



# A Practical Guide to Low-Power Design

## User Experience with CPF



power  forward<sup>TM</sup>

## **Foreword**

*Energy consumption is a major, if not the major, concern today. The world is facing phenomenal growth of demand for energy from the Far East coupled with the unabated and substantial appetite for energy in the US and Europe. At the same time, population growth, economic expansion and urban development will create greater demand for more personal-mobility items, appliances, devices and services. Recognizing these worrisome trends, the U.S. Department Of Energy (DOE) has identified the reduction of energy consumption in commercial and residential buildings as a strategic goal. The Energy Information Administration at DOE attributed 33% of the primary energy consumption in the United States to building space heating and cooling—an amount equivalent to 2.1 billion barrels of oil. At these levels, even a modest aggregate increase in heating ventilation and air conditioning (HVAC) efficiency of 1% will provide direct economic benefits to people, enabling reduction and better management of electric utility grid demand, and reducing dependence on fossil fuels. In addition to the global relevance of efficient energy usage, there are the micro-economic and convenience concerns of families, where energy consumption is putting pressure on domestic budgets and where battery life of home mobile appliances is becoming a major selection factor for consumers.*

*What can electronics makers do to help? Energy usage can be optimized at the chip, board, box, system, and network level. At each of these levels there are major gains that can be achieved. Low-power design has been a substantial research theme for years in IC design. Several important results have been used to limit energy consumption by fast components such as microprocessors and digital signal processors. However, while the trend has been improving, the energy consumption of, for example, Intel and AMD microprocessors is still very important, so that additional research is warranted. As we traverse layers of abstractions towards systems and networks, the attention paid to low energy consumption is not increasing proportionally; an important issue to consider moving forward on the energy conservation path.*

*Companies should take a holistic view in the energy debate. By carefully managing the interactions between the different layers of abstraction and by performing a global trade-off analysis, companies may take a leadership position. We understand that at this time, enough attention has not been paid to energy consumption as the design goals have been centered on performance and cost. We also believe that no one company or institution acting alone can tackle all the issues involved. Leveraging the supply chain, EDA companies, partners' research organizations and Universities offers a way to corral the available resources and focus on the problem.*

*Focusing on the IC design area, process engineers cannot solve the problem alone: 90nm and smaller process nodes are burning more power with increased design complexity and clock frequencies. Static power is becoming the predominant source of energy waste. It is up to the design, EDA and IP community to create methodologies that support better designs, higher performance, lower costs and higher engineering productivity, in the context of low power.*

*I applaud the efforts of Cadence and the Power Forward Initiative members to develop, in a very a short period of time, a methodology that uses the Common Power Format. Partners and competitors alike worked closely across the entire design and manufacturing ecosystem, from advanced designers of low-power SoCs, to EDA vendors, to foundries, to IP vendors, to ASIC vendors, to design service companies. They all recognized the serious needs and formulated a working solution.*

*I believe that this guide will be a fundamental reference for designers and will help the world in saving a substantial amount of energy!*

***Dr. Alberto Sangiovanni-Vincentelli, Professor, The Edgar L. and Harold H. Buttner Chair of Electrical Engineering, University Of California, Berkeley, Co-founder, CTA and Member of the Board, Cadence Design Systems.***

## **Preface**

*In 2005, it was clear that power had become the most critical issue facing designers of electronic products. Advanced process technology was in place, power reduction techniques were known and in use, but design automation and its infrastructure lagged. Low-power design flows were manual, error-prone, risky, and expensive. The pressure to reduce power was ever more pervasive and the methodologies available were undesirable.*

*Recognizing this burgeoning design automation and infrastructure problem, Cadence as the EDA leader took the initiative to tackle this crisis. To solve the broader design problem holistically, the effort had to involve the entire electronic product development design chain, including systems and EDA companies, IP suppliers, foundries, ASIC and design services companies as well as test companies. In May of 2006, we teamed up with 9 other industry leaders to form the Power Forward Initiative (PFI) to address the obstacles to lower-power IC design. Within Cadence, technologists from over 15 business groups realized that to incorporating an efficient, automated low-power design solution into existing design flows would require, significant innovation in every step of the design flow. Through intensive collaboration across the team, it was concluded that implementing advanced power reduction techniques could be best facilitated by a separate, comprehensive definition of power intent that could be applied at each step in the design, verification and implementation stages. The Common Power Format was born.*

*The founding members of PFI: Applied Materials, AMD, ARM, ATI, Freescale, Fujitsu, NEC Electronics, NXP, and TSMC came together with Cadence to devise, refine and validate the holistic, CPF-enabled design, verification and implementation methodology. From the very outset, the goal was to quickly enable the rapid deployment of a design automation solution that comprehends power at every stage of the design process. The scope of the R&D effort was huge, spanning software and algorithmic technology innovation, solution kits, methodology development, and challenging software validation problems. The vision was simple but success depended on execution at a scope never attempted before in the history of EDA.*

*Starting in 2006, the founding companies of PFI created and reviewed the CPF specification. They then initiated proof point projects that validated design flows using the Cadence® Low Power Solution with complex designs and power intent specified in CPF. By the fall of 2006, PFI members completed validation of a robust methodology and CPF specification and it was ready for broad deployment and standardization. The CPF specification was publicly contributed to the Si2 Low Power Coalition (LPC) in December 2006. In March 2007, it became a Si2 standard, open and freely available to everyone in the industry. Since then, the Si2 LPC has continued to investigate new opportunities for CPF and plot out the evolution of this holistic low-power format. With a growing movement towards developing greener electronic products, interest in PFI, the Si2 LPC, and the adoption of CPF-enabled methodology continues to expand rapidly. A uniform vision and belief in the energy efficient electronic products drove the industry-wide team at an accelerated pace.*

*The result, A Practical Guide to Low Power Design, embodies the collective intellectual work and experience of some of the best engineers in the electronics industry. Our goal in developing this living, web-based book is to share our experience with the world's design community. As new designs are completed, new chapters in low-power design will be written and added to the guide.*

*Finally, I want to acknowledge all the people involved in this effort. This diverse pan-industry team of dedicated individuals worked with passion and commitment to bring this solution to life. Working on a noble cause that has positive and measurable impact on the state of the art in electronic design as well as positive ramifications for the environment has been exciting for us all. Together, we have built an ecosystem to accelerate low-power design.*

**Dr. Chi-Ping Hsu, Corporate Vice President, IC Digital and Power Forward, Cadence Design Systems.**

## Acknowledgements

*This book has been made possible by the personal passion and commitment of scores of dedicated people. We would like to offer our thanks and gratitude to these individuals from companies in the Power Forward Initiative for the countless hours spent in reviewing the CPF specification, providing feedback, and engaging in complex proof point projects to validate CPF-enabled design flows. We will attempt to acknowledge many of them here, but for each mentioned, there are numerous others on their teams who worked diligently to make CPF, CPF-based low-power design methodology and ultimately this book a reality.*

*Special thanks go to Toshiyuki Saito from NEC Electronics as his vision inspired this book project. He articulated the need to capture the collective Power Forward Initiative experiences in one place to make them available for the benefit of the broader electronics design community.*

*We thank the founding members of the Power Forward Initiative — ATI, AMD, Applied Materials, ARM, Freescale, Fujitsu, NEC, NXP, and TSMC — for having the vision to recognize the challenges of low-power design and the commitment to work on developing a holistic low-power intent specification.*

*Special thanks also go to those companies who engaged in early proof point projects, with a nascent CPF specification, to validate the solution for low power design. We are grateful for the hard work of engineering teams at ARC, AMD, ARM, Freescale, Fujitsu, NEC, NXP, and TSMC for their CPF-based design projects.*

*We express our gratitude to the following individuals and to their companies for contributing the resources to participate in the Power Forward Initiative; that work served as the basis for this book.*

<i>ARC International</i>	<i>Karl Aucker, Gagan Gupta, Colin Holehouse</i>
<i>ARM, Inc.</i>	<i>Keith Clarke, Joe Convey, John Goodenough</i>
<i>AMD Corporation</i>	<i>Ed Chen, Dan Shimizu, Ward Vercruysse, Gill Watt</i>
<i>Cadence Design Systems</i>	<i>Mohit Bhatnagar, Pinhong Chen, Yonghao Chen, John Decker, Phil Giangarra, Mitch Hines, Anand Iyer, Lisa Jensen, Tony Luk, Pankaj Mayor, Michael Munsey, Koorosh Nazifi, Rich Owen, Susan Runowicz-Smith, Saghir Shaikh, Randy Shay, Tim Yu, Qi Wang Tony Willis, William Winkler</i>
<i>Calypto Design</i>	<i>Anmol Mathur, Devadas Varma</i>
<i>Faraday</i>	<i>Lee Chu, C. J. Hsieh, Chung Ho, Albert Chen</i>
<i>Freescale Semiconductor</i>	<i>Arijit Dutta, Dave Gross, Milind Padhye, Joe Pumo</i>
<i>Fujitsu Electronics</i>	<i>Yoshimi Asada, Tsutomu Nakamori</i>
<i>Globetech Solutions</i>	<i>Stylianios Diamantidis</i>
<i>Global Unichip Corporation</i>	<i>Albert Li, Kurt Huang</i>
<i>NEC Electronics</i>	<i>Toshiyuki Saito, Hideyuki Okabe, Toshihiro Ueda, Hiroshi Kikuchi</i>
<i>NXP Semiconductors</i>	<i>Barry Dennington, Herve Menager</i>
<i>Sequence Design</i>	<i>Vic Kulkarni, Tom Miller</i>
<i>Si2</i>	<i>Nick English, Steve Schulz, Sumit Dasgupta</i>
<i>SMIC</i>	<i>Feng Chen</i>
<i>Tensilica</i>	<i>Ashish Dixit, Jagesh Sanghavi</i>
<i>TSMC</i>	<i>Chris Ho, David Lan, L.C. Lu, Ed Wan</i>
<i>Virage Logic</i>	<i>Oscar Siguenza, Manish Bhatia</i>
<i>Improv Systems</i>	<i>Victor Berman</i>
<i>UMC</i>	<i>Garry Shyu</i>

*We gratefully acknowledge Neyaz Khan, who developed the book outline and contributed the introduction and verification chapters, Tim Yu who contributed the front-end design chapter and Wei-Li Tan who contributed the low-power implementation chapter. Special thanks to Holly Stump who was executive editor for the book, her dedication and expertise contributed greatly to the entire project. And last but not least, thanks to Susan Runowicz-Smith who tirelessly managed the entire project.*

## Table of Contents

Introduction to Low Power.....	6
Low Power Today.....	6
Power Management .....	9
Complete Low-Power RTL-to-GDSII Flow Using CPF .....	23
A Holistic Approach to Low-Power Intent .....	31
Verification of Low-Power Intent with CPF .....	32
Power Intent Validation .....	32
Low-Power Verification.....	34
CPF Verification Summary .....	50
Front-End Design with CPF .....	52
Architectural Exploration.....	52
Synthesis Low-Power Optimization .....	54
Automated Power Reduction in Synthesis.....	56
CPF-Powered Reduction in Synthesis .....	62
Simulation for Power Estimation.....	72
CPF Synthesis Summary .....	75
Low-Power Implementation with CPF .....	76
Introduction to Low-Power Implementation .....	76
Gate-Level Optimization in Power-Aware Physical Synthesis .....	80
Clock Gating in Power-Aware Physical Synthesis.....	80
Multi-V <sub>th</sub> Optimization in Power-Aware Physical Synthesis .....	81
Multiple Supply Voltage (MSV) in Power-Aware Physical Synthesis.....	82
Power Shutoff (PSO) in Power-Aware Physical Synthesis .....	85
Dynamic Voltage/Frequency Scaling (DVFS) Implementation .....	93
Substrate Biasing Implementation.....	94
CPF Implementation Summary .....	98
CPF User Experience .....	99
ARC Energy PRO: Technology for Active Power Management .....	99
NEC Electronics: Integrating Power Awareness in SoC Design with CPF .....	107
FUJITSU: CPF in the Low-Power Design Reference Flow.....	126
NXP User Experience: Complex SoC Implementation with CPF.....	143
References and Bibliography .....	162
Low-Power Links.....	164
Power Forward Initiative .....	164
Cadence Low-Power Links.....	164
CPF Terminology Glossary .....	165
Design Objects .....	165
CPF Objects .....	165
Special Library Cells for Power Management.....	166
Index .....	167



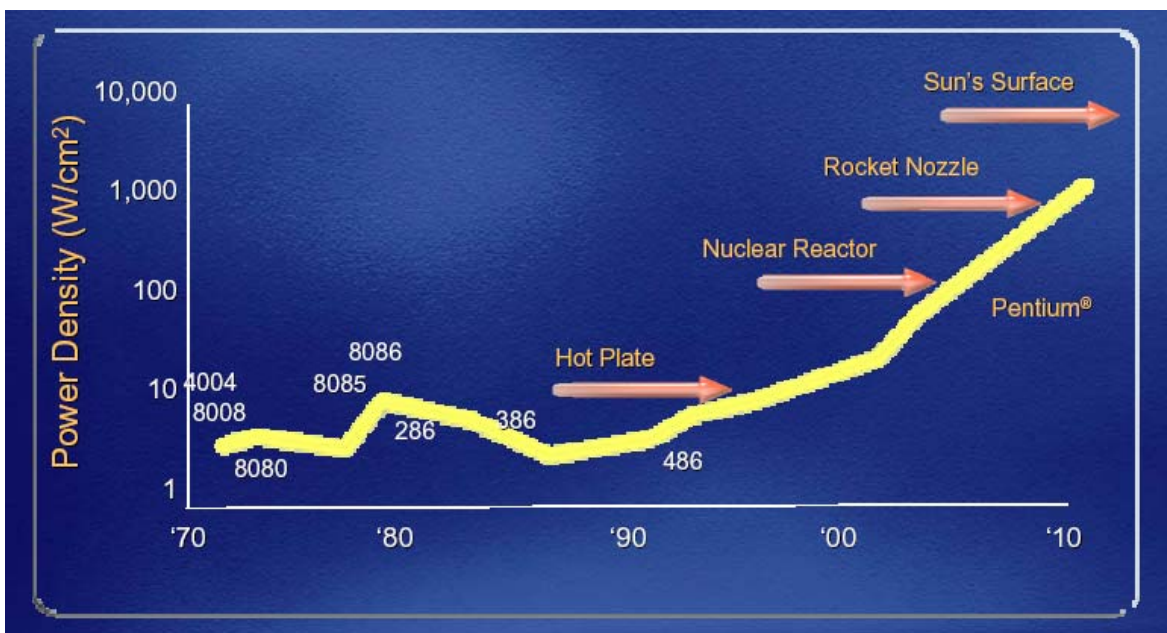
# Introduction to Low Power

## Low Power Today

It's no secret that power is emerging as the most critical issue in system-on-chip (SoC) design today. Power management is becoming an increasingly urgent problem for almost every category of design, as power density—measured in watts per square millimeter—rises at an alarming rate.

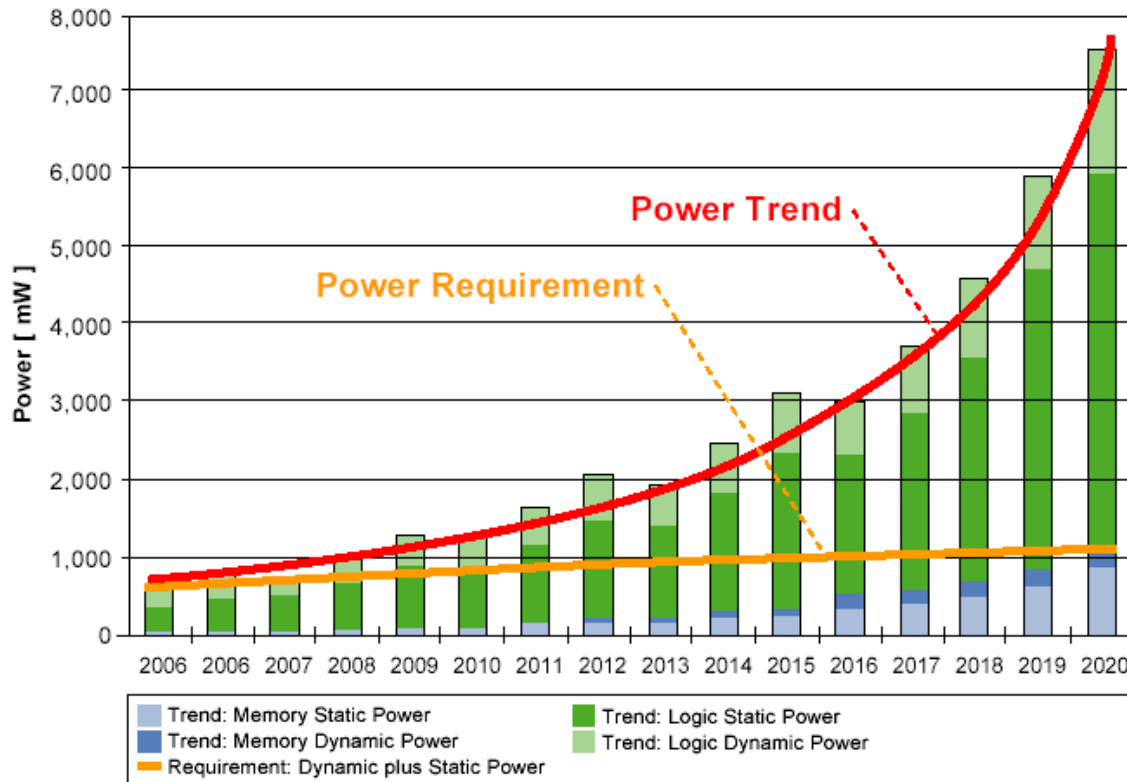
From a chip-engineering perspective, effective energy management for an SoC must be built into the design starting at the architecture stage; and low-power techniques need to be employed at every stage of the design, from RTL to GDSII.

Fred Pollack of Intel first noted a rather alarming trend in his keynote at MICRO-32 in 1999. He made the now well-known observation that power density is increasing at an alarming rate, approaching that of the hottest man-made objects on the planet, and graphed power density as shown in Figure 1.



**Figure 1. Power density with shrinking geometry. Courtesy Intel Corporation [\(Ref. 1\)](#)**

The power density trend versus power design requirements for modern SoCs is mapped in Figure 2. The widening gap represents the most critical challenge that designers of wireless, consumer, portable, and other electronic products face today.

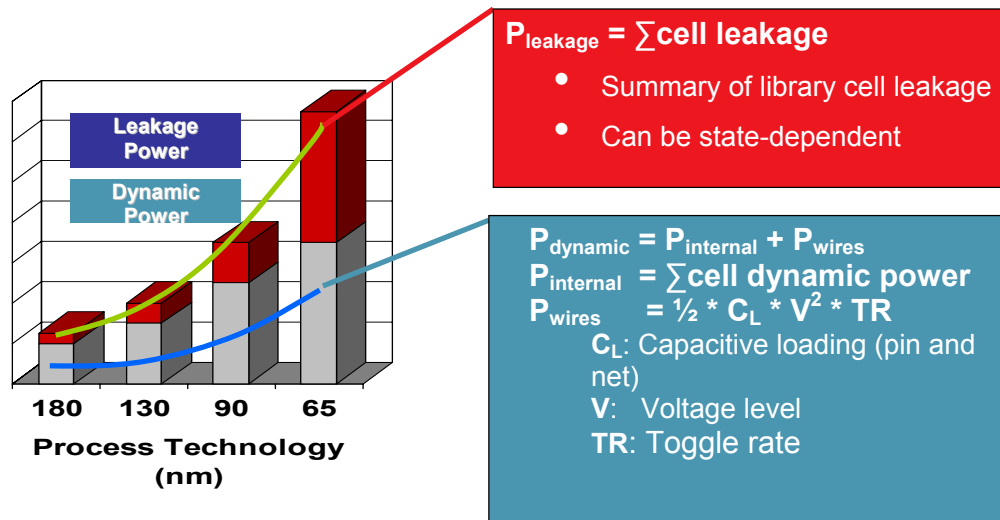


Source: IRTS 2005 Power Consumption Trends for Soc-PE

**Figure 2. IC power trends: actual vs. specified. Courtesy Si2 LPC. (Ref. 2)**

Meanwhile, the design efforts in managing power are rising due to the necessity to design for low power as well as for performance and costs. This has ramifications for engineering productivity, as it impacts schedules and risk.

