describes a particular power strategy. For example, if we assume a decision has already been made that a chip will have two voltage domains and that the design includes a 3D graphics accelerator, the user may say, "Show me the estimated timing, area, and power associated with having this function in both domains," and then generate two CPF files for further, more-detailed evaluation.

Power Analysis

Power analysis, like timing analysis, needs to be consistent and convergent throughout a design flow. Early power analysis that is not implementation-aware will not have any insight into what types of timing, area, and power optimizations that will be required to meet the design's constraints. This amounts to no better than a guess. Similarly, even implementation-aware power analysis can be widely inaccurate, unless it uses switching activity vectors that closely relate to real system modes. Unrealistic early power estimation can have serious negative effects, such as causing cost overruns due to a more expensive package, pursuing the wrong architecture or optimization strategy, or schedule overruns.

Throughout the implementation flow, power analysis needs to accurately calculate and report all components of power consumption, including active power and leakage power. Power analysis also needs comprehensive reporting that enables designers to understand where power is consumed and how it could be minimized. A good design flow should use consistent power calculation that uses the best available implementation insight from early power estimates through signoff power calculation, so that power is constantly measured, refined, and properly acted upon.

Top-Down Multi-Objective Optimization

One key requirement for low-power design is the ability of logic and physical synthesis engines to concurrently optimize for timing, area, and power tradeoffs. For example, by replacing nominal-Vt cells with high-Vt cells, optimization engines can reduce power at the expense of performance. However, power should not be an afterthought to the optimization process where first the timing targets are achieved and later on cell swapping is done to reduce power. This means that—beginning with the logic synthesis stage and all the way to routing—the optimization engines need to create structures that can meet performance targets while minimizing power consumption.

Similarly, in the case of MSV designs, it is important that the optimization engines understand power domains top-down so that they have full visibility to optimize across entire timing paths. Using a top-down optimization approach for MSV designs leads to superior timing, area, and power tradeoffs. Furthermore, the ability to explore power-timing tradeoffs at the top-most level eliminates the need for multiple iterations that are common in bottom-up partitioning. MSV requires additional logic, such as level shifters, to allow transitions between different voltage domains. This logic adds delay, load, and other factors that need to be considered during implementation and verification.

Low-Power Verification Starts Here

There are different types of verification. Two different verification objectives that most folks think about are: “Does this design do what I wanted it to?” and “Does this design do what I expected it to?” However, it’s also important to answer the question: “Is this design implemented the way I thought it was?” A mix of simulation-based and static formal verification techniques is generally used on low-power designs to verify the design against these objectives. Simulation requires a testbench that can drive the design in all of the representative (and maybe the illegal) power modes necessary, and measures that the response matches expectation. Static formal tools prove the correctness of the design under test, either against a reference design or against a set of design rules. They have the advantage that no testbench is required.

In low-power designs, the amount of verification that should be carried out increases exponentially with the complexity of the power architecture—with the number of power domains in the design and the number of combinations of their states. It is essential to use a simulator that is fully power-aware for elaboration, execution, and debug.

The Cadence Incisive® Enterprise Simulator models low-power component behavior, such as state-retention registers and isolators, from the power intent file, unless such components are already instantiated in the design under test. The simulator correctly corrupts functional logic in the power-off state. Full power-aware debug tools allow you to verify that domains in the off-state do not corrupt those in the on-state, and that domains return to the on-state correctly upon restoration of power. Assertions are generated to automate the creation of power-aware testbenches, and a low-power plan is generated from the power intent file to enable metric-driven closed-loop verification. Close integration with the Cadence Virtuoso® AMS Designer Simulator provides power-aware mixed-signal verification. Low-power functional verification metrics are collected and reported, and an easy-to-use graphical verification environment allows for quick debugging of issues related to errors in the low-power design intent.

The Cadence Encounter® Conformal® product family ensures a design “conforms”—i.e. that the design is implemented the way the designers think it is. For example, Conformal Constraint Designer automates the generation, validation, and refinement of constraints to ensure that timing constraints are valid throughout the entire design process, thereby helping designers achieve rapid timing closure. Similarly, Conformal Equivalence Checker helps designers verify and debug multimillion-gate designs early in the design cycle without using test vectors.

In the case of low-power designs, Conformal Low Power combines equivalence checking and implementation checking of the low-power logic, using formal techniques to enable full-chip verification of designs optimized for low power. Low-power project design teams typically apply Conformal Low Power at multiple check points throughout the design flow. For example, Conformal Low Power can be run first to check that everything in the CPF file is valid before RTL simulation, then pre-and post-synthesis, and at every iteration of the physical netlist.