Power management is a must for all designs of 90nm and below. At smaller geometries, aggressive management of leakage current can greatly impact design and implementation choices. Indeed, for some designs and libraries, leakage current exceeds switching currents, thus becoming the primary source of power dissipation in CMOS, as shown in Figure 3.



**Figure 3. Process technology vs. leakage and dynamic power**

Until recently, designers were primarily concerned with improving the performance of their designs (throughput, latency, frequency), and reducing silicon area to lower manufacturing costs. Now power is replacing performance as the key competitive metric for SoC design.

These power challenges affect almost all SoC designs. With the explosive growth of personal, wireless, and mobile communications, as well as home electronics, comes the demand for high-speed computation and complex functionality for competitive reasons. Today's portable products are expected not only to be small, cool, and lightweight, but also to provide extremely long battery life. And even wired communications systems must pay attention to heat, power density, and low-power requirements. Among the products requiring low-power management are the following:

- Consumer, wireless, and handheld devices: cell phones, personal digital assistants (PDAs), MP3 players, global positioning system (GPS) receivers, and digital cameras
- Home electronics: game consoles for DVD/VCR players, digital media recorders, cable and satellite television set-top boxes, and network and telecom devices
- Tethered electronics such as servers, routers, and other products bound by packaging costs, cooling costs, and Energy Star requirements supporting the Green movement to combat global warming

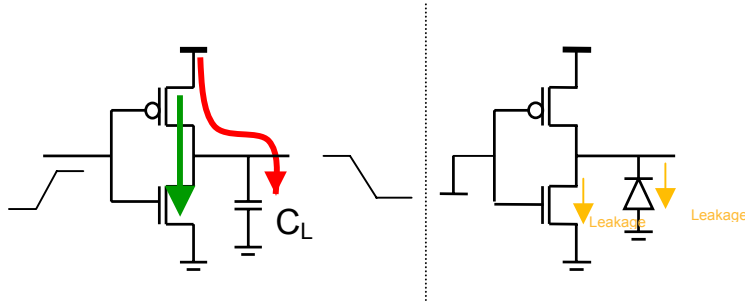
For most designs being developed today, the emphasis on active low-power management—as well as on performance, area, and other concerns—is increasing.



## Power Management

### Power Dissipation in CMOS

Let's take a quick look at the sources of power dissipation. Total power is a function of switching activity, capacitance, voltage, and the transistor structure itself.



**Figure 4. Power dissipation in CMOS**

$$Power = P_{switching} + P_{short-circuit} + P_{leakage}$$

*Total power* is the sum of dynamic and leakage power.

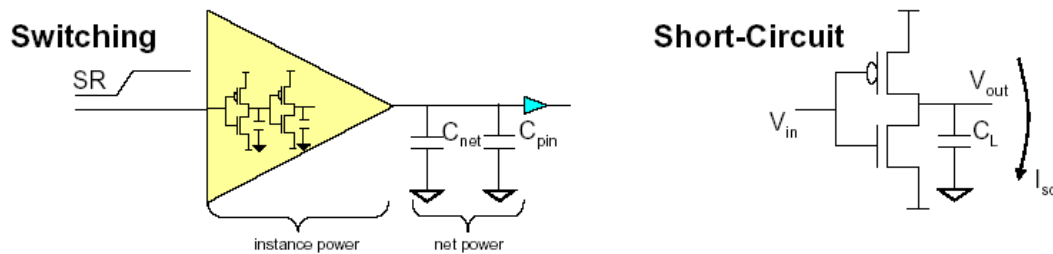
*Dynamic power* is the sum of two factors: switching power plus short-circuit power. Switching power is dissipated when charging or discharging internal and net capacitances. Short-circuit power is the power dissipated by an instantaneous short-circuit connection between the supply voltage and the ground at the time the gate switches state.

$$P_{\text{switching}} = a \cdot f \cdot C_{\text{eff}} \cdot V_{\text{dd}}^2$$

Where  $a$  = switching activity,  $f$  = switching frequency,  $C_{\text{eff}}$  = effective capacitance,  $V_{\text{dd}}$  = supply voltage

$$P_{\text{short-circuit}} = I_{\text{sc}} \cdot V_{\text{dd}} \cdot f$$

Where  $I_{\text{sc}}$  = short-circuit current during switching,  $V_{\text{dd}}$  = supply voltage,  $f$  = switching frequency



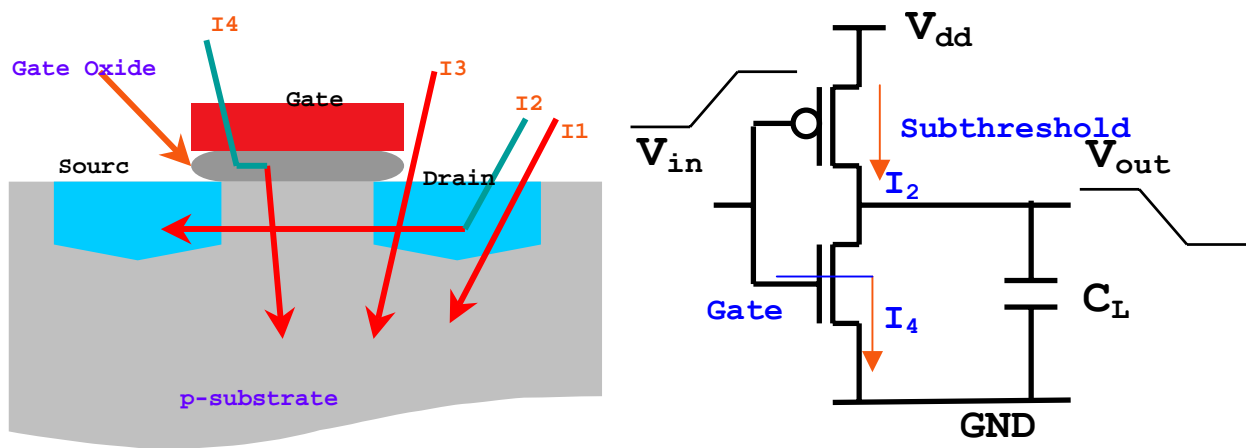
**Figure 5. Dynamic power in CMOS**

Dynamic power can be lowered by reducing switching activity and clock frequency, which affects performance; and also by reducing capacitance and supply voltage. Dynamic power can also be reduced by cell selection—faster slew cells consume less dynamic power.

Leakage power is a function of the supply voltage  $V_{\text{dd}}$ , the switching threshold voltage  $V_{\text{th}}$ , and the transistor size.

$$P_{\text{Leakage}} = f(V_{\text{dd}}, V_{\text{th}}, W/L)$$

Where  $V_{\text{dd}}$  = supply voltage,  $V_{\text{th}}$  = threshold voltage,  $W$  = transistor width,  $L$  = transistor length



**Figure 6. Leakage power in CMOS**

Of the following leakage components, sub-threshold leakage is dominant.

- I1: Diode reverse bias current
- I2: Sub-threshold current
- I3: Gate-induced drain leakage
- I4: Gate oxide leakage

While dynamic power is dissipated only when switching, leakage power due to leakage current is continuous, and must be dealt with using design techniques.

### **Techniques for Switching and Leakage Power Reduction**

The following table defines some common power management techniques for reducing power:

Power Management Technique	Definition
<b><i>Clock tree optimization and clock gating</i></b>	<i>Portions of the clock tree(s) that aren't being used at any particular time are disabled.</i>
<b><i>Operand isolation</i></b>	<i>Reduce power dissipation in datapath blocks controlled by an enable signal; when the datapath element is not active, prevent it from switching.</i>
<b><i>Logic restructuring</i></b>	<i>Move high switching operations up in the logic cone, and low switching operations back in the logic cone; a gate-level dynamic power optimization technique.</i>

<b>Logic resizing (transistor resizing)</b>	<i>Upsizing improves slew times, reducing dynamic current. Downsizing reduces leakage current. To be effective, sizing operations must include accurate switching information.</i>
<b>Transition rate buffering</b>	<i>Buffer manipulation reduces dynamic power by minimizing switching times.</i>
<b>Pin swapping</b>	<i>By swapping gate pins, switching occurs at gates/pins with lower capacitive loads.</i>
<b>Multi-<math>V_{th}</math></b>	<i>With the use of multi-threshold libraries, individual logic gates use transistors with low switching thresholds (faster with higher leakage) or high switching thresholds (slower with lower leakage).</i>
<b>Multi-supply voltage (MSV or voltage islands)</b>	<i>Selected functional blocks are run at different supply voltages.</i>
<b>Dynamic voltage scaling (DVS)</b>	<i>In this subset of DVFS, selected portions of the device are dynamically set to run at different voltages on the fly while the chip is running.</i>
<b>Dynamic voltage and frequency scaling (DVFS)</b>	<i>Selected portions of the device are dynamically set to run at different voltages and frequencies on the fly while the chip is running. Used for dynamic power reduction.</i>
<b>Adaptive voltage and frequency scaling (AVFS)</b>	<i>In this variation of DVFS, a wider variety of voltages are set dynamically, based on adaptive feedback from a control loop; involves analog circuitry.</i>
<b>Power shutoff (PSO), or power gating</b>	<i>When not in use, selected functional blocks are individually powered down.</i>
<b>Memory splitting</b>	<i>If the software and/or data are persistent in one portion of a memory but not in another, it may be appropriate to split that block of memory into two or more portions. One can then selectively power down those portions that aren't in use.</i>
<b>Substrate biasing (body-biasing or back-biasing)</b>	<i>Substrate biasing in PMOS biases the body of the transistor to a voltage higher than <math>V_{dd}</math>; in NMOS, to a voltage lower than <math>V_{ss}</math>.</i>

## Clock tree optimization and clock gating

In normal operation, the clock signal continues to toggle at every clock cycle, whether or not its registers are changing. Clock trees are a large source of dynamic power because they switch at the maximum rate and typically have larger

capacitive loads. If data is loaded into registers only infrequently, a significant amount of power is wasted. By shutting off blocks that are not required to be active, clock gating ensures power is not dissipated during the idle time.

Clock gating can occur at the leaf level (at the register) or higher up in the clock tree. When clock gating is done at the block level, the entire clock tree for the block can be disabled. The resulting reduction in clock network switching becomes extremely valuable in reducing dynamic power.

## Operand Isolation

Often, datapath computation elements are sampled only periodically. This sampling is controlled by an enable signal. When the enable is inactive, the datapath inputs can be forced to a constant value. The result is that the datapath will not switch, saving dynamic power.

## Multi- $V_{th}$

Multi- $V_{th}$  optimization utilizes gates with different thresholds to optimize for power, timing, and area constraints. Most library vendors provide libraries that have cells with different switching thresholds. A good synthesis tool for low-power applications is able to mix available multi-threshold library cells to meet speed and area constraints with the lowest power dissipation. This complex task optimizes for multiple variables and so is automated in today's synthesis tools.

## MSV

Multi-supply voltage techniques operate different blocks at different voltages. Running at a lower voltage reduces power consumption, but at the expense of speed. Designers use different supply voltages for different parts of the chip based on their performance requirements. MSV implementation is key to reducing power since lowering the voltage has a squared effect on active power consumption. MSV techniques require level shifters on signals that go from one voltage level to another. Without level shifters, signals that cross voltage levels will not be sampled correctly.

## DVS/DVFS/AVFS

Dynamic voltage and frequency scaling (DVFS) techniques—along with associated techniques such as dynamic voltage scaling (DVS) and adaptive voltage and frequency scaling (AVFS)—are very effective in reducing power, since lowering the voltage has a squared effect on active power consumption. DVFS techniques provide ways to reduce power consumption of chips on the fly by scaling down the voltage (and frequency) based on the targeted performance requirements of the application. Since DVFS optimizes both the frequency and the



voltage, it is one of the only techniques that is highly effective on both dynamic and static power.

Dynamic voltage scaling is a subset of DVFS that dynamically scales down the voltage (only) based on the performance requirements.

Adaptive voltage and frequency scaling is an extension of DVFS. In DVFS, the voltage levels of the targeted power domains are scaled in fixed discrete voltage steps. Frequency-based voltage tables typically determine the voltage levels. It is an open-loop system with large margins built in, and therefore the power reduction is not optimal. On the other hand, AVFS deploys closed-loop voltage scaling and is compensated for variations in temperature, process, and IR drop using dedicated circuitry (typically analog in nature) that constantly monitors performance and provides active feedback. Although the control is more complex, the payoff in terms of power reduction is higher.

### Power Shutoff (PSO)

One of the most effective techniques, PSO—also called *power gating*—switches off power to parts of the chip when these blocks are not in use. This technique is increasingly being used in the industry and can eliminate up to 96 percent of the leakage current.

Power gating is employed to shut off power in standby mode. A specific power-down sequence is needed, which includes isolation on signals from the shut-down domain. Erroneous power-up/down sequences are the root cause of errors that can cause a chip re-spin. This needs to be correctly and exhaustively verified along with functional RTL to ensure that the chip functions correctly with sections turned off and that the system can recover after powering up these units.

Deploying power shutoff also requires isolation logic and possibly state retention of key state elements or, in other words, state retentive power gating (SRPG). For multi-supply voltage (MSV), level shifters are also needed.

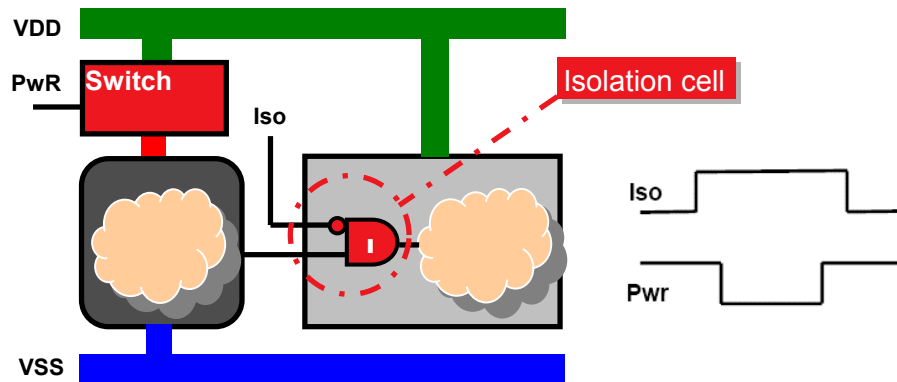
#### *Isolation*

Isolation logic is typically used at the output of a powered-down block to prevent floating, unpowered signals (represented by unknown or X in simulation) from propagating from powered-down blocks.

The outputs of blocks being powered down need to be isolated before power can be switched off; and they need to remain isolated until after the block has been fully powered up. Isolation cells are placed between two power domains and are typically connected from domains powered off to domains that are still powered up.

In some cases, isolation cells may need to be placed at the block inputs to prevent connection to powered-down logic. If the driving domain can be OFF when the

receiving domain is ON, the receiving domain needs to be protected by isolation. The isolation cells may be located in the driving domain, with special isolation cells, or they may be in the receiving domain.

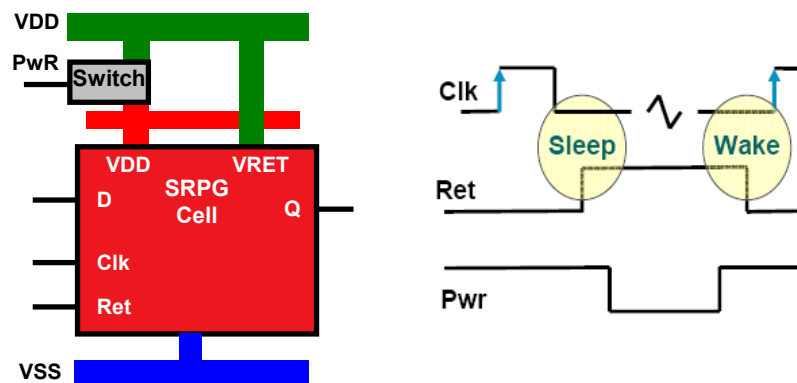


**Figure 7. Isolation gate and power-down switch**

### *State Retention*

In certain cases, the state of key control flops needs to be retained during power-off. To speed power-up recovery, state retention power gating (SRPG) flops can be used. These retain their state while the power is off, provided that specific control signaling requirements are met.

Cell libraries today include such special state retention cells. A key area of verification is checking that these library-specific requirements have been satisfied and the flop will actually retain its state.

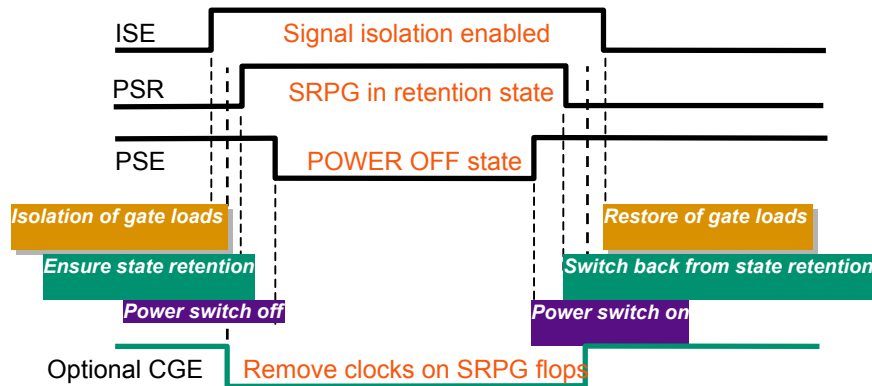


**Figure 8. State retention power gating**

### *Power Cycle Sequence*

For power-down, a specific sequence is generally followed: isolation, state retention, power shutoff (see Figure 9). For the power-up cycle, the opposite

sequence needs to be followed. The power-up cycle can also require a specific reset sequence.



**Figure 9. Power-up/down sequence**

Given that there are multiple—possibly nested—power domains, coupled with different power sequences, some of which may share common power control signals and multiple levels of gated clocks, the need for verification support is tremendous. The complexity and possible corner cases need to be thoroughly analyzed; functional and power intent must be analyzed and thoroughly verified together using advanced verification techniques.

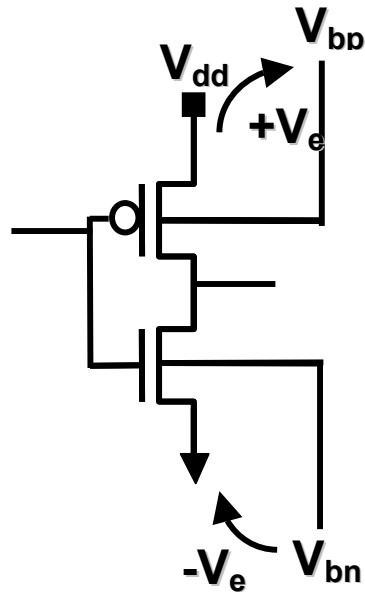
## Memory Splitting

In many systems, the memory capacity is designed for peak usage. During normal system activity, only a portion of that memory is actually used at any given time. In many cases, it is possible to divide the memory into two or more sections, and selectively power down unused sections of the memory.

With increasing SoC memory capacity, reducing the power consumed by memories is increasingly important.

## Substrate bias (Reverse body bias)

Since leakage currents are a function of device  $V_{th}$ , substrate biasing—also known as *back biasing*—can reduce leakage power. With this advanced technique, the substrate or the appropriate well is biased to raise the transistor thresholds, thereby reducing leakage. In PMOS, the body of transistor is biased to a voltage higher than  $V_{dd}$ . In NMOS, the body of transistor is biased to a voltage lower than  $V_{ss}$ .



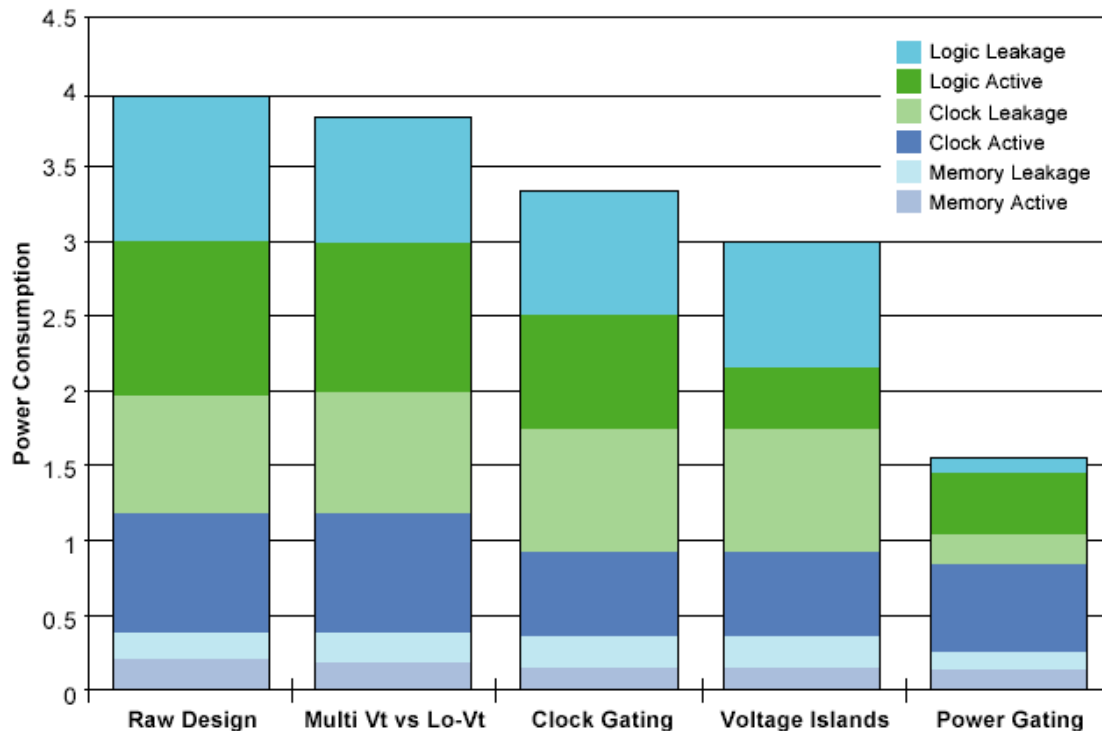
**Figure 10. Body bias**

Since raising  $V_{th}$  also affects performance, an advanced technique allows the bias to be applied dynamically, so during an active mode of operation the reverse bias is small, while in standby the reverse bias is stronger.

Area and routing penalties are incurred. An extra pin in the standard cell library is required and special library cells are necessary. Body-bias cells are placed throughout the design to provide voltages for transistor bulk. To generate the bias voltage, a substrate-bias generator is required, which also consumes some dynamic power, partially offsetting the reduced leakage.

Substrate bias returns are diminishing at smaller processes in advanced technologies. At 65nm and below, the body-bias effect decreases, reducing the leakage control benefits. TSMC has published information pointing to a factor of 4x reduction at 90nm, and only 2x moving to 65nm ([Ref. 3](#)) Consequently, substrate biasing is predicted to be overshadowed by power gating.

In summary, there are a variety of power optimization techniques that attack dynamic power, leakage, or both. Figure 11 shows the effect of introducing several power reduction techniques on a raw RTL design, on both active and static power.



**Figure 11. Power reduction techniques.** Courtesy Chip Design magazine, 2007 (Ref. 4)

### **The Need for a Common Power Format (CPF)**

Low-power design flows need to specify the desired power architecture to be used at each major step and for each task. Conventional design flows have failed to address the additional considerations for incorporating advanced low-power techniques. Consequently, design teams often resorted to methodologies that were ad hoc or highly inflexible. These methodologies required the designer to manually model the impact of low power during simulation, and provide multiple definitions for the same information: one set for synthesis, one for placement, one for verification, and yet another for equivalency checking.

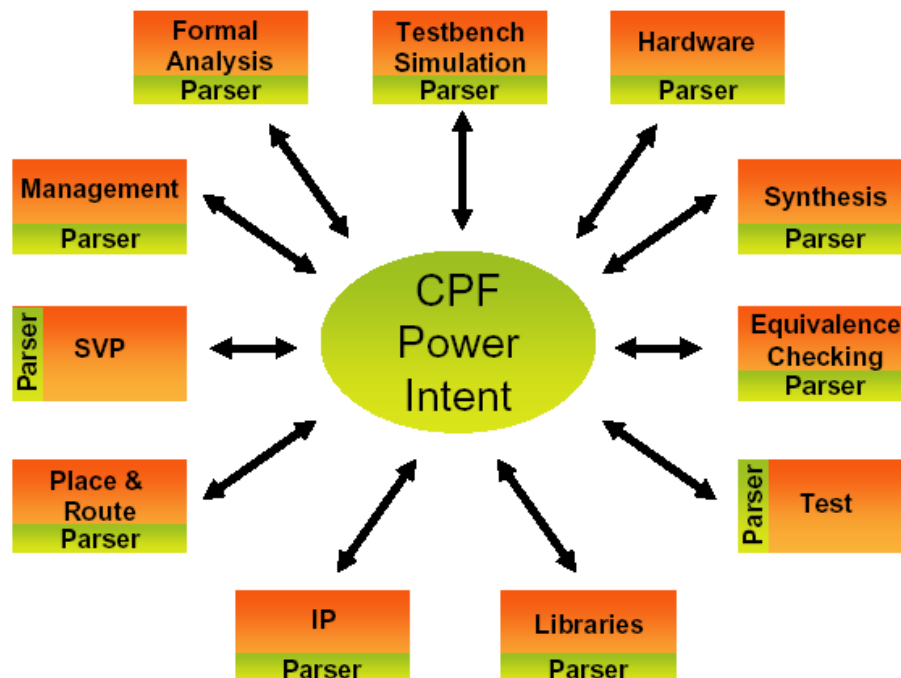
Yet after all that manual work, the old flows had no way of guaranteeing consistency. This posed a tremendous risk to the SoC; there was no way to be sure that what was verified matched what was implemented. The results were lower productivity, longer time to market, increased risk of silicon failure, and inferior trade-offs among performance, timing, and power.

To help design teams adopt advanced power reduction techniques, the industry's first complete low-power standard was developed. The Common Power Format (CPF), approved by the Silicon Integration Initiative (Si2), is a format for specifying power-saving techniques early in the design process, enabling them to share and reuse low-power intelligence.

The benefits of CPF include the following:



- **Improved quality of silicon (QoS):** Through easy-to-use “what-if” exploration early in the flow, designers can identify the optimal power architecture to achieve the desired specifications. Subsequently, optimization engines in the implementation flow help achieve superior trade-off among timing, power, and area targets.
- **Higher productivity and faster time to market:** A high degree of integration and automation helps design teams maintain high productivity levels. In addition, by reducing the number of iterations within the flow and limiting silicon re-spins, design teams can predictably address time-to-market concerns.
- **Reduced risk:** By providing functional modeling of low-power constructs, minimizing the need for manual intervention, and using a robust verification methodology, design teams can eliminate silicon failure risks that stem from functional and structural flaws.



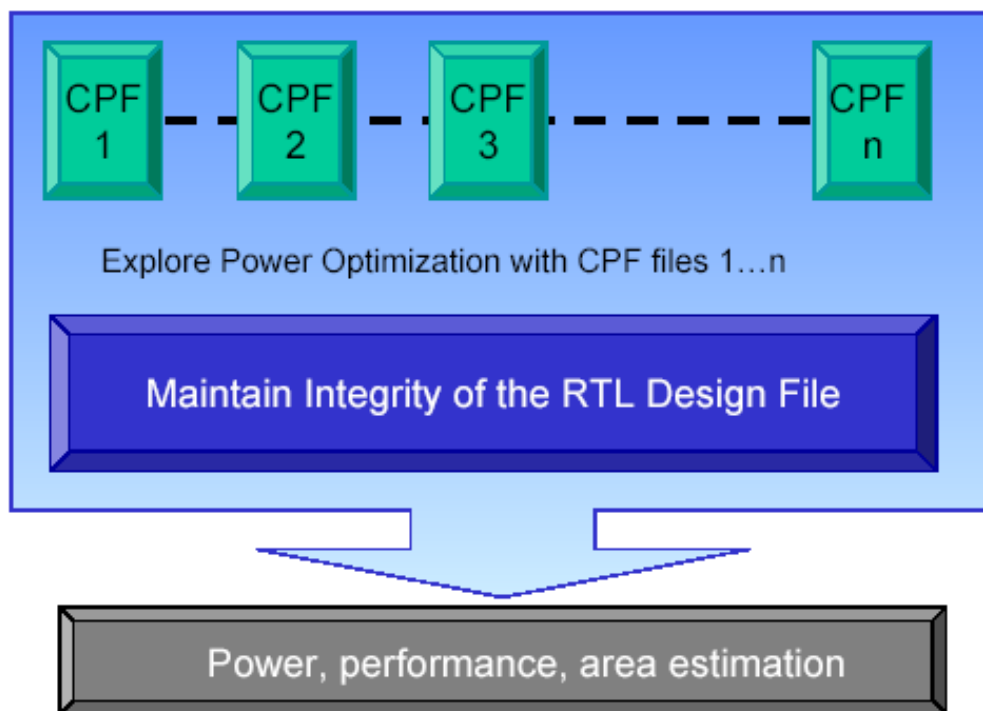
*Figure 12. CPF-enabled flow: Power is connected in a holistic manner*

### **Capturing Power Intent Using CPF**

The power intent for the full chip can be effectively captured using the Common Power Format. Advanced low-power SoC design tools support the low-power intent captured in the CPF commands. The RTL files are not modified with the power intent; power intent is inherently separate from design intent, and so is captured separately. The RTL files (design intent), CPF file (power intent), and SDC files (timing intent) capture the full design requirements.

In the past when designers had to change the RTL to include low-power constructs, it precluded design reuse. Designers found they had to change legacy code that was golden—and of course, if it were changed, it had to be verified again. And if the same block were used in multiple places in the design (as is common), designers would have to copy and modify the block for every power domain it was used in. This was a huge problem with the old flow; and consequently, a huge benefit for the CPF-enabled flow.

No RTL changes are required for a CPF-based flow; the power intent is captured in the CPF. With CPF, the golden RTL is used throughout the flow, maintaining the integrity of the RTL design file and enabling design reuse. The RTL can be instantiated  $n$  number of times, and each instance will have a different low-power behavior as specified by the corresponding CPF. The CPF file serves as an easy-to-use, easy-to-modify specification that captures power intent throughout the flow: design, verification, and implementation. It also contains library and other technology-specific information used for synthesis and implementation.



*Figure 13. Exploring power intent with CPF while preserving RTL*

### Using CPF to Capture Power Intent

The following example demonstrates how CPF can capture low-power intent for a design, specifically for multiple power domains with power shutoff.

In this design, the top-level contains two switchable power domains *pdA* and *pdB*, which can be powered down by the control signals specified by the individual ***–shutoff\_condition{}***. There is also a default power domain *pdTop*. All instances that are not assigned to a specific power domain are considered to belong to the *default* power domain. Figure 14 shows a block-level diagram for the design used to capture power shutoff intent, followed by a description of the CPF commands.

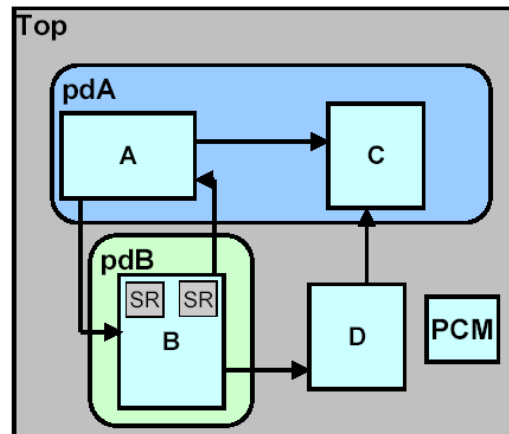
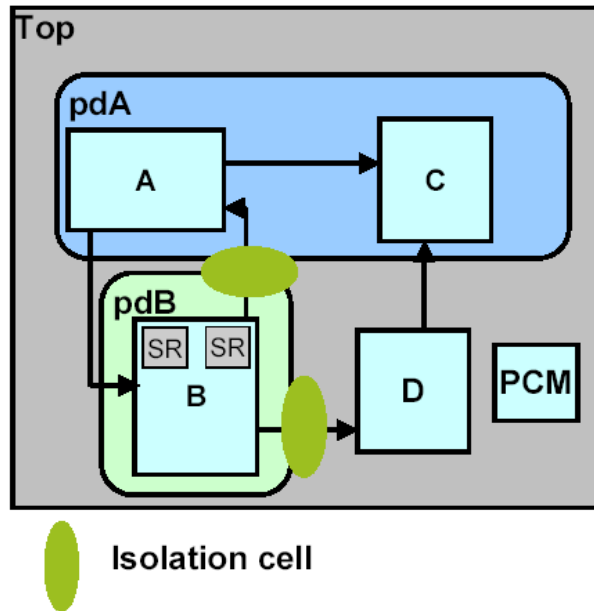


Figure 14. Multiple power domains and PSO

Below is the multiple power domain description using CPF:

```
# Define the top domain
set_design TOP
# Define the default domain
create_power_domain \
-name pdTop -default
# Define PDA
create_power_domain \
-name pdA \
-instances {uA uC} \
-shutoff_condition {!uPCM/pso[0]}
# Define PDB – PSO when pso is low
create_power_domain
-name pdB \
-instances {uB} \
-shutoff_condition {!uPCM/pso[1]}
```

When a block is powered down, the outputs need to be isolated and driven to the appropriate value. This is done by the ***create\_isolation\_rule*** command in CPF. Some key control flops need to be retained in a powered-down block. This is specified by the ***create\_state\_retention\_rule*** command.



**Figure 15. Isolation and state retention**

An isolation and state retention description using CPF follows:

```
# Active high Isolation
set hiPin {uB/en1 uB/en2}
create_isolation_rule \
-name ir1 \
-from pdB \
-isolation_condition {uPCM/iso} \
-isolation_output high \
-pins $hiPin
# Define State-Retention (SRPG)
set srpgList {uB/reg1 uB/reg2}
create_state_retention_rule \
-name sr1 \
-restore_edge {uPCM/restore[0]} \
-instances $srpgList
```

In this example, all power control signals are generated by an on-chip power controller, which may also be responsible for creating control signals for off-chip power regulators. CPF is TCL based. The example specifies the “pins” and “instances” that were created and will be recognized.

Power intent in CPF can be captured flat (from the top down) or hierarchically (bottom up). In situations where pre-existing IP is used, the IP will often have its own CPF describing state retention and isolation requirements. The CPF for the IP is used in the chip-level CPF.

## Complete Low-Power RTL-to-GDSII Flow Using CPF

### CPF-Enabled Design Tools

While some CPF commands are universal, there are individual commands that apply only to certain tools. As such, these individual tools ignore some CPF commands that do not contain useful information for them. For example, a simulator would ignore the CPF that specifies the timing and physical libraries information used in synthesis and physical implementation.

The following sections describe how each individual design and implementation tool uses the power intent specified in the CPF file throughout the low-power flow.

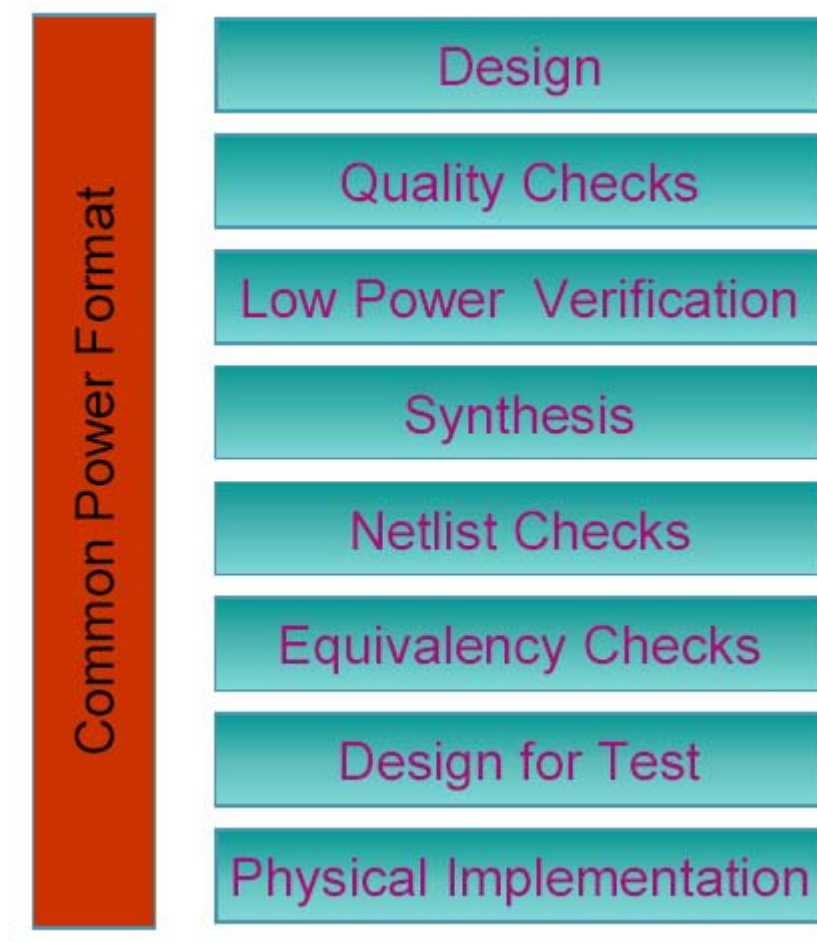


Figure 16. Low-power flow

### Verification of Power Intent

The first step in the low-power flow is to define and capture the design intent for the SoC in RTL, and the power intent by creating a CPF file. Power options can



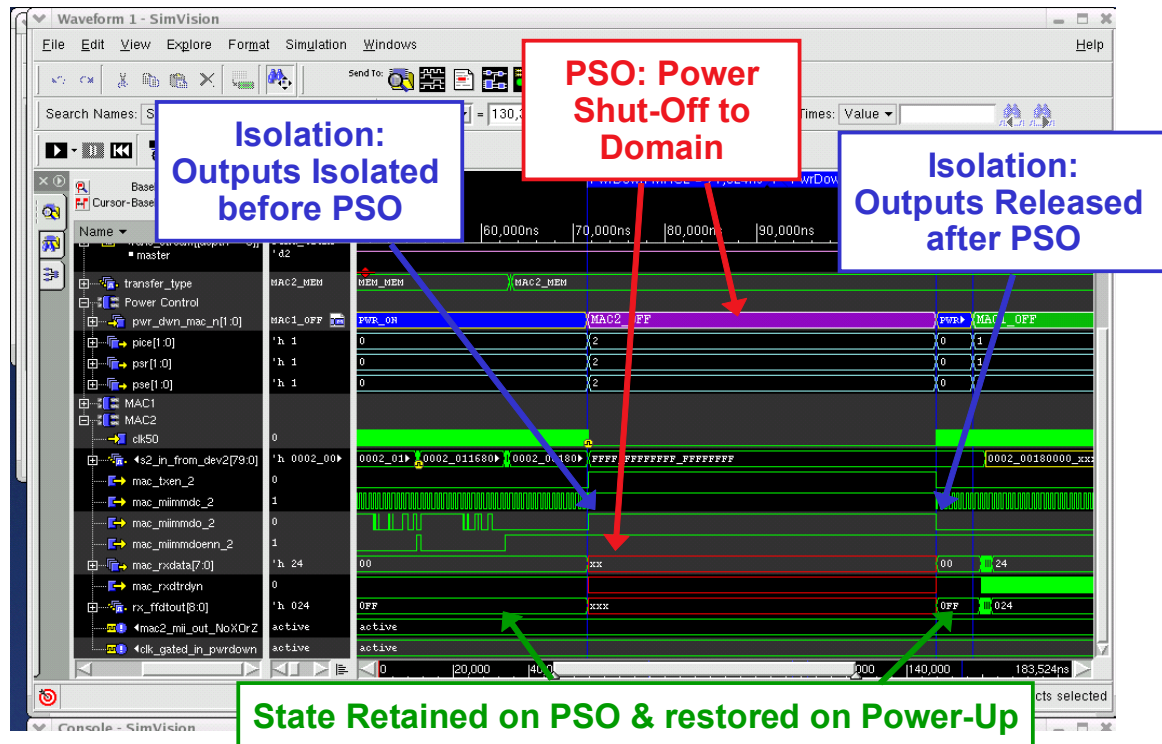
then be easily explored using CPF, while maintaining the integrity of the design as captured in the golden RTL.