The Cadence Palladium® system greatly speeds functional verification, making it practical to analyze software in a system-level environment. Additionally, the Palladium system offers a Dynamic Power Analysis (DPA) capability. The solution’s ability to run various design or implementation scenarios—and determine their impact on power dissipation under a realistic application environment—is vital to striking a balance between power budget and expected performance. It delivers power calculation by capturing the necessary power activities in a common DPA power database. This capability further enables the sharing of verification resources while DPA is computing the power profile offline.

Power-Aware Test Considerations

In this context, “test” refers to actually verifying the physical chip on a device tester. In addition to internal scan chains and boundary scan logic, SoCs often include special built-in self-test (BIST) structures, including logic BIST (LBIST) and memory BIST (MBIST). The problem is that these functions aren’t used in the normal operation of the chip, but they add to the device’s power consumption while in test mode.

Similarly, in order to reduce memory buffer size on the device tester and to speed the test sequence, it is now common to take the output vector sequence generated by automatic test pattern generation (ATPG) and to compress it. The compressed test-patterns are then fed into the chip, where special logic is used to decompress them. Once again, these functions aren’t used in the normal operation of the chip, but they add to the device’s power consumption while in test mode.

The end result is that, if these effects are not fully accounted for, good devices that would work in the field may fail on the tester. It is for this reason that design for test (DFT) has to be introduced early in the design cycle. By incorporating DFT structures into the RTL pre-synthesis, the design team is provided with a more complete picture of the chip as early as possible, thereby allowing them to fully understand the area, performance, and power implications associated with any test circuitry. However, the real issue here is that during test it is possible to get much higher levels of activity than is possible during normal device operation. Thus, any simulations that have been done to measure peak power will be irrelevant. Since adding the headroom for test would “bust” the power budget, the test vector generation needs to be fully power-aware, thereby minimizing the activity that is generated by the vectors without impacting coverage.

This leads to the requirement that the entire test environment be power-domain aware. For example, when testing across different power structures, the system should not attempt to hook up two scan chains in different voltage domains without including an appropriate level shifter. Similarly, the system should not attempt to shift test data through a block that’s been powered down. Last but not least, the test strategy must actively test any special power structures, including isolation logic, power switches, state-retention cells, and so forth.

Moving into the Physical Design Phase

At this stage, the design moves into the Cadence Encounter Digital Implementation System which delivers a complete solution for variation- and manufacturing-aware design closure, low power, mixed-signal implementation, and integrated signoff in a single, scalable, multi-CPU-enabled design environment.

The Encounter Digital Implementation System combines RTL synthesis, silicon virtual prototyping, automated floorplan synthesis, clock network synthesis, design for manufacturability and yield, low-power, mixed-signal design support, and nanometer routing. It also offers the latest capabilities to support advanced designs at the 40nm, 32/28nm, and 22/20nm technology nodes.

One aspect of the development process that is very important to understand is that not everything proceeds in a nice top-down flow—this is just a fact of life. For example, the design team may have performed some detailed power grid planning for critical blocks, but they will typically want to be far along in the process before they start investing significant amounts of time and effort in creating a detailed power grid for the entire chip.

Based on the CPF file, the Encounter Digital Implementation System can be used to automatically create power domains and to layout the power grid very quickly. Once again, Conformal Low Power will be used to check the gate-level netlists against actual real-world power and ground connections, and also to verify that any isolation logic is associated with appropriate power domains and has correct connections in real world. Where the design team is using a flow comprised completely of Cadence products, part of this process will be done in the RTL synthesis phase. Additionally, when bringing in IP that does not have the necessary power structures, the Encounter Digital Implementation System can be used to add them.

Similarly in the case of the clock tree (or trees), the design team ideally will want a “pretty good “netlist and “pretty good” floorplan before they invest a lot of time and effort in a detailed clock tree implementation. Previously, clock tree synthesis (CTS) has been separate from physical optimization. While this approach has generally worked down to the 40nm node, it is now breaking down because on-chip variation, low-power demands, and complexity are causing a significant “timing gap” between the ideal clocks used pre-CTS and the propagated clocks used post-CTS.

Cadence Clock Concurrent Optimization (CCOpt) is a new approach to timing optimization that holistically addresses this timing gap. CCOpt is the only tool that optimizes the clock insertion as well as logic delays simultaneously, instead of doing them separately. In addition to CTS, it encompasses timing-driven placement, incremental physical optimization, physical clock gating, and post-clock tree optimization. CCOpt performs concurrent useful-skew and datapath optimization, as well as time borrowing for faster timing closure, leading to significant productivity improvements. Observed results have included clock-tree power reductions of 30%, clock-tree area reductions of 30%, and chip performance improvements up to 100MHz for a GHz+ design. Additional benefits include up to 30% reduction in IR drop (because registers and RAMs are triggered at different times) and a skew profile that reduces peak current by up to 40% without any impact on timing.

Low-Power Formal Verification

Low-power designs may fail due to a number of structural errors, such as missing level-shifters/isolation logic, redundant level-shifters/isolation cells, bad power-switch connections, and bad power and ground connections. Some possible functional errors include bad state-retention sleep/wake sequences and bad logic for power gating and isolation. Thus, it is important that designers use robust formal verification techniques to identify and rectify these structural and functional errors.

Physical Implementation of Low-Power Structures

Multiple power domains in MSV and PSO techniques require the insertion, placement, and connection of specialized power structures, such as level shifters, power pads, switch cells, isolation cells, and state-retention cells.

In particular, PSO requires a special (and more complex) set of considerations for implementation and analysis. For example, one consideration on the implementation side is that designers need to make a tradeoff between using fine- and coarse-grained power gating. Fine-grained gating includes a power-gating transistor in every standard cell, which has a large area penalty but eases implementation. Coarse-grained gating reduces the area penalty by using a single gate (many designs use multi-stage switching to control the inrush current) to cut off power to an entire block, but requires sophisticated analysis to determine the gating current of the switched-off block.

On the analysis side, designers must consider the impact of powering up several blocks at the same time, which could lead to IR drop in the adjacent circuitry caused by the large rush currents associated with power-up. This requires dynamic power analysis to better understand the power-up characteristics with a goal of reducing rush currents and minimizing IR drop impact.

Clock-Gating Challenges

Though clock gating has been used for some time as an effective technique to reduce dynamic power, today’s stringent power specs demand ever more sophisticated gating techniques. The most sophisticated form of clock gating currently available is multi-stage gating, in which a common enable is split into multiple sub-enables that are active at different times and/or under different operating modes. And, of course, the process is further complicated when it comes to clock trees that pass through different power domains, especially when one or more of those power domains are candidates for power shutoff.

Additionally, since the clock network is one of the most power-hungry nets on a chip, it is critical to design with power dissipation in mind. Clock tree synthesis develops the interconnect that connects the system clock into all the cells in the chip that uses the clock. For successful low-power designs, clock tree synthesis also needs to be power-aware.

Power Grid Implementation and Analysis

Power domains must be shaped and placed; power pads and switches must be placed and optimized; and power routing must be planned. Silicon virtual prototyping helps the partitioning process by minimizing the wire lengths of any high switching probability wires, which lowers the dynamic power. This requires tools that can understand which wires contribute the most capacitance and that are able to find ways to minimize the interconnect capacitance through optimal partitioning and floorplanning.

The performance of a design's power networks, or grid, has a direct impact on its performance. Voltage (IR) drops on VDD nets and ground bounce on VSS nets affect a design's overall timing and functionality and—if ignored—can cause silicon failure. High currents in the power grids also induce electromigration (EM) effects, causing the power routing to wear out (i.e. become more resistive) during a chip's lifetime.

A complete picture of power grid robustness can only be obtained when effects such as IR drop, ground bounce, and EM are accurately computed and analyzed. Timing, signal integrity, and power analysis should be performed with the effects of IR drop and ground bounce included once the power grid is routed. These are full-chip issues that must be addressed by verification tools that have the capacity and performance required to analyze detailed representations of the entire chip in a reasonable amount of time.

Power Rail Analysis

Power rail analysis should be performed early in the design flow to ensure robust power grids from time zero. Power switches and de-coupling capacitance must be optimized using power rail analysis to ensure that margin-driven over-design is limited. As the design progresses, power rail analysis should be used to continuously validate that the on-chip power delivery is still within specification as the design becomes complete.

Power rail signoff verification demands a hierarchical, full-chip solution that delivers the capacity and performance required to analyze detailed representations of the entire chip in a reasonable amount of time. Timing, signal integrity, and power analysis should be performed with the effects of IR drop and ground bounce included once the power grid is routed. By analyzing and verifying power grids, timing, and signal integrity at the full-chip level, designs can be taped-out with high level of confidence in achieving first-pass silicon success.