The second step is to verify the contents of the CPF file using quality checks, which ensure that the CPF is syntactically correct, the power intent is complete, and the design and power intent are in alignment. For example, this stage can analyze the design and, using formal techniques, identify if there are missing isolation or level shifter definitions. Finding these missing definitions early by using formal techniques will save time in simulation and synthesis debugging later.

The next step in the low-power flow is to verify the correct functionality of the system with low-power behavior (CPF file) superimposed on top of normal functional behavior (RTL) through simulation.

In the flow described, PSO is effectively simulated to ensure that the chip functions correctly with sections turned off and that the system can recover after powering up these units. The control signals specified in the CPF for isolation, retention, and PSO are generated in the power controller. Low-power behavior is triggered in the simulator when the corresponding control signals are asserted. At the simulation stage of the flow, these control signals are not required to be connected to the design units in the various power domains. This will be done at the physical synthesis stage of the flow.

Note that no RTL changes are required as part of the CPF-based flow, and low-power cells need not be inserted in the RTL as part of the simulation process. Different power options can be explored by varying the power intent in the CPF and observing the corresponding low-power simulation behavior.



**Figure 17. Simulation of low-power behavior with Incisive Unified Simulator**

The simulator powers down part of the design, forcing all internal design elements to unknowns, or Xs. Just before power shutoff, the isolation signal is asserted—at which time, the simulator forces all outputs of the block to the specified CPF values. Between isolation and power-off, the retention signal is asserted by the power controller, which causes the simulator to store the current values of all retention flops specified in the CPF.

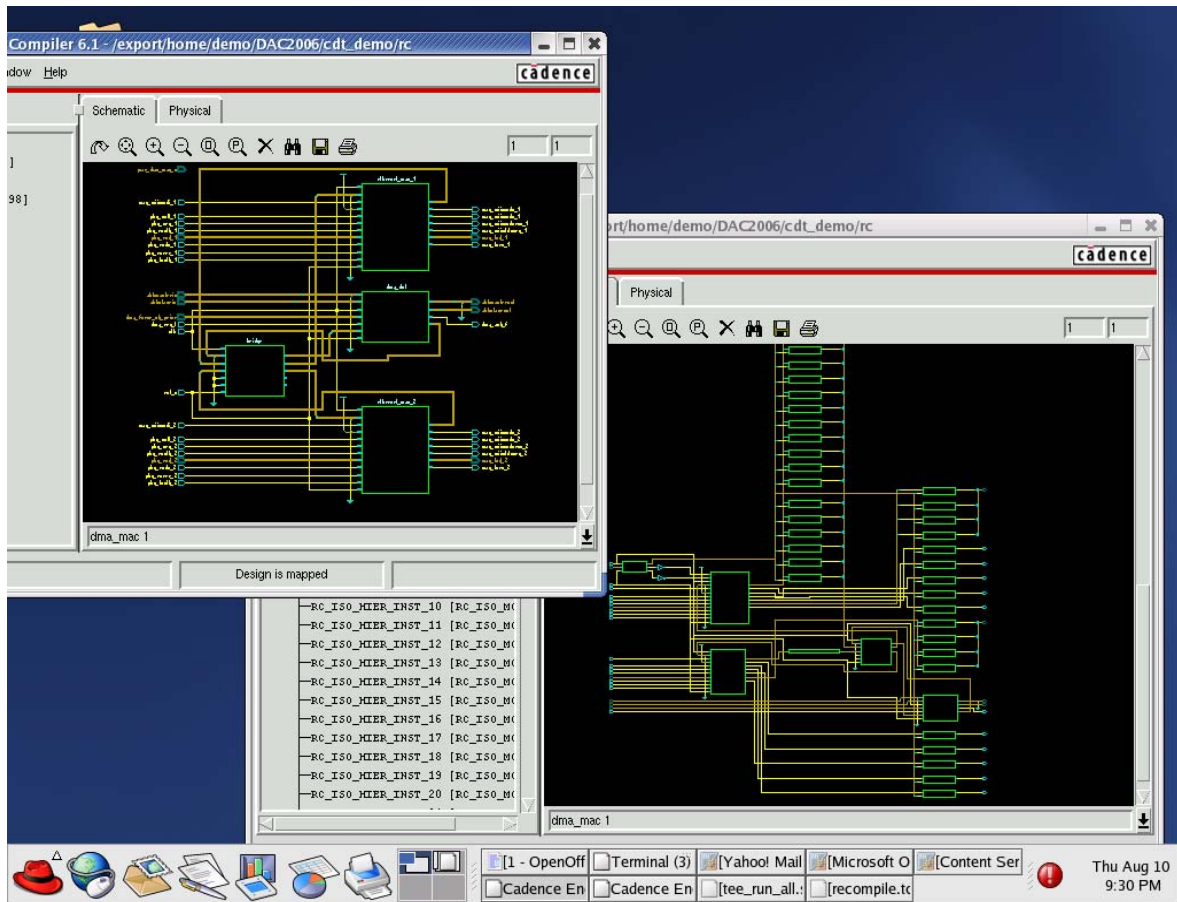
On power-up, the opposite sequence occurs: Power is switched on, followed by restoration of the retained values in the retention flop, and finally removal of the isolation values forced on the outputs. An important distinction is that the state retention and isolation are virtual at this stage; the RTL has not been modified in any way to emulate these functions. By making these virtual based on the CPF specification, the power intent is separated from the design intent, enabling design reuse.

For more details, see the chapter titled “Verification of Low-Power Intent with CPF.”

## Low-Power Synthesis

The design and verification tasks are iterative for optimal performance and power. Once the low-power behavior of a device has been verified to satisfy the design intent in the CPF commands, the next step is synthesis of the low-power features. In the synthesis phase, the low-power structures are synthesized directly into the gate-level netlist using the same CPF file used during simulation.

In Figure 18, the left screen shows the design synthesized without CPF. The right screen shows the same design synthesized after adding the CPF file to the synthesis constraints.



**Figure 18. Low-power synthesis using RTL Compiler**

The compiler infers the low-power behavior specified in the CPF and adds the following low-power cells to the design:

- Isolation cells to all outputs of power domains
- Isolation cells to inputs where specified
- Level shifters to signals crossing voltage domains
- Replacement of all flops with retention flops where specified

The synthesis tool inserts all low-power cells in the netlist except the power switches (elements that actually turn the block power on and off), which are inserted into the netlist during place and route.

As previously noted, during RTL simulation it is not necessary to hook up the power controller to the parts of design being powered down, isolated, etc. During simulation, virtual connections are created automatically by referring to the power control signals at the outputs of the power controller. During the synthesis phase, these virtual connections are replaced by RTL connections to the appropriate design units. All low-power cells are automatically connected during synthesis, as specified in the CPF and as simulated previously.

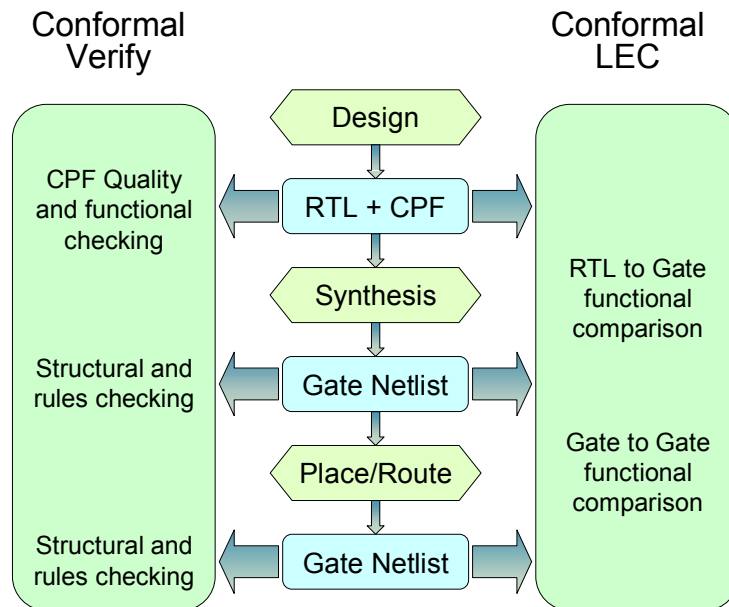
Modern synthesis tools can synthesize a design in multiple modes concurrently. One characteristic of having multiple power modes is the presence of different constraint files. This is especially true in DVFS applications, where the frequency is changed based on the current voltage level. Effective low-power synthesis requires the engine to optimize these different timing modes simultaneously. Optimizing just the “worst” timing is not sufficient, as different critical timing paths can be introduced in different modes.

The synthesis tool’s optimization engine automatically calculates the worst-case paths in the design. In addition, synthesis can support top-down multi-supply voltage synthesis, assigning different libraries to different voltage domains in the chip and performing top-down analysis and optimizations.

For more details, see the chapter titled “Front-End Design with CPF.”

### ***Structural Checks***

Formal verification, such as Cadence Conformal Low Power (CLP), is heavily used throughout the low-power flow as shown in Figure 19.



**Figure 19. Use of Conformal Low Power throughout the flow**

Formal verification of low-power designs encompasses two elements: low-power verification and logical equivalency. For low-power verification, the focus is on ensuring that the design is electrically correct from a low-power perspective. The flow will verify that the retention and isolation are complete and correct as specified in the CPF file.

Checks at this stage include tests for missing isolation or level shifter cells, checks that state retention and isolation control signals are driven correctly by domains that remain powered up, and tests for power control functionality. In later stages of the flow (post placement), these checks also ensure that gate power pins are hooked to the appropriate power rails, that the always-on cells are appropriately powered, and that there are no “sneak” paths from power-down domains back to logic.

Logical equivalency adds to the classic logical comparison. Logical equivalency checks (LEC) have been used for a number of years. The addition of low-power structures increases the complexity because isolation and state retention cells have been added to the netlist. These cells are not in the RTL, but are specified in the CPF. So the LEC tool must be able to formally prove that the synthesis engine has inserted these cells correctly, and that the netlist is logically equivalent to the golden RTL and power intent.

Note that these checks should be run throughout the entire flow. In particular, it is important to run these checks after synthesis and test logic insertion, and after



place and route (before tape-out). After tape-out quality routing, the checks should be run on a physical netlist, with power and ground connections.

### **Power-Aware Test**

Power complicates a chip's testability and the test logic insertion methodology. For low-power test, there are two key issues. First, the design must be testable. On-tester power consumption can dwarf operational power consumption, even at tester clock speeds, because efficient test patterns cause a very high percentage of the logic to be switching at a given time. Some chips would melt on the tester unless different blocks are shut down at different times, as they are in various functional modes of operation. So, for PSO test, scan chains must be constructed to minimize power domain crossing and to bypass switchable domains when they are shut down.

Once the design partitioning is understood, the second issue can be addressed. Power-aware manufacturing tests can be created. These tests now have two goals: limit the switching activity on the chip and test the advanced power logic such as level shifters, PSO logic, and state retention gates.

Current EDA solutions combine DFT capabilities, such as constructing scan chains that are power domain aware, with advanced test pattern generation. To reduce power consumption during manufacturing test, these power domain-aware scan chains can be controlled during test by inserting logic that enables direct control of which power domains are being tested. Combined with power domain-aware ATPG, this solution tests advanced power structures and reduces power consumed during test (see Figure 20).

Also, the vectors themselves can be constructed so that the changing values of the "filler" bits are controlled to reduce the switching activity. This means that the power consumed during the shifting of the scan patterns is controllable.

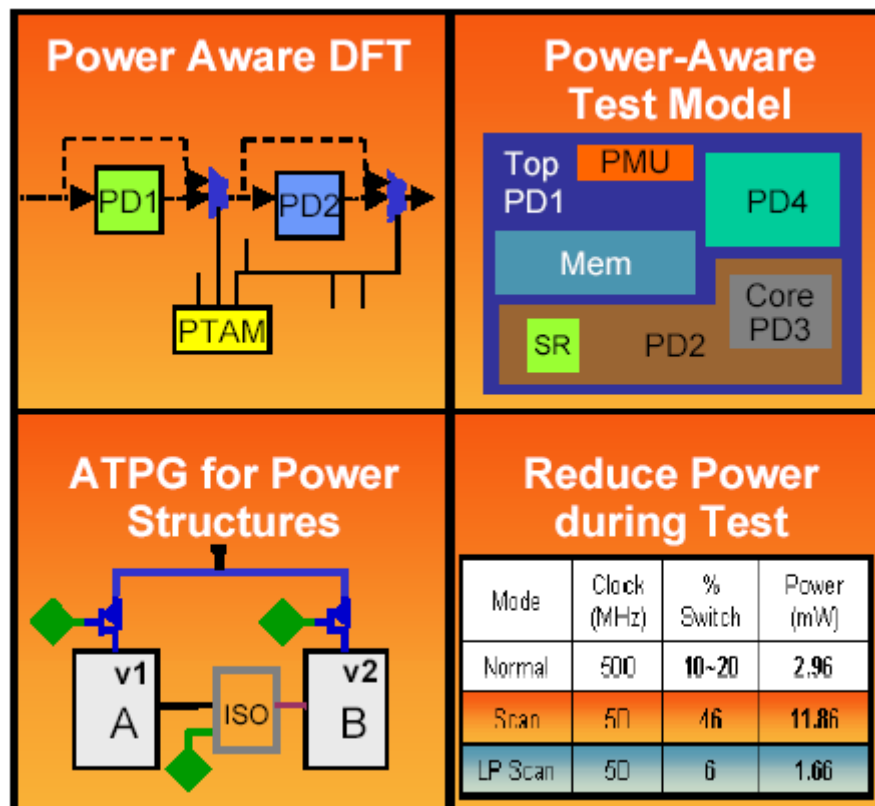


Figure 20. Test the low-power design, reduce power during test

### Low-Power Implementation

Once the gate-level netlist has been analyzed for structural and functional correctness, and functional equivalence checks have been run, back-end flow and implementation can occur. The low-power implementation flow enables physical implementation designers to achieve the lowest power consumption using an integrated, efficient flow. Using CPF power intent information that is consistent with the rest of the low-power flow, designers minimize power consumption, while preserving timing and area, and driving to signoff.

The flow starts with loading in the design and the CPF. The place and route software scans for relevant commands that are then applied to the design to identify power domains, power nets, switches, etc. Power domain and other low-power information comes directly from the loaded CPF file and does not have to be manually loaded, eliminating a time-consuming and error-prone engineering task.

A fully CPF-enabled low-power implementation platform implements low-power techniques, ranging from the basic to the most advanced. Its features include:

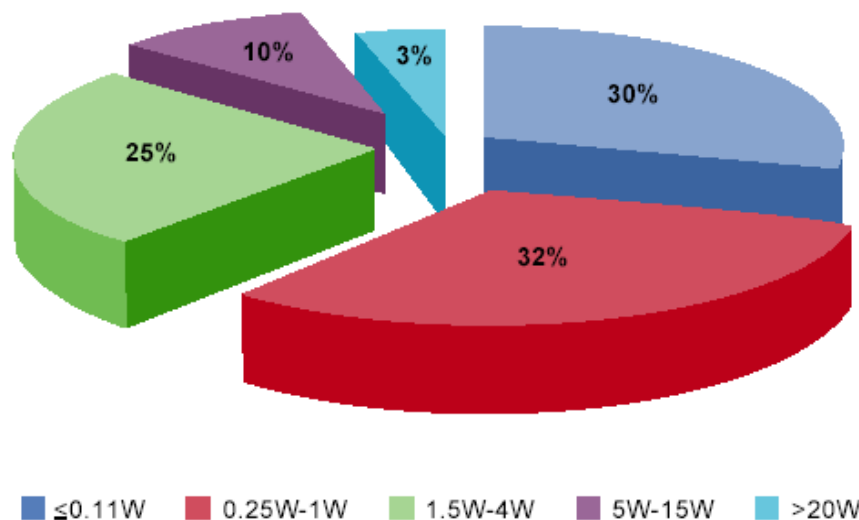
- Automatic power switch insertion
- Automatic generation of block-level CPF during partitioning
- Power domain-aware placement and optimization
- Power-aware clock tree synthesis
- Multimode, multicorn analysis and optimization
- Automated decoupling capacitor insertion
- Power- and SI-aware signoff timing analysis, including dynamic power analysis

For more details, see the chapter titled “Low-Power Implementation with CPF.”

## A Holistic Approach to Low-Power Intent

The requirement for low power will only accelerate. As shown in Figure 21, over half of the design investigations today are under 1W. Battery power, costs, and reliability are critical success factors for portables and consumer electronics. Even for products with higher-power budgets, like servers and routers, power-per-functionality goals and Energy Star requirements keep power issues in the forefront.

Total # Design Investigations Tracked = 19,720 (Jan 2006-Feb 2007)



**Figure 21.** Courtesy Chip Design Trends Newsletter, John Blyler, April 2007 ([Ref. 5](#))

Designing with low-power intent demands a holistic approach from RTL through GDS. As power starts to replace performance as the key competitive aspect of SoC design, new methodologies are emerging based on the Common Power Format standard.

CPF ensures power intent is preserved, integrated, and consistent throughout the entire flow: design, verification, and implementation.

## Verification of Low-Power Intent with CPF

Once the low-power intent for a design has been captured in CPF, the task of verifying it starts. The verification flow starts with the creation of a verification plan, which also contains metrics to measure the extent to which all low-power constructs in the design have been exercised. It also specifies the target coverage needed to meet the low-power verification goals.



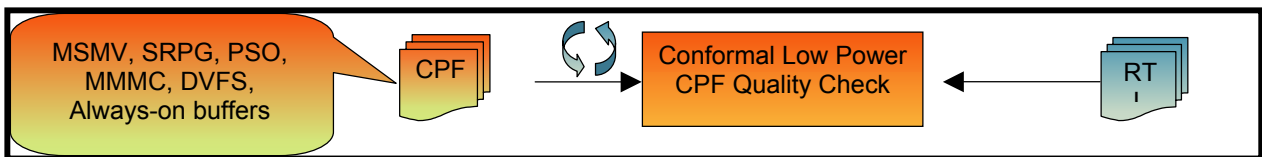
*Figure 22. Low-power verification flow*

## Power Intent Validation

The first task is to perform power intent validation, sometimes known as CPF quality checks. This actually verifies the correctness of the CPF file itself, and can be done with formal verification tools such as Conformal Low Power (CLP). The goal is to identify all CPF errors as soon as possible and have a clean CPF before starting the low-power simulation effort. A raw or freshly created CPF can have multiple errors in a variety of areas:

- Syntax
- Semantics
- Design object
- Inconsistent power intent
- Incomplete power intent

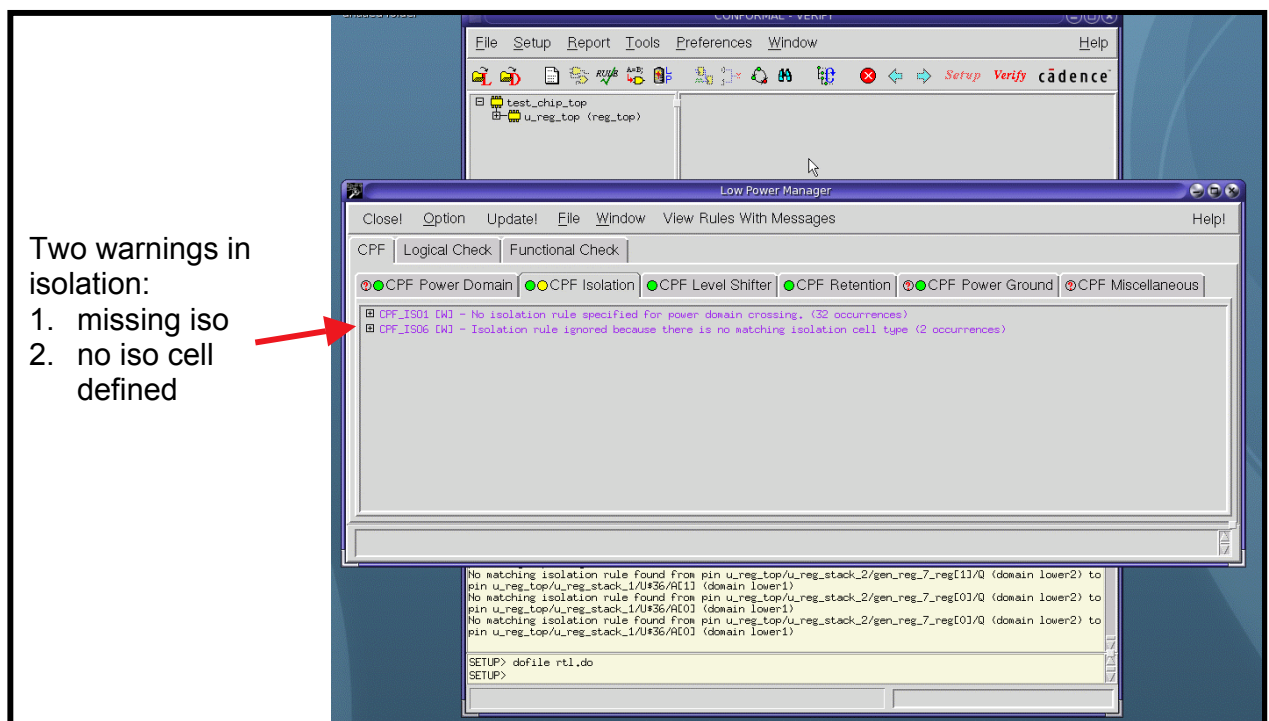
Power intent validation is run using CLP before low-power simulation and logic synthesis, as shown in Figure 23 and Figure 24.