Signal Integrity Considerations

With increasing clock frequency and the lowering of supply voltages, there is increasing sensitivity to signal integrity (SI) effects such as crosstalk-induced delay changes and functional failures. Furthermore, the use of advanced power management techniques makes SI analysis even more complicated. In the case of a PSO design, for example, a spurious signal caused by an SI aggressor could shut down an entire module. Similarly, the use of multiple power domains can lead to the creation of super-aggressors, where victim nets in a low-voltage domain can be weaker and aggressor nets in a high-voltage domain can be stronger. Overall, multi-Vt, MSV, and PSO techniques require implementation and analysis tools to be power-aware so they can account for these SI effects.

Proceeding to Signoff

A "big fat" power grid may supply all the power the chip needs, but it wastes die area, costs money, and involves extra capacitance that has to be charged. By comparison, an under-designed power grid can result in reliability issues, where—in a worst-case scenario—the chip tests-out fine but fails in the field. The end result is that designing an optimal power grid with the right amount of margin to cover variations in the manufacturing process involves a complex, multi-dimensional tradeoff.

Of course, "beauty is in the eye of the beholder." What do we mean by "optimal?" Well, in the context of power—and from the engineering perspective—"optimal" means designing a power grid that gets the job done without wasting power in the smallest number of design iterations. This involves a similar "successive refinement" approach as is used for timing, in which the design team starts with fast estimations, hones in on problem areas, and then refines the design in an environment of continuous convergence.

In the case of power, the team starts by saying: "How many power and ground pins are we going to need to supply this chip?" The power architecture and underlying power grid is then continuously refined throughout the implementation process.

This avoids the hidden costs of over-design, where everything is "padded out just in case," but the designers never come back to revisit it. When working in an environment that supports continuous convergence, any excess "padding" is eliminated as part of the process.

All of this is embodied by the Encounter Digital Implementation System, which brings power and timing together to locate, isolate, and resolve issues, and to avoid over-design and under-design. Earlier in the process, the design team used emulation of the RTL to focus in on "hot-spots" (by which we mean areas of high activity that stress the power architecture and power grid). These hot-spots were used as starting points for more in-depth evaluation in the software simulation domain.

Summary

In recent years, power consumption has moved to the forefront of digital IC 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 a dramatic rate.

Design teams across the semiconductor industry are adopting power management techniques to meet their market requirements. In addition to the usual considerations of chip performance, functionality, and cost, these new power-related requirements include improved battery life, reduced system cost, cooler operation and improved reliability.

Although advanced low-power techniques such as MSV, PSO, and DVFS offer maximum power reduction, these techniques may have profound impact on the chip design, verification, and implementation methodologies. Often, the complexities associated with these advanced techniques result in low-power designs that suffer from sub-optimal timing, power, and area tradeoffs; reduced productivity and turnaround time; and increased risk of silicon failure.

To enable the adoption of advanced low-power techniques by mainstream users, there is a need for a design flow that holistically addresses the architecture, design, verification, and implementation of low-power designs. The Common Power Format (CPF) has emerged as an effective means to capture the design's power intent early on and to communicate this intent throughout the design flow. CPF has now been used on more than 500 successful advanced low-power designs since its release in 2007.

Cadence has a comprehensive and proven solution for low-power design throughout the entire SoC development process. Starting at the very beginning of the architectural phase of the design, the Cadence InCyte Chip Estimator enables accurate estimation of a chip's size, dynamic and leakage power consumption, performance achievability, and cost. It provides an architectural exploration environment where users can quantify and compare a vast number of chip implementation options (including different technology nodes, process libraries, and IP offerings) to balance technical and economic goals.

To accurately estimate power at both the RTL and gate netlist levels, the design team can use state-of-the-art Dynamic Power Analysis (DPA) that works within hardware emulators such as Cadence Palladium systems. This type of system offers tremendous throughput and is cycle accurate, thereby allowing users to dynamically capture and analyze power switching activities for peak and average SoC power consumption, including the effects of embedded software and other real-world stimuli.

The Cadence Encounter Digital Implementation System combines RTL synthesis, silicon virtual prototyping, automated floorplan synthesis, clock network synthesis, design for manufacturability and yield, low-power and mixed-signal design support, and nanometer routing. It also offers the latest capabilities to support advanced designs at the 40nm, 32/28nm, and 22/20nm technology nodes. Meanwhile, Cadence Incisive verification technologies verify low-power intent with no disruption or modification to the functional verification environment. These technologies include Incisive Enterprise Manager, Incisive Enterprise Simulator, and Incisive Formal Verifier, all of which integrate with Encounter Conformal Low Power.

In summary, the design, implementation, and verification tools and flows provided by Cadence address all areas of power management and are design-proven to solve the SoC low-power problem comprehensively.