**Figure 23. Power intent validation**

Some of the validation checks include:

- *Design object check* to see that all objects referenced in CPF are in the design database
- *Library CPF check* for consistency between CPF, Liberty, and LEF
- *CPF specification inconsistency check* to see, for example, if an isolation rule specifies an inconsistent location
- *CPF specification completeness check* to discover, for example, if there are missing isolation rule definitions between two power domains
- *CPF implementation consistency check* to find, for example, if a power net is not connected to any power domain in the RTL



**Figure 24. Power intent validation quality-check errors**

## Low-Power Verification

### Verification Planning and Management

Verification planning starts with bringing all the stakeholders together—including system engineers, architects, designers, and verification engineers—to capture the verification intent. It is “the process of analyzing the design specification with an aim toward quantifying the scope of the verification problem and specifying its solution” (Ref. 6). In other words, all parties must agree upon what needs to be verified, and how it will be verified.

A verification plan helps track the overall progress of the verification effort, while also providing an understanding of the functional coverage and identifying holes in the coverage space.

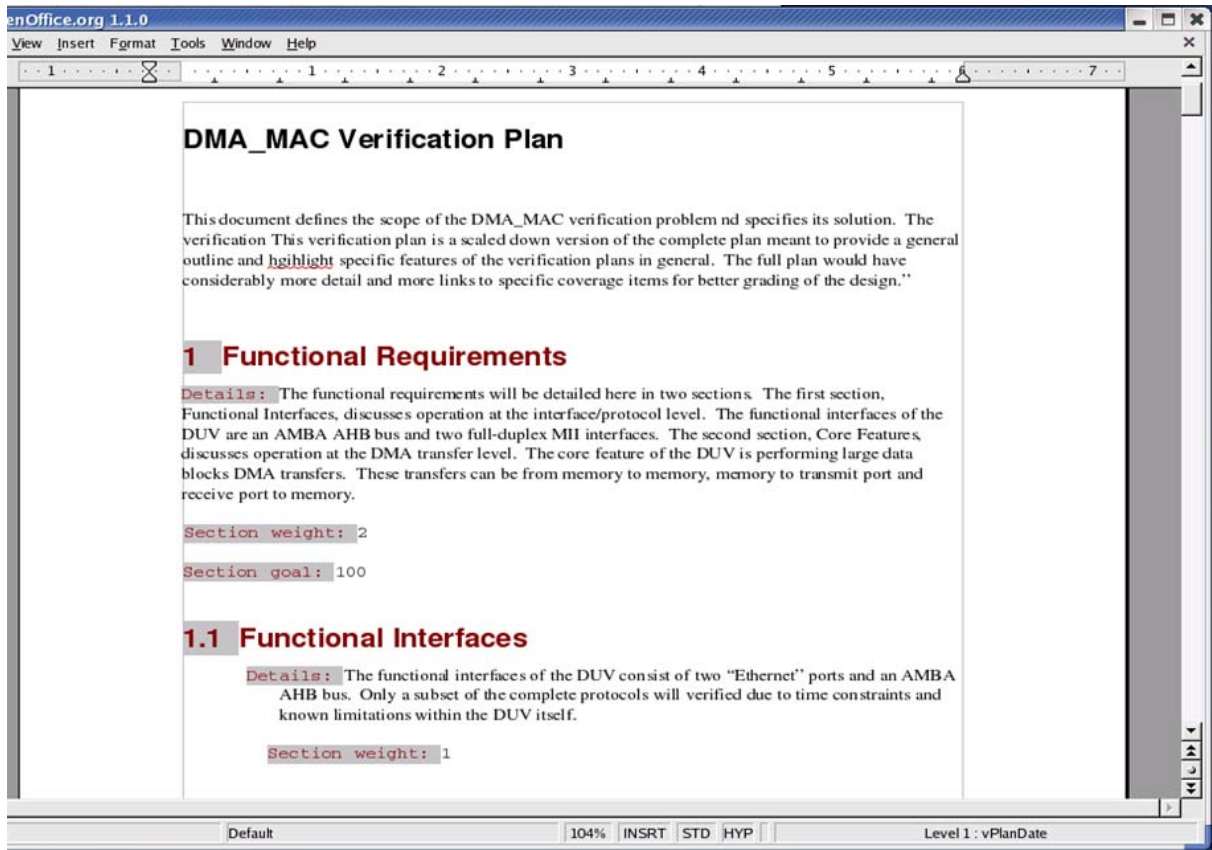
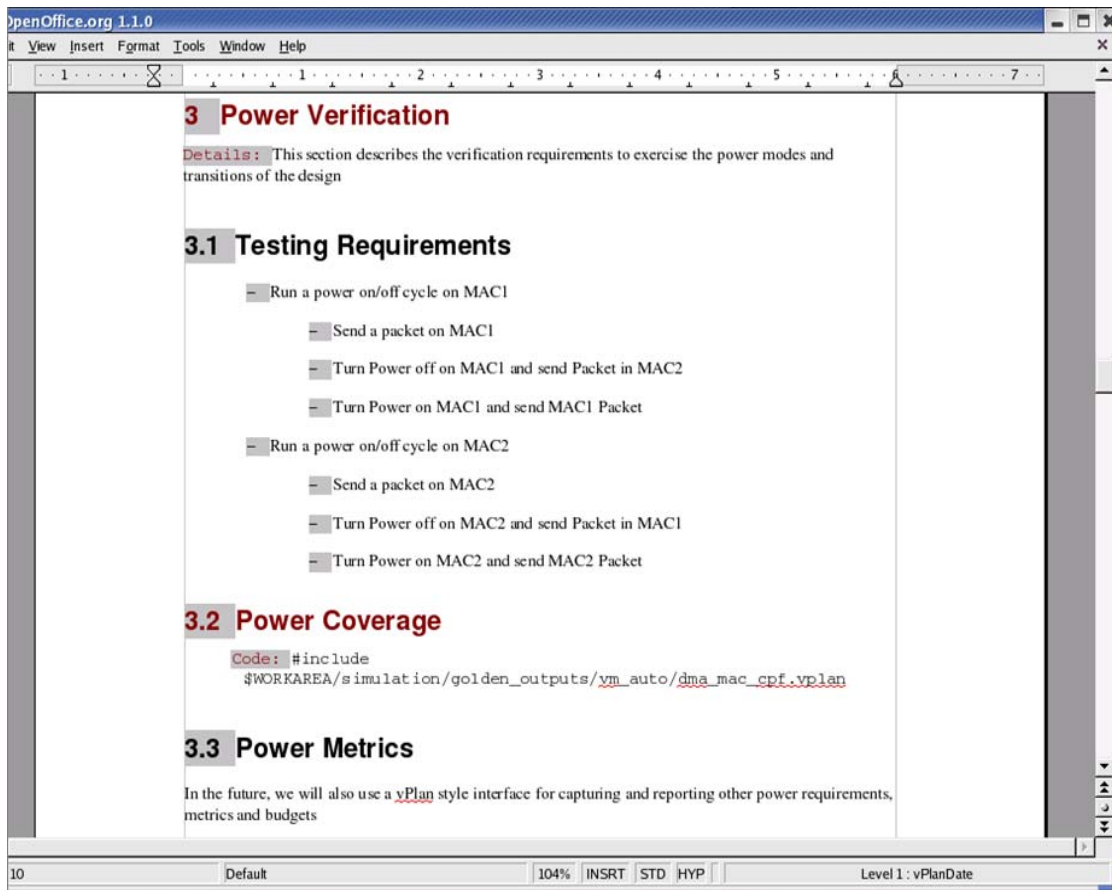


Figure 25. Device functional verification plan

### Capturing Power Intent

The verification plan must also contain a section on the verification of power intent. This section describes the verification requirements to exercise all power modes, and to control signal transitions that are needed to exercise the targeted power modes. It also specifies the desired behavior of design elements, and the

conditions and sequences of events that would lead to the design elements being in a desired power state (see Figure 26).



**Figure 26. Verification plan for capturing power intent**

### **Executable Verification Plan**

The end product of the planning stage is generation of a machine-executable verification plan that can be used to track the progress of the verification effort using metrics like functional coverage.

As shown in Figure 26, an executable verification plan for power coverage is automatically created from captured power intent and becomes part of the overall verification plan for the SoC.

### **The Role of Functional Coverage in the Verification of Power Intent**

Functional coverage is widely used in the industry to measure the quality of a verification effort and to answer the basic question, “Am I done verifying my design?” (Ref. 7) Similarly, functional coverage can be used to gauge—and quantitatively measure—the quality and completeness of the power simulations. This is done by first creating a coverage model around the power control elements



of the design, then managing the verification effort efficiently to optimize the collection of coverage data.

### **Functional Closure of Power Intent**

Power closure is achieved in two steps:

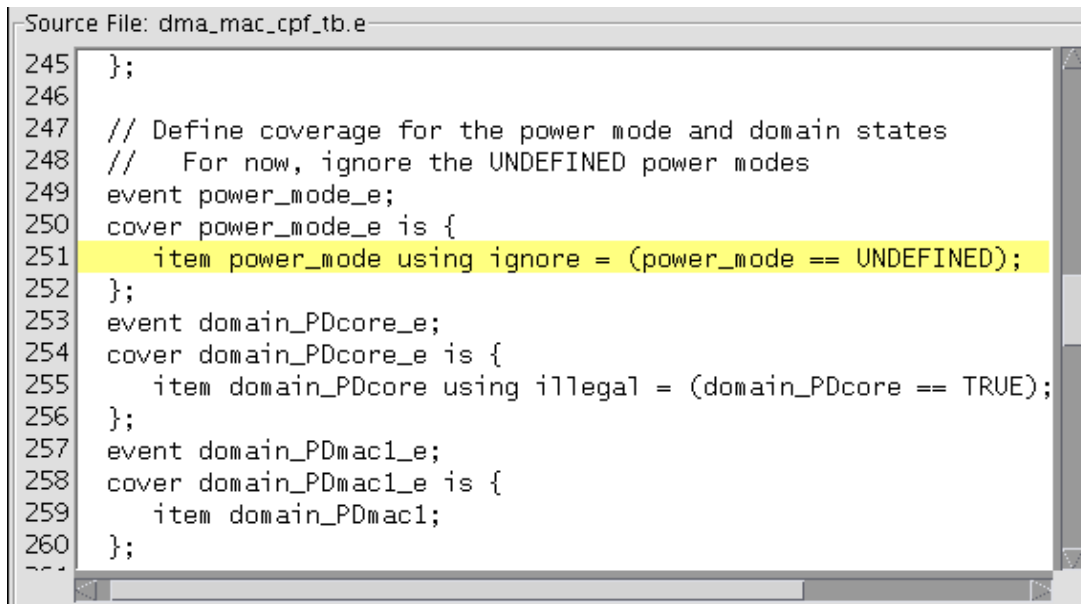
- Coverage model design for power intent
- Coverage-based closure of power goals

### **Coverage Model Design for Power Intent**

Once the features of interest have been extracted from the design and captured in the verification plan, the next step is to quantify the functionality that needs to be tested. This step is typically referred to as coverage model design (see [Ref. 8](#) for a detailed analysis and step-by-step process).

For low-power verification, how well the power intent has been functionally verified is measured by using functional coverage models to capture power intent. The cover groups needed to collect and capture metrics for low-power simulations are also automatically created.

These cover groups collect coverage for all power control signals, and track all power domains and power modes being exercised as well as mode transitions including illegal modes. The CPF file is parsed for intended power intent and the corresponding e code is generated automatically (see Figure 27).



```
Source File: dma_mac_cpf_tb.e
245 };
246
247 // Define coverage for the power mode and domain states
248 // For now, ignore the UNDEFINED power modes
249 event power_mode_e;
250 cover power_mode_e is {
251     item power_mode using ignore = (power_mode == UNDEFINED);
252 };
253 event domain_PDcore_e;
254 cover domain_PDcore_e is {
255     item domain_PDcore using illegal = (domain_PDcore == TRUE);
256 };
257 event domain_PDmac1_e;
258 cover domain_PDmac1_e is {
259     item domain_PDmac1;
260 };
---
```

**Figure 27. Power coverage models**

## Coverage-Based Closure of Power Goals

What does “closure” really mean in the context of achieving power goals? Power closure is formally defined as achieving predefined verification goals using specified metrics such as coverage. In Figure 28, the metrics are functional coverage from targeted cover groups created to measure power coverage and assertions. The coverage goals in the test case are specified in the executable verification plan and the results captured during simulation. As shown in the figure, the cumulative coverage results are then annotated onto the corresponding elements in the verification plan to reflect achieved verification goals. These are then used to determine power closure.

### Coverage Analysis—Achieving Closure

Coverage is one of the key metrics for determining the completeness of the verification effort. For low-power verification, coverage collected from automatically created cover groups is used to analyze overall completeness of the low-power verification effort. The executable verification plans (vPlans), also automatically created, are used to show overall cumulative coverage data over multiple simulation runs.

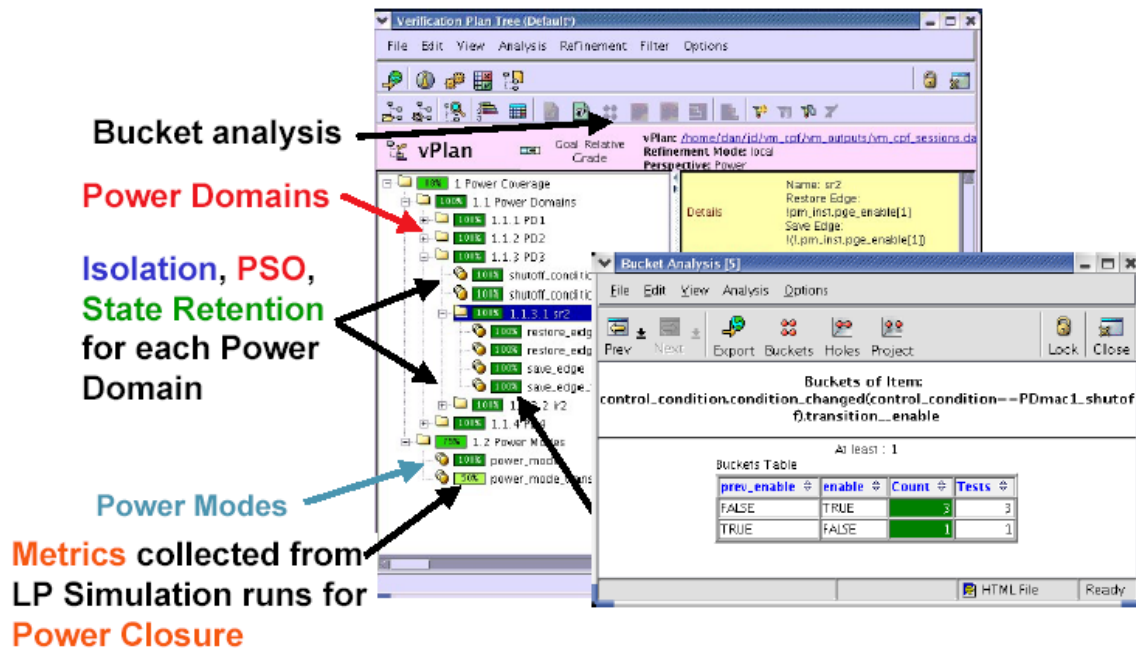


Figure 28. Power domain coverage data

On examining the results shown in Figure 28, all of the control signals have been fully exercised for each power domain (PD1, PD2, etc.). Further examination of one of the control signals for PSO shows that the signal has transitioned the

required number of times in each direction; that is, targeted functional coverage has been fully achieved, thus showing 100 percent for the power domains.

However, the overall value for power coverage is shown to be only 88 percent. On further analysis—that is, looking at the buckets for power mode—holes are identified in the coverage space that correspond to a missing test case. Some conditions have never been verified and need to be comprehensively covered in order to achieve power closure.

Let's take a closer look at the power mode coverage (see Figure 29). Coverage is collected for each power mode and for each valid power mode transition, as defined in the CPF. On running bucket analysis for a given mode transition, all mode transitions are examined.

It becomes clear that although all power domains have been fully exercised, certain legal and valid mode transitions have not occurred as part of the overall verification tests run so far. These holes in the coverage space need to be fulfilled in order to complete the task of verification and to achieve closure of the low-power verification effort.

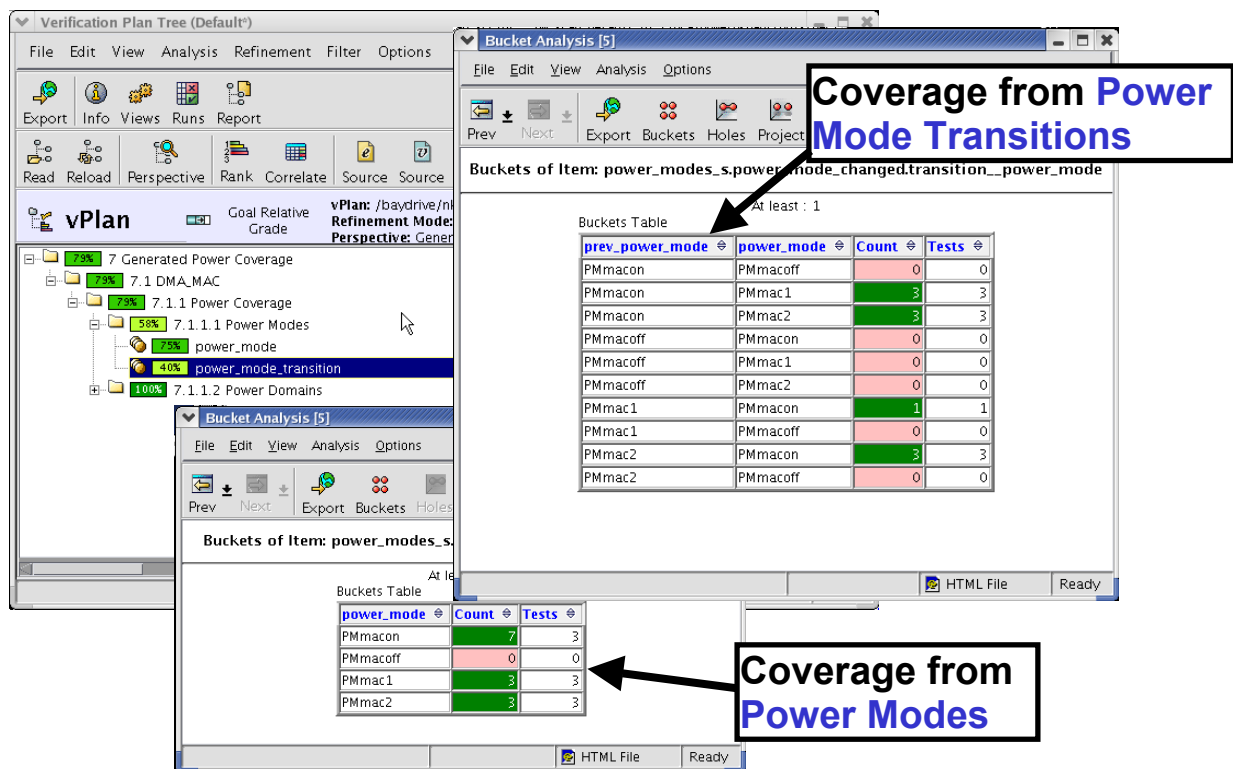


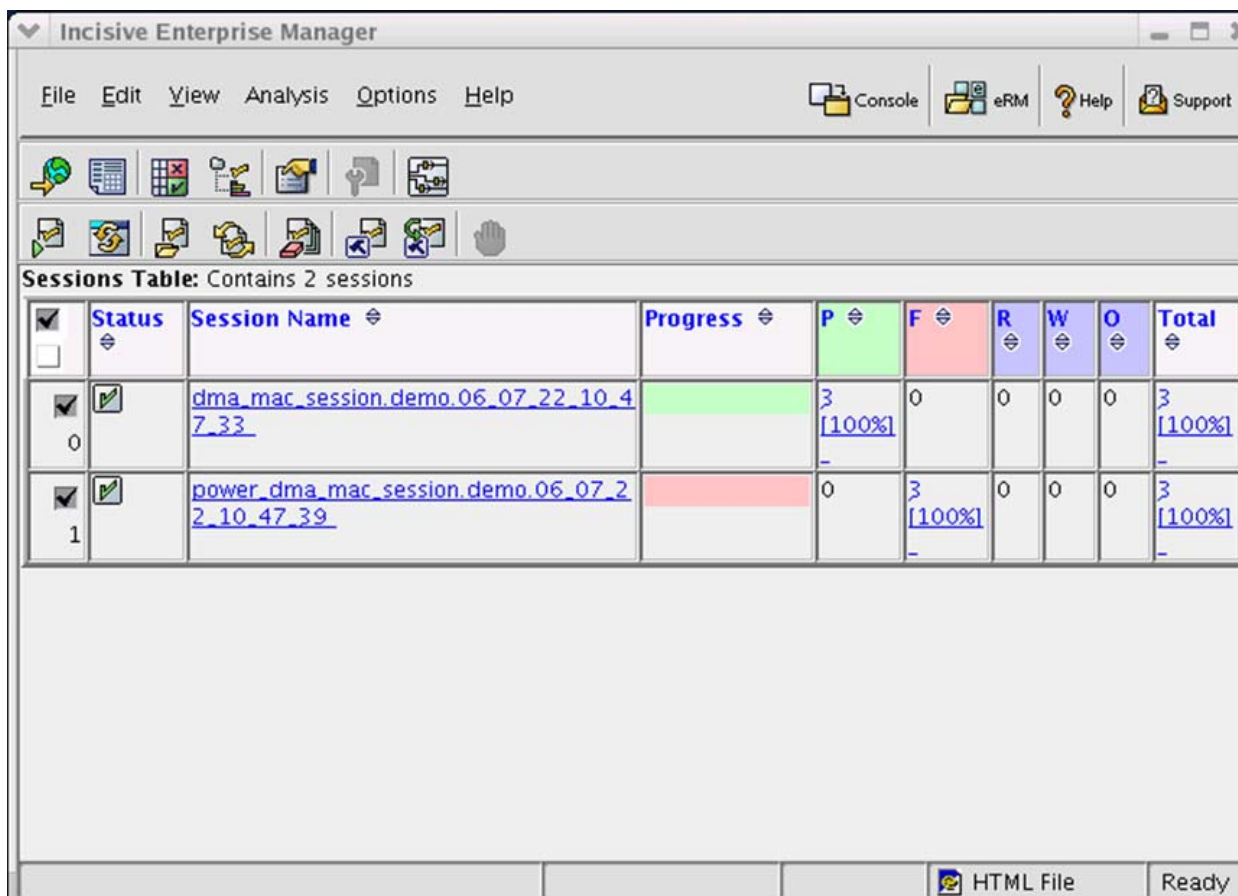
Figure 29. Power mode coverage

## Verification Management

The management of a large amount of simulation data is a daunting task in itself. When numerous sessions are run, each with its own variables, the amount of data becomes unmanageable. Analysis is very time-consuming, often requiring more time to analyze the data than to run simulations. The sessions can also span multiple platforms: hardware, software, accelerators, assertions, formal verification, etc. Effective data management very quickly becomes a key ingredient of the verification effort.

The main purpose is to manage, control, and automate the process of functional closure to achieve the verification goals. The goals can be specified in terms of metrics like functional coverage, or property proofs, or any other parameters that can track the progress and quality of verification itself.

The overall management of low-power data is done by tools like the Incisive Enterprise Manager, which manages, runs, and collects all metrics and other relevant data for each simulation run in each session (see Figure 30).



The screenshot shows the Incisive Enterprise Manager application window. It has a menu bar (File, Edit, View, Analysis, Options, Help) and a toolbar with icons for Console, eRM, Help, and Support. Below the toolbar is a 'Sessions Table' with the caption 'Contains 2 sessions'. The table has columns for Status, Session Name, Progress, P, F, R, W, O, and Total. Two sessions are listed: 'dma\_mac\_session.demo.06\_07\_22\_10\_4\_7\_33' and 'power\_dma\_mac\_session.demo.06\_07\_22\_10\_47\_39'. Both sessions show 100% progress. The status of the first session is 0 and the second is 1.

Status	Session Name	Progress	P	F	R	W	O	Total
0	<a href="#">dma_mac_session.demo.06_07_22_10_4_7_33</a>	100%	3	0	0	0	0	3
1	<a href="#">power_dma_mac_session.demo.06_07_22_10_47_39</a>	100%	0	3	0	0	0	3

Figure 30. Verification management—simulation sessions

Failure analysis is performed to correlate failed simulation runs to the run parameters. It is very useful for root-cause analysis, like first failures.

As seen in Figure 31, the root cause of failure that affects all three runs is the firing of an assertion, signifying the error that caused the first failure in all three runs. Automatic rerun of failing jobs can also be performed with management tools.

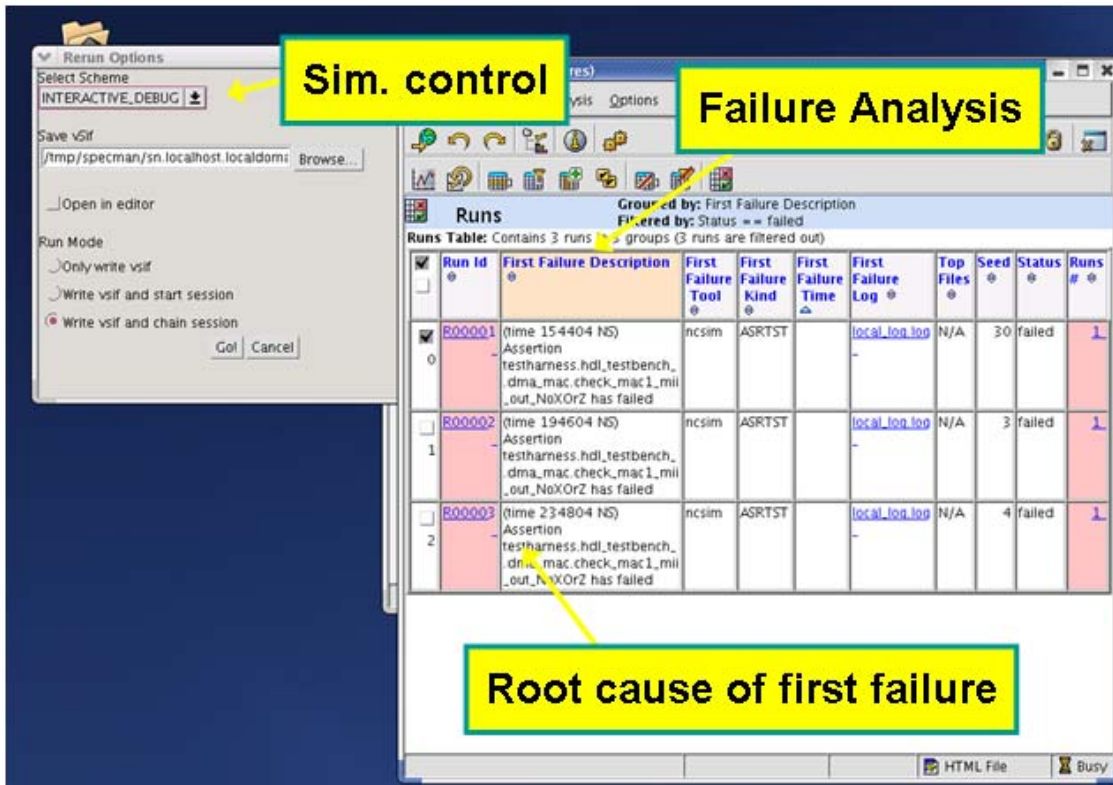


Figure 31. Verification management—failure analysis

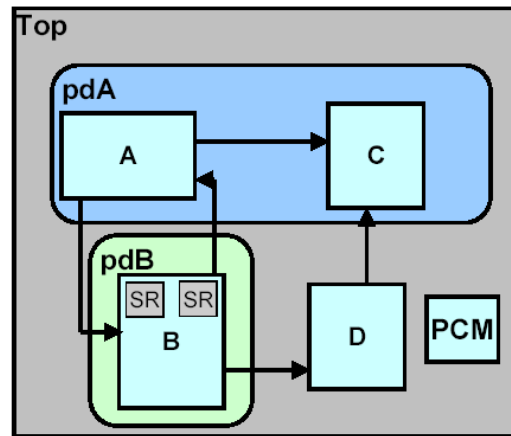
## Verification of Power Intent

Any effective low-power solution needs to truly augment functional RTL by capturing power intent in a form that can be used by all related tools—simulation, synthesis, and back end—for both functional and structural verification. The Common Power Format provides such a vehicle, as discussed in the following sections.

No RTL changes are required to capture power intent. With different low-power behavior specified in CPF, RTL instances can have different power behavior.

## Capturing Power Intent Using CPF

Figure 32 illustrates a circuit with multiple power domains and power shutoff.



**Figure 32. Multiple power domains and PSO**

Following is a description of a multiple power domain for the circuit using CPF:

```
# Define the top domain
set_design TOP
# Define the default domain
create_power_domain \
  -name pdTop -default
# Define PDA
create_power_domain \
  -name pdA \
  -instances {uA uC}
  -shutoff_condition {!uPCM/pso[0]}
# Define PDB – PSO when pso is low
create_power_domain
  -name pdB \
  -instances {uB} \
  -shutoff_condition {!uPCM/pso[1]}
```

The scope of the design for which CPF is intended is set by using the **set\_design** command. The CPF file for a hierarchical design can contain multiple **set\_design** commands. The first **set\_design** command specifies the top module of the design, which is at the root of the design hierarchy and is referred to as the top design.

Subsequent **set\_design** commands must each be preceded by a **set\_instance** command, which specifies the name of a hierarchical instance in the top design. The **set\_design** that follows this **set\_instance** command specifies the

corresponding module name of this instance. This module becomes the current design; design objects in the hierarchy of the module can be specified with respect to this current design.

All low-power simulations are controlled by the corresponding control signal asserted by the power controller in the design. Note that the actual control signals need *not* be connected manually to the appropriate power domains to enable low-power simulations. This is an added advantage for architectural explorations where different design units can be simulated with desired low-power behavior without modifying RTL in any way. Once the desired power configuration has been verified, the control signals can be automatically connected in RTL by the synthesis tool.

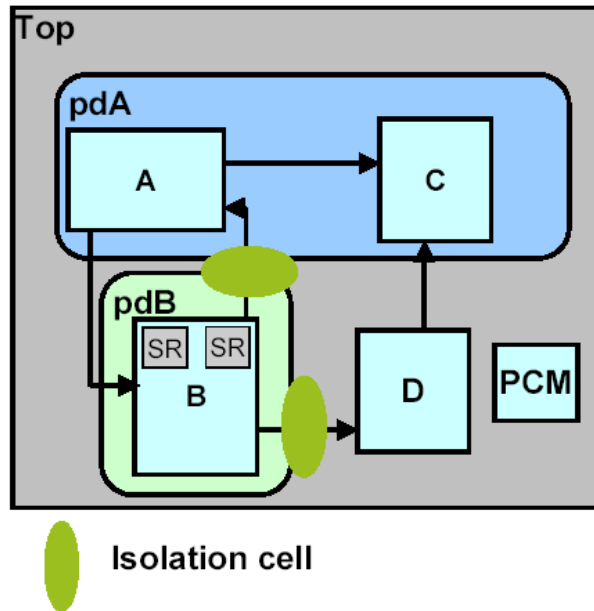
The ***create\_power\_domain*** command creates a power domain and specifies the instances and boundary ports and pins that belong to it. By default, an instance inherits the power domain setting from its parent hierarchical instance or the design, unless that instance was associated with a specific power domain. In addition, all top-level boundary ports are considered to belong to the default power domain, unless they have been associated with a specific domain.

Created power domains are associated with the design objects based on the order of the logical hierarchy. The order in which they are created is irrelevant. A default power domain must be specified for the top design, identified by the first ***set\_design*** command.

When a block is powered down, there is a need to isolate the outputs and drive it to the appropriate value. This is done by the ***create\_isolation\_rule*** command in CPF. Some key control flops need to be retained in a powered-down block. This is specified by the ***create\_retention\_rule*** command.

Figure 33 illustrates a circuit with multiple power domains and power shutoff, including isolation cells.





**Figure 33. Isolation for powered-down blocks**

Following is an isolation rule description using CPF:

```
# All outputs of Power-Domain pdB
# isolated high on rising edge of "iso"
sethiPin {uB/en1 uB/en2}create_isolation_rule \
  -name ir1 \
  -from pdB \
  -isolation_condition {uPCM/iso} \
  -isolation_output high \
  -pins $hiPin
```

The **create\_isolation\_rule** command defines a rule for adding isolation cells. Individual pins can be selected to have an isolation value of **high**, **low**, or **hold**. Both input and output isolation can be supported. A number of other conditions for isolation can be selected using an appropriate combination of the **-to** and **-from** options triggered by the control signal specified by the **-isolation\_condition**.

Isolation behavior is virtually imposed by the simulator based on the defined rules, without the need for isolation cells in the RTL. The isolation cells are then inserted using the same rules during the synthesis phase.

Now let's take a look at level shifters, as shown in Figure 34.

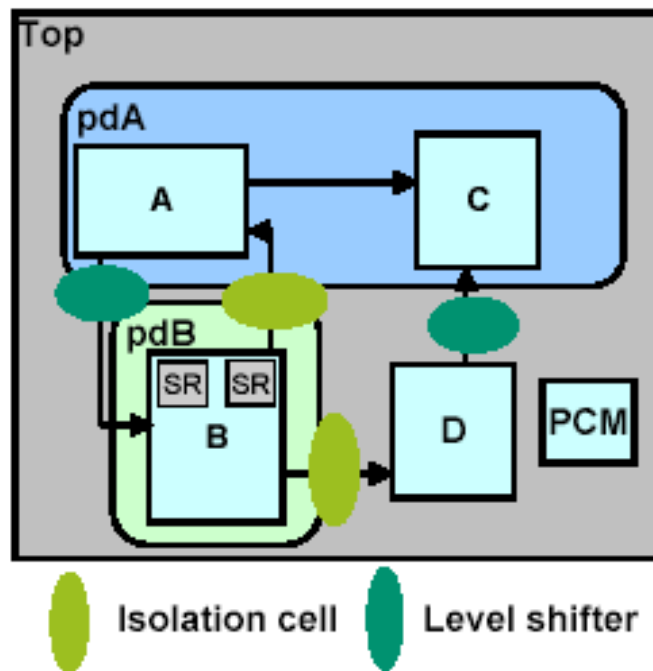


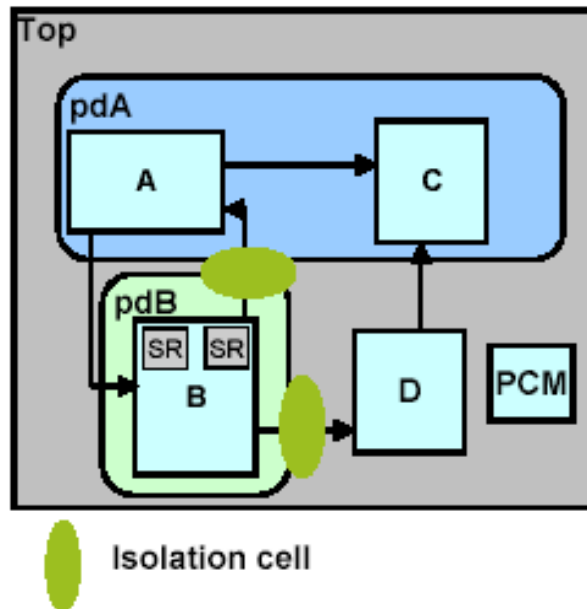
Figure 34. Level shifters

The CPF for the level shifters is as follows:

```
# Define Level-Shifters in the
# "to" domain
create_level_shifter_rule -name lsr1 \
  -to {pdB} -from {pdA}
create_level_shifter_rule -name lsr2 \
  -to {pdA} -from {pdB}
create_level_shifter_rule -name lsr3 \
  -to {pdTop} -from {pdB}
create_level_shifter_rule -name lsr4 \
  -to {pdA} -from {pdTop}
```

The **create\_level\_shifter\_rule** command defines rules for adding level shifters in the design.

State retention is also an issue in many designs (see Figure 35).



**Figure 35. State retention**

The CPF for the state retention is as follows:

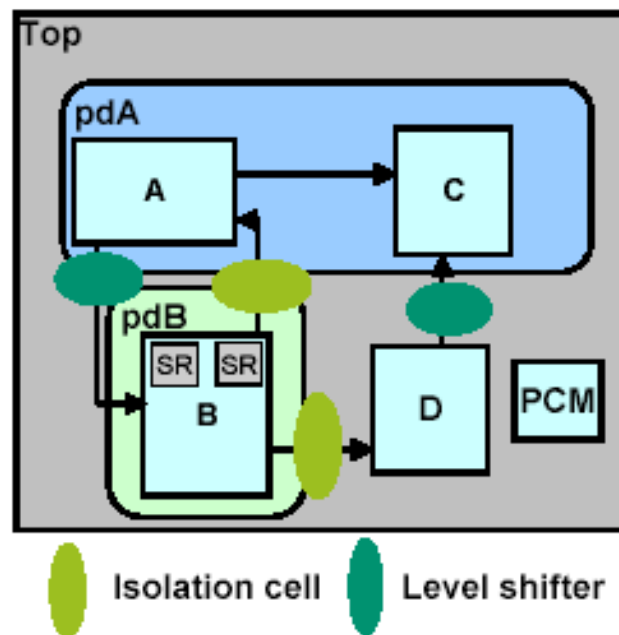
```
# Define State-Retention (SRPG)
# State stored on falling edge of
# restore[0] and restored on rising-edge
set srpgList {uB/reg1 uB/reg2}
create_state_retention_rule \
  -name sr1 \
  -restore_edge {uPCM/restore[0]} \
  -instances $srpgList
```

The **create\_state\_retention\_rule** command defines the rule for replacing selected registers or all registers in the specified power domain with state retention registers, as shown above. The store and restore behavior is triggered in simulation by the control signals from the power controller, as specified in the **–save** and **–restore** expression. Note that if **–save** is not specified, it is the logical **NOT** of the **–restore** signal.

The **create\_nominal\_condition** specifies a nominal operating condition with the specified voltage. It is used to track the different voltage levels required by individual power modes. Both are shown in Figure 36.

The **create\_power\_mode** command is used to define all legal modes of operation in a design such that each power mode represents a unique combination of operating voltage levels for individual power domains. This is needed to support power-saving schemes like dynamic voltage and frequency scaling (DVFS). Note

that at least one –**default** mode must be specified, which represents the power mode at the initial state of the design.



*Figure 36. Power modes*

The CPF for the power modes design is as follows:

```

# First, define the conditions.
# Top is always high, pdA/pdB can be
# medium or low
create_nominal_condition -name high \
    -voltage 1.2
create_nominal_condition -name medium \
    -voltage 1.0
create_nominal_condition -name low \
    -voltage 0.8
create_nominal_condition -name off \
    -voltage 0
# Define the modes
create_power_mode -name PM1 \
    -domain_conditions {pdTop@high \
        pdA@medium pdB@medium}
create_power_mode -name PM2 \
    -domain_conditions {pdTop@high \
        pdA@low pdB@low}
# Mode where pdB is off
create_power_mode -name PM3 \
    -domain_conditions {pdTop@high \
        pdA@low pdB@off}
# Close the design (for completeness)
end_design

```

### CPF-Based Low-Power Simulation

Once the power intent has been captured, the low-power simulator can simulate power cycles. Figure 37 shows a power shutoff sequence. Low-power behavior for power shutoff, isolation, and state retention is applied as specified in the CPF.

Power control signal definitions for Figure 37 and Figure 38, showing power shutoff and power-up sequences, are as follows:

- pice[1:0]: enable isolation on mac2 and mac1 respectively
- psr[1:0]: enable state retention on mac2 and mac1 respectively
- pse[1:0]: enable power shutoff on mac2 and mac1 respectively

In the power-cycle sequence, a specific sequence needs to be followed for both power-down and power-up cycles: isolation, followed by state retention, followed by power shutoff (see Figure 37). For the power-up cycle, the opposite sequence needs to be followed (see Figure 38). This needs to be constantly monitored.

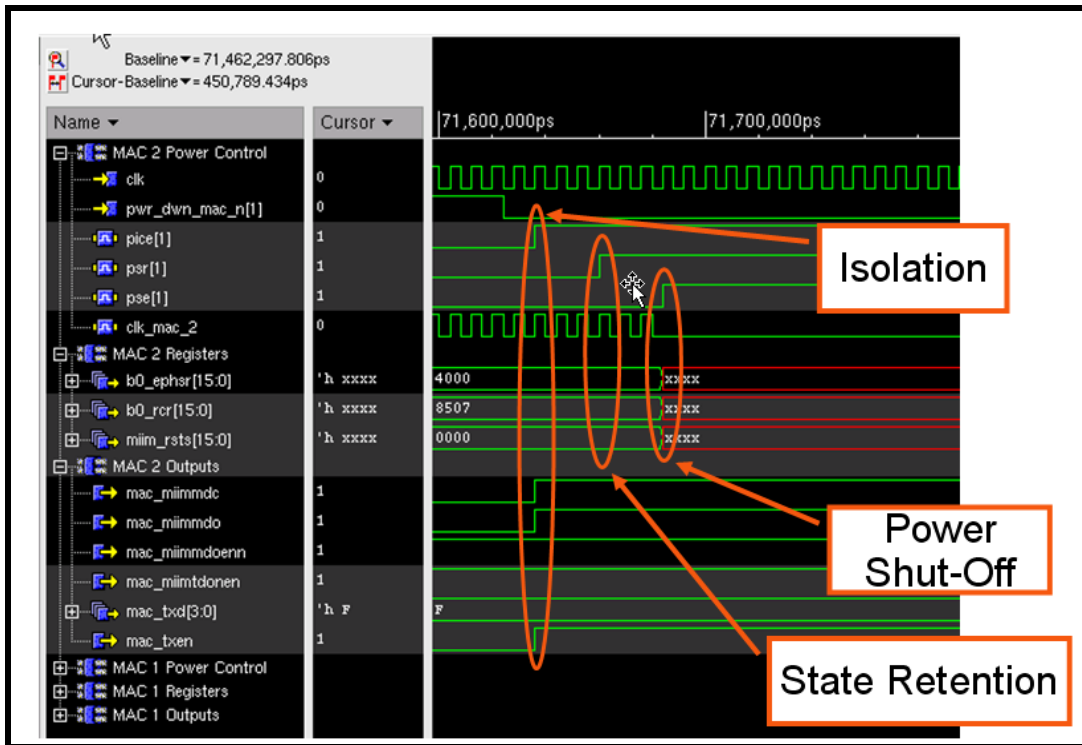


Figure 37. Power shutoff sequence

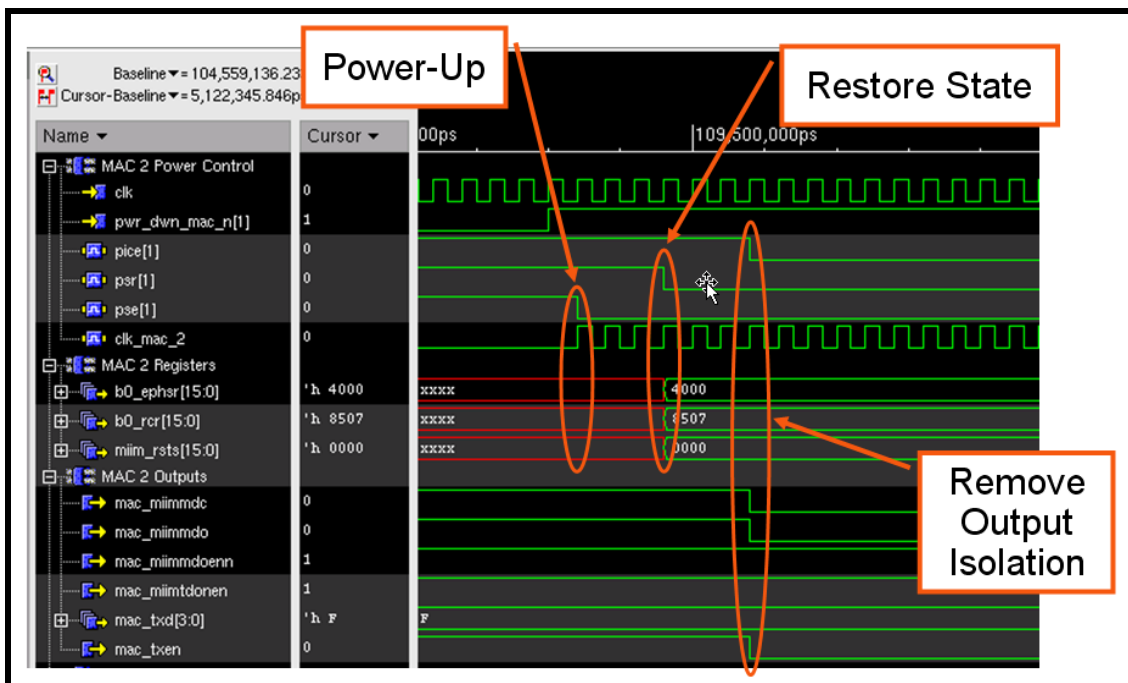


Figure 38. Power-up sequence

The following sections show how PSO behavior can be successfully verified:

- Power gating of targeted power domains
- Isolation of specified primary outputs
- State loss due to power shutoff of specified SRPG flops
- State restored on power-up of specified SRPG flops

## Failure Analysis

Failure analysis is the process of reviewing failed simulation results to determine the root cause of failures as they relate to the run-time parameters. While there are several factors that can lead to simulation failures, the emphasis in this section is on catching erroneous behavior while verifying power intent.

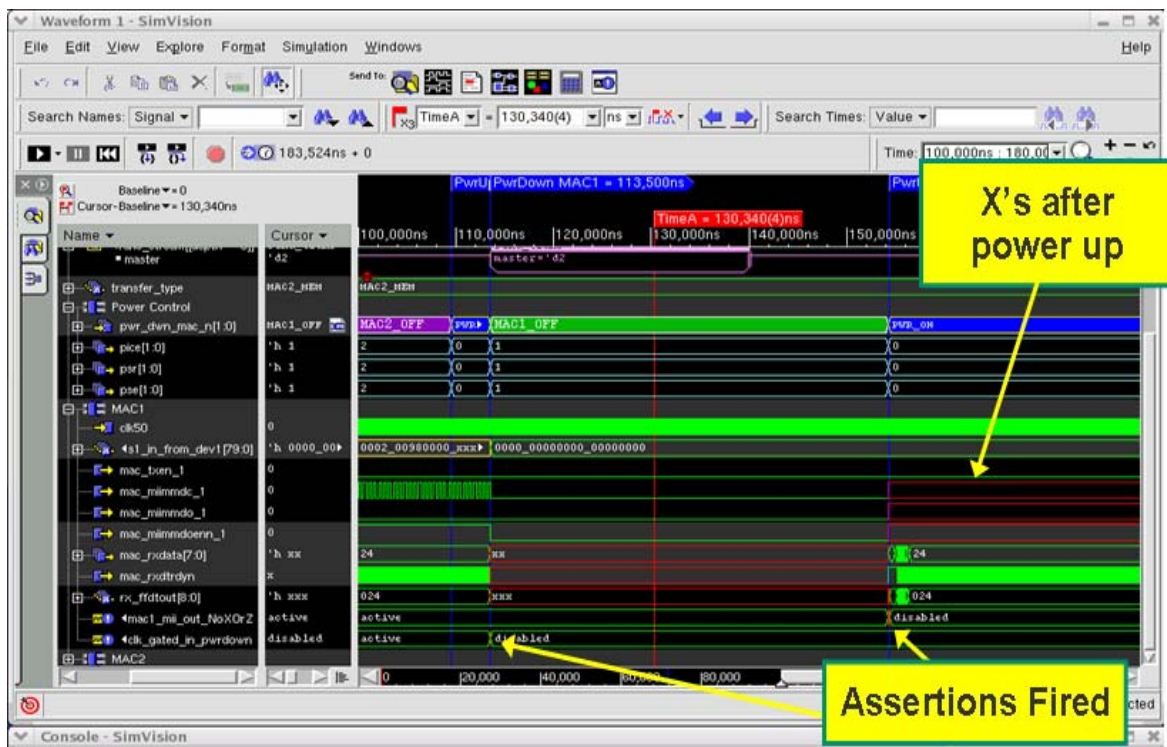


Figure 39. Incorrect sequence—power cycle with errors

## Assertion-Based Checks

The three main phases of interest during the simulation of low-power behavior are:

- *Power-down*: the time from when the device decides to power off until the device is actually powered off
- *Power shutoff*: the time taken until the device is actually shut off
- *Power-up*: the time from when the device decides to power up until it is actually operational



Note that the PSL assertion code segment in Figure 40 shows a power cycle with errors; the assertions flag incorrect PSO behavior during both power-down and power-up sequences. The PSL assertions show some examples of how assertion-based checkers are coded to catch erroneous behavior during the various stages of the power cycle shown in Figure 39.

Assertions provide coverage data to supplement those obtained from cover groups. They can also be used to define properties and constraints for designs being analyzed using a formal verification tool.

```
// Isolation must occur before Power-Off
PwrAsrt_ISO_before_PSO: assert always ({!iso; iso} |-> (!pso before !iso));

// Isolation must hold steady during Power Off
PwrAsrt_hold_ISO_in_PSO: assert always ({!iso; iso[*]; !pso} |-> (iso until ps0));

// Full power cycle
PwrAsrt_Full_PwrCycle: assert always ({!iso; iso[*2:5]; !pso[*]; ps0} |-> {iso[*]; !iso});
```

**Figure 40. PSL-based assertions for low-power control checks**

## CPF Verification Summary

Low-power verification is an important task in the overall low-power flow. In the old days, when low-power cells were manually inserted in the gate-level netlist—almost as an afterthought—potential bugs were introduced that were not verified. This resulted in many re-spins due to problems with missing or incorrect level shifters, power net connectivity, and other issues.

With the CPF-based flow, the effects of power management techniques such as MSV and PSO can be verified as part of the functionality of the device under test. The effects of different low-power trade-offs can also be easily verified by simply modifying the low-power intent in the CPF and running low-power simulations. Since this step does not require any changes to the golden RTL, it is very efficient.

Low-power assertions help detect any errors in the control signals that actuate and control low-power behavior in the device.

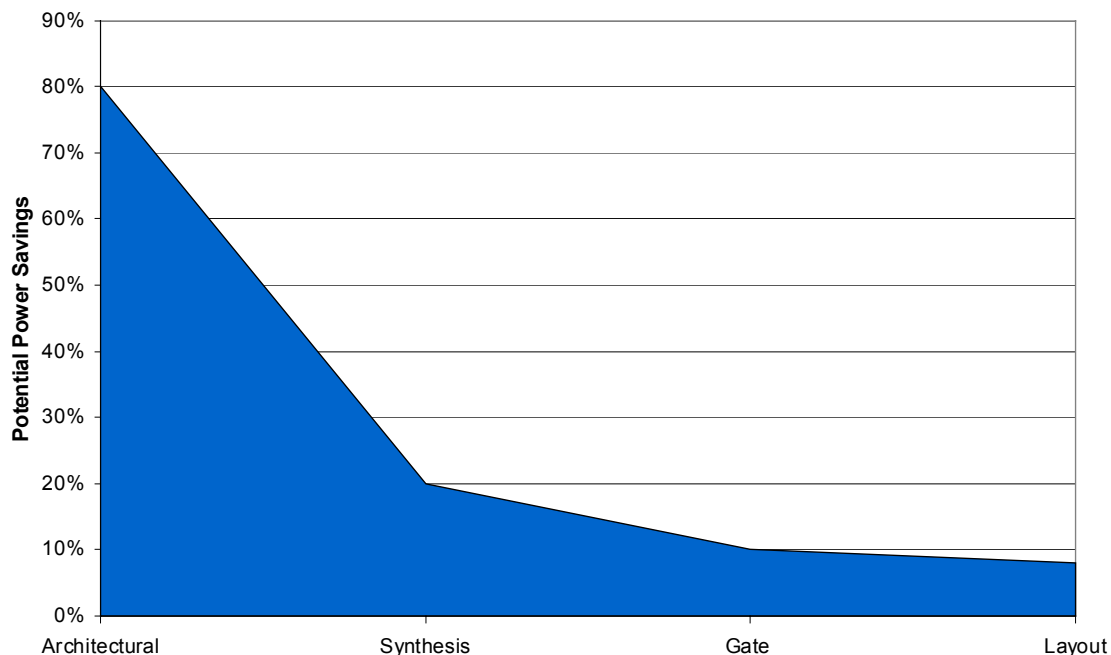
The low-power verification effort is assisted by automation that helps create an executable verification plan, which becomes part of the verification environment. Power coverage data is also automatically collected from low-power simulation, assisting in closure of the low-power verification effort.

Formal tools are used for automated checking of power intent captured in the CPF file and for syntactical, structural, and functional checks throughout the low-power flow.

## Front-End Design with CPF

### Architectural Exploration

When power targets are aggressive, it is important to design for low-power intent from inception. The earlier that power is considered in the design, the larger the savings can be. The majority of power is determined by decisions made at or before synthesis. Exploring various micro-architectures and their associated power architectures is possible only early in the design flow; it is too costly and time-consuming during implementation. CPF accelerates early optimization of power.



**Figure 41. Effect of power management early in the design**

### Exploring Micro-Architectures

A key decision in creating a low-power design is choosing the most appropriate micro-architecture, the state and processing elements, and how data flows. Especially at smaller geometries, the trade-offs between power, performance, and silicon area are not always intuitive.

For example, the IEEE 802.11a standard for wireless communications transmitters includes functional blocks such as the controller, scrambler, convolutional encoder, interleaver, and IFFT. The IFFT performs a 64-point Inverse Fast Fourier Transform (IFFT) on the complex frequencies; its architectural exploration follows.

Alternative micro-architecture implementations include a purely combinational version, a synchronous pipelined version, and five super-folded pipelined versions with 16, 8, 4, 2, and 1 bfy4 nodes, respectively.

The amount of energy required to process one OFDM symbol, with performance held constant, ranged from 4mW to over 34mW. Surprisingly, the 802.11 transmitter block design using the purely combinational IFFT consumed the least power, while the super-folded pipelined version using only a single bfy4 node consumed 8.5X more power ([Ref. 9](#)) During and after selecting the best micro-architecture, designers must trade off power, performance, and area (price of the silicon) with different power-saving techniques. But these techniques are only as effective as the micro-architecture allows.

### ***Exploring Power Architectures with CPF***

As the designer explores various micro-architectures to determine the best choice, the corresponding power architectures must also be considered. Building on an RTL architectural selection, CPF comes into play to rapidly explore power reduction with multi- $V_{dd}$ , multi- $V_{th}$ , dynamic voltage frequency scaling, power gating, etc. These techniques are highly design-dependent, may be used individually or in combination, and may be cumulative or (often) not. CPF allows early analysis of savings that can be achieved with various techniques for the particular RTL design—before investing significant time and effort in implementation.

Power architecture exploration is accelerated with the CPF-enabled synthesis flow. It is easier to explore architectures with CPF by changing the central power commands or power constraint file, not by changing the RTL. Working at a higher level of abstraction enables exploration of more power architectures in less time.

The design team should determine and plot the design's components of power consumption early on. This estimate can be done using a spreadsheet before any RTL is complete. Identify which blocks can afford the benefits of power reduction techniques. Estimating the ratio of leakage power to dynamic power for each block is also valuable, so designers can select appropriate power reduction techniques.

When RTL becomes available, the designers can do an RTL power analysis even before the design is synthesized. This analysis will not be as accurate as gate-level analysis, but will allow the designer to quickly explore the potential power savings achieved with a given technique. If the power analysis engine is integrated with the synthesis engine, the designer can also determine the effect of reduced voltage on design timing. The quick turnaround of RTL-based analysis lets the design team find the optimum power architecture early in the design flow.

Looking forward to implementation, the design team can evaluate the trade-offs between power savings vs. timing impact vs. area (price).

By using CPF, the ARC Energy PRO architecture proof-point project realized over 50 percent power reduction in certain modes of operation, with PSO and DVFS power management techniques. See the chapter titled “CPF User Experience: ARC” for more information ([Ref. 10](#)).

## Synthesis Low-Power Optimization

Selecting power optimization techniques in synthesis spans a variety of possibilities with concomitant benefits and penalties. The most common power management techniques for reducing power are reviewed in the following table, along with the impact of each on both active and dynamic power.

Naturally, the impact of power management techniques will vary dramatically based on the design itself, and the implementation of the low-power technique.

	Dynamic Power Savings	Leakage Power Savings	Timing Penalty	Area Penalty	Complexity and TTM Penalties	Implementation Impact	Design Impact	Verification Impact
<b>Dynamic power reduction techniques</b>								
<i><b>Clock gating</b></i>	20%	~0X	~0% <i>Clock tree insertion delay</i>	<2%	None	Low	Low	None
<i><b>Operand isolation</b></i>	<5%	~0X	~0% <i>May add a few gates to pipeline</i>	None	None	None	None	None
<i><b>Logic restructuring</b></i>	<5%	~0X	~0%	Little	None	None	None	None
<i><b>Logic resizing</b></i>	<5%	~0X	~0%	~0% to -10%	None	None	None	None
<i><b>Transition rate buffering</b></i>	<5%	~0X	~0%	Little	None	None	None	None
<i><b>Pin swapping</b></i>	<5%	~0X	~0%	None	None	None	None	None
<b>Leakage power reduction techniques</b>								
<i><b>Multi-<math>V_{th}</math></b></i>	0%	2–3X	~0% <i>Automated</i>	2 to -2%	Low	Low	None	None

<b>Multi-supply voltage (MSV)</b>	40–50%	2X	~0% <i>Adds level shifters; clock scheduling issues due to latency changes</i>	<10% <i>Power routing and power inter-connect; level shifters</i>	High <i>Design time, turnaround time, TTM</i>	Medium	Medium	Low
<b>DVFS</b>	40–70%	2–3X	~0% <i>Adds level shifters, power-up sequence; clock scheduling issues due to dynamic latency changes</i>	<10% <i>Adds level shifters and a power management unit</i>	High <i>Design time, turnaround time, TTM</i>	High	High	High
<b>Power shutoff (PSO)</b>	~0%	10–50X	4–8% <i>Adds isolation cells, complex timing, wakeup time, rush currents</i>	5–15% <i>Adds isolation cells, state retention cells, always-on cells; may have wider power grid due to rush currents; power management unit</i>	High <i>System architecture, support for power control, verification, synthesis, implementation, DFT</i>	Medium-high	High	High
<b>Memory splitting</b>	~0%	Varies	Varies <i>Adds isolation cells for power shutoff</i>	Varies	Varies	Medium-high	High	High
<b>Substrate biasing</b>	~0%	10X	10%	<10%	High	High	Medium-high	Medium

By defining the power intent in CPF, designers can synthesize their design using supported power reduction methods like clock gating and multi- $V_{th}$ , and determine

if power requirements will be met. If more complex techniques with a higher penalty are needed to meet the target power budget, CPF is helpful in exploring these methods.

This is possible because synthesis is power-aware, with CPF providing added value and automated control of the downstream SoC implementation without manual intervention.

Using multiple CPF files, different power architectures can be synthesized and analysis can be performed to determine the feasibility of a given power architecture. For example, implementing power gating as a power reduction technique is a question of trade-offs, since it may require many state retention and always-on cells, while the power advantage may be marginal. Using low-power cells like state retention, isolation, and level shifters can have a significant impact on timing and physical design.

## Automated Power Reduction in Synthesis

First, we will address the techniques that are automated in today's synthesis tools and do not specifically require CPF. In design with low-power intent, synthesis tools automatically perform a variety of power optimization techniques, including clock gating, operand isolation, logic restructuring, logic resizing, transition rate buffering, and pin swapping.

It is also possible to define various types of optimization in CPF for synthesis, to automatically implement other techniques during synthesis, such as MSV, DVFS, and PSO.

### Clock Gating

In most designs, data is loaded into registers very infrequently, but the clock signal continues to toggle at every clock cycle. Often, the clock signal drives a large capacitive load, making these signals a major source of dynamic power dissipation.

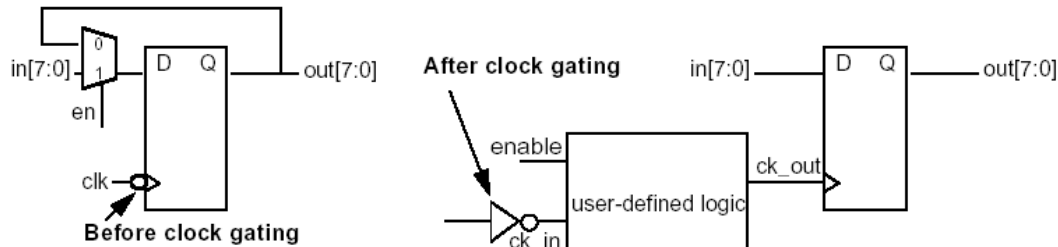
Clock gating reduces power dissipation for the following reasons:

- Power is not dissipated during the idle period when the register is shut off by the gating function.
- Power is saved in the gated-clock circuitry.
- The logic on the enable circuitry in the original design is removed.



## Clock-Enabled Register Example

Consider a multiplexer (MUX) at the data input of a register. This MUX is controlled by an enable signal. The inferred logic block in the original RTL, before and after the clock-gating attribute is set, is shown in Figure 42.

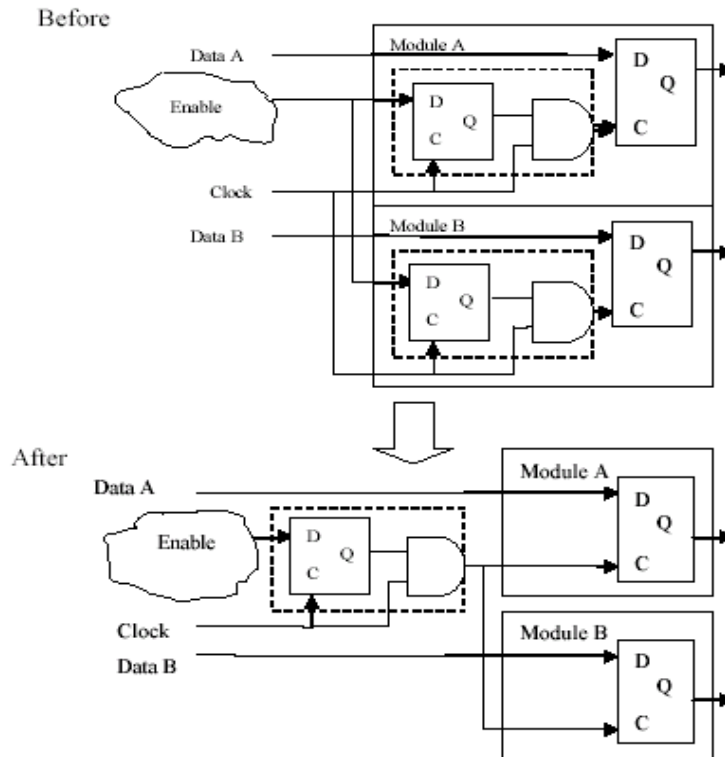


**Figure 42. Clock gating**

Synthesis sees this type of description as a perfect candidate for clock gating. If the data input to a flip-flop can be reduced to a MUX between the data pin and the output pin of the flip-flop, the synthesis tool can model this flip-flop by connecting the “data input” directly to the data pin of the flip-flop, and by using the MUX enable to gate the clock signal of the flip-flop via an inserted clock-gating element as illustrated.

## De-Cloning Local Clock Gating

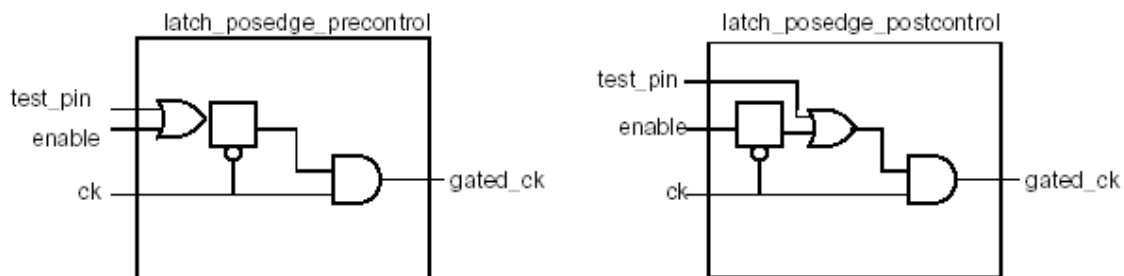
If the clock-gating logic of different registers in the design uses the same enable signal, RTL Compiler can merge these clock-gating instances for any such identically gated registers. This process is called clock-gating de-cloning, shown in Figure 43.



**Figure 43. Clock-gating de-cloning**

### DFT Attributes for Clock-Gating Logic

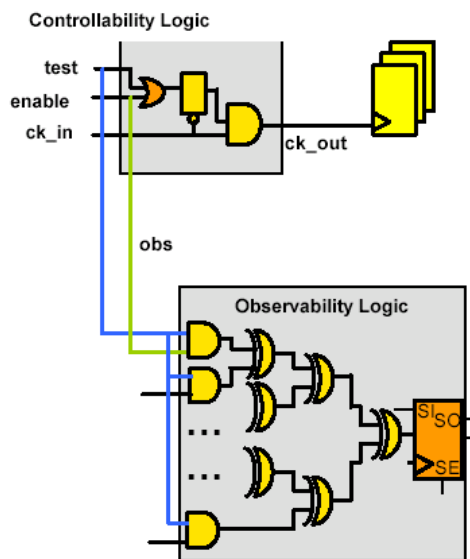
Design for Test (DFT) is also important in low-power design. To increase test coverage, ensure that the clock-gating logic inserted by the low-power engine is controllable and observable. First, select a clock-gating cell that contains test control logic, indicating whether the test control logic is located before or after the latch. The figure below depicts the possible location of test control logic.



**Figure 44. Test control logic**

Then, specify the test control signal that must be connected to the test pins of the integrated clock-gating cells, and connect the test signals. There are two scenarios to connect the test pins of the clock-gating logic:

- *Set up observability logic prior to mapping:* If the control signal is specified before synthesis starts, the RC low-power engine can connect the signal to the test enable pin of the clock-gating logic during clock-gating insertion.
- *Insert the observability logic after mapping:* If the control signal is specified after the design is already synthesized, there are commands to connect the test signal at that stage.



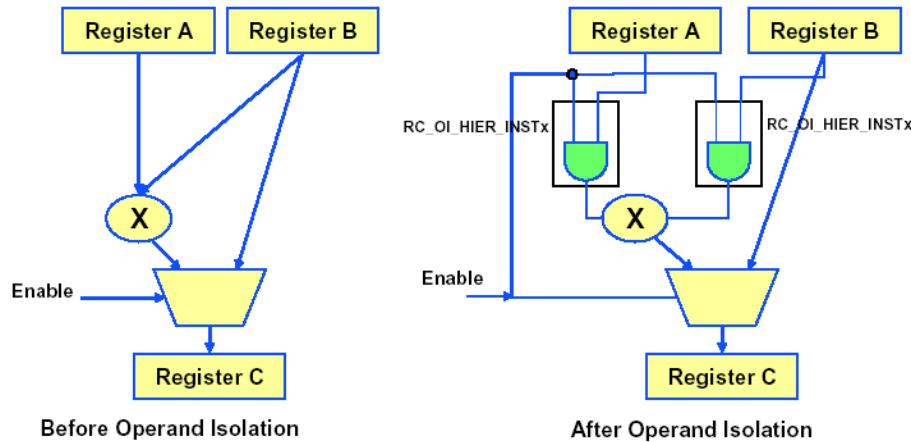
**Figure 45. Controllability and observability logic inserted for DFT**

## Operand Isolation

Operand isolation reduces dynamic power dissipation in datapath blocks controlled by an enable signal. Thus, when enable is inactive, the datapath inputs are disabled so that unnecessary switching power is not wasted in the datapath.

Operand isolation is implemented automatically in synthesis by enabling an attribute before the elaboration of the design. Operand isolation logic is inserted during elaboration, and evaluated and committed during synthesis based on power savings and timing.

Figure 46 illustrates the concept and how it contributes to the power savings.

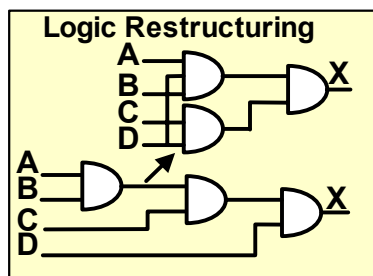


**Figure 46. Operand isolation**

In the digital system shown as Before Operand Isolation, register C uses the result of the multiplier when the enable is on. When the enable is off, register C uses only the result of register B, but the multiplier continues its computations. Because the multiplier dissipates the most power, the total amount of power wasted is quite significant.

One solution to this problem is to shut down (isolate) the function unit (operand) when its results are not used, as shown in After Operand Isolation. The synthesis engine inserts AND gates at the inputs of the multiplier and uses the enable logic of the multiplier to gate the signal transitions. As a result, no dynamic power is dissipated when the result of the multiplier is not needed.

## Logic Restructuring



**Figure 47. Logic restructuring**

A gate-level dynamic power optimization technique, logic restructuring can, for example, reduce three stages to two stages through logic equivalence transformation, so the circuit has less switching and fewer transitions.

## Logic Resizing

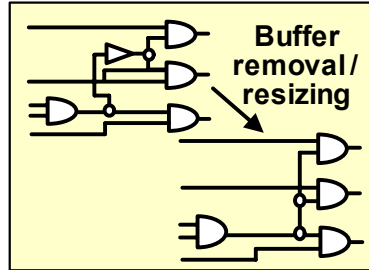


Figure 48. Logic resizing

By removing a buffer to reduce gate counts, logic resizing reduces dynamic power. In the figure, there are also fewer stages; both gate count and stage reduction can reduce power and also, usually, area.

## Transition Rate Buffering

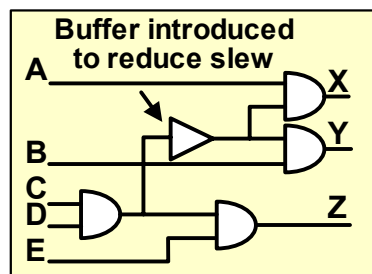


Figure 49. Transition rate buffering

In transition rate buffering, buffer manipulation reduces dynamic power by minimizing switching times.

## Pin Swapping

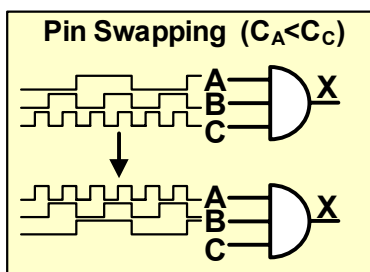


Figure 50. Pin swapping

Figure 50 shows an automated pin-swapping algorithm. The pins are swapped so that most frequently, switching occurs at the pins with lower capacitive load. Since the capacitive load of pin A is lower, there is less power dissipation.

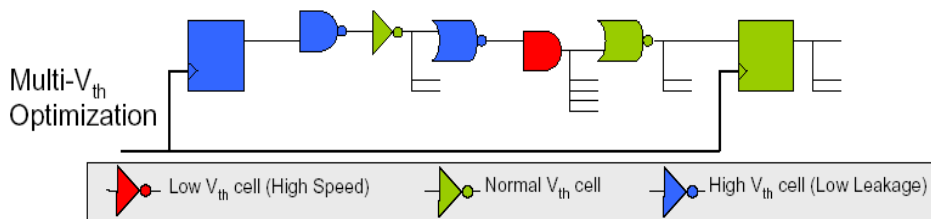
## CPF-Powered Reduction in Synthesis

Now let's talk about how CPF helps in the synthesis stage. CPF, in conjunction with synthesis, enables a variety of sophisticated power reduction techniques, including MSV, DVFS, and PSO. These techniques can affect both active and, predominantly, leakage power.

### Multi- $V_{th}$

The most common leakage reduction technique is to use specially designed high- $V_{th}$  cells where possible in the netlist. The low- $V_{th}$  gates switch more quickly in response to their input signals, but consume more leakage power. The high- $V_{th}$  gates switch more slowly, but consume less leakage power.

The synthesis tool should be able to limit the maximum leakage power for the design by performing multi- $V_{th}$  leakage optimization. The compiler chooses cells with high  $V_{th}$  to replace the cells with low  $V_{th}$  in areas where it won't affect critical timing paths. Low- $V_{th}$  cells are placed in areas that do not meet timing.



**Figure 51. Multi- $V_{th}$  optimization**

CPF also has the capability to include multi- $V_{th}$  libraries:

```
define_library_set -name Vlib1 -libraries Lib1
define_library_set -name Vlib2 -libraries {Lib2 Lib3}
define_library_set -name Vlib3 -libraries Lib4
```

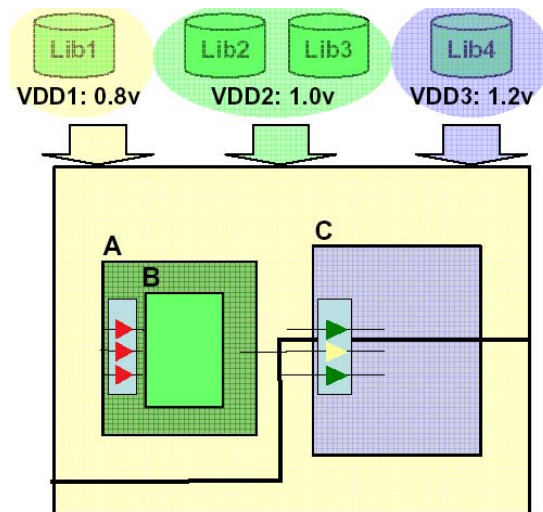
### Multi-Supply Voltage (MSV) Design

Multi-supply voltage techniques can reduce power consumption of SoCs that do not require all blocks to operate at maximum speeds at all times. Designers use different supply voltages for different blocks of the chip based on their performance requirements. MSV implementation is key to reducing power, since lowering the voltage has a squared effect on active power consumption.

Top-down MSV synthesis features include the following:

- Multiple voltage domains
- Assign libraries to domains
- Assign blocks to domains
- Top-down analysis and optimization
- Level shifter insertion

Synthesis uses the power domain concept to describe switchable blocks with different supply voltages. Level shifters are added to ensure that blocks operating at different voltages will operate correctly when integrated together in the SoC. Level shifters must ensure the proper drive strength and accurate timing as signals transition from one voltage level to another. A power domain is a collection of design blocks or instances that share the same supply voltage. Figure 52 illustrates how libraries and design blocks are associated with power domains.



**Figure 52. MSV synthesis**

The following steps describe how to create power domains in CPF and perform MSV synthesis.

First, in CPF:

- Define power domains VDD1, VDD2, and VDD3 in CPF
- Define the nominal conditions and assign the technology libraries to the nominal conditions
- Define the power mode and attach the nominal conditions to the power domains
- Associate the operating corners with the power modes
- Create level shifter rules

Then, during synthesis:

- Read the design RTL



- Elaborate the design
- Read in the CPF file

The design (Figure 52) has three sub-designs: A, B, and C. The low-power constraints used to drive the synthesis tool, in conjunction with the RTL, are in the TOP.cpf file.

Below is sample CPF syntax for TOP.cpf:

```
create_power_domain -name VDD1 -default
update_power_domain -name VDD1 -internal_power_net VDD1
create_power_domain -name VDD2 -instances B
update_power_domain -name VDD2 -internal_power_net VDD2
create_power_domain -name VDD3 -instances
update_power_domain -name VDD3 -internal_power_net VDD3
create_level_shifter_rule -name LVLH2L -from VDD1 -to VDD2
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -from VDD1 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -from VDD2 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to

define_library_set -name Vlib1 -libraries Lib1
define_library_set -name Vlib2 -libraries {Lib2 Lib3}
define_library_set -name Vlib3 -libraries Lib4

create_operating_corner -name corner1 -voltage 0.80 -process 1 -temperature 125 -library_set Vlib1
create_operating_corner -name corner2 -voltage 1.0 -process 1 -temperature 125 -library_set Vlib2
create_operating_corner -name corner3 -voltage 1.2 -process 1 -temperature 125 -library_set Vlib3

create_nominal_condition -name Vdd_low -voltage 0.8
update_nominal_condition -name Vdd_low -library_set Vlib1
create_nominal_condition -name Vdd_mid -voltage 1.0
update_nominal_condition -name Vdd_mid -library_set Vlib2
create_nominal_condition -name Vdd_hi -voltage 1.2
update_nominal_condition -name Vdd_hi -library_set Vlib3

create_power_mode -name PM_base -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_hi} -default
update_power_mode -name PM_base -sdc_files top.sdc

create_analysis_view -name base_view -mode PM_base -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner3}
...
```

## ***Dynamic Voltage Frequency Scaling (DVFS) Synthesis***

Dynamic voltage frequency scaling is another advanced key to power reduction, since lowering the voltage has a squared effect on active power consumption. The DVFS technique provides ways to reduce power consumption of chips “on the fly,” or dynamically, by scaling the voltages and clock frequencies based on the performance requirements of the application.

### **Top-down DVFS Synthesis**

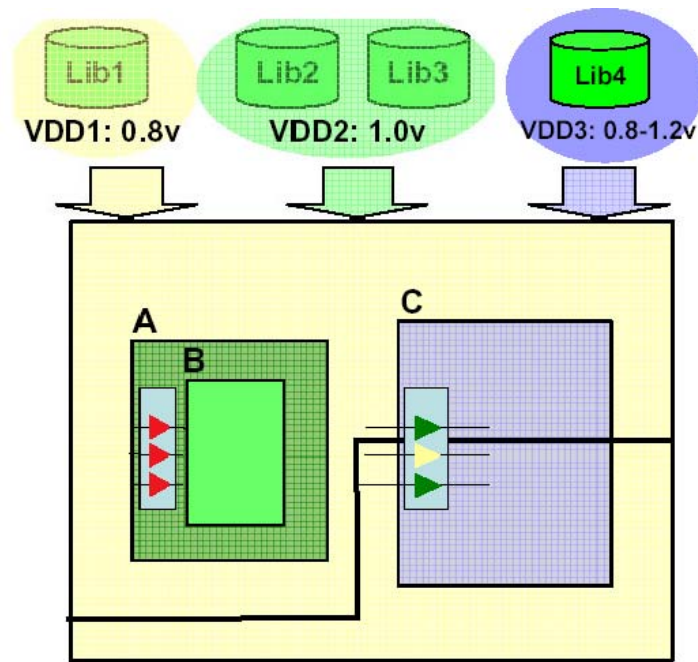
DVFS features and methodology include the following:

- Multiple voltage domains (variable power supply)
- Assign libraries to domains
- Assign blocks to domains
- Top-down analysis and optimization
- Level shifter insertion

To reduce the total power consumption of the chip, the design uses variable supply voltages for different parts of the chip based on their performance requirements. Here are the requirements for DVFS:

- Variable power supply
- Capable of generating required voltage levels
- Minimal transition energy losses
- Quick voltage-transient response
- Voltage scaling
- Scale the frequency in the same proportion to meet signal propagation delay requirements
- Power scheduler that intelligently computes the appropriate frequency and voltage levels needed to execute the various applications (tasks or jobs)

The synthesis tool uses a multimode, multi-corner concept to describe and optimize the variable power domains. In Figure 53, the voltage of the VDD3 power domain can scale in the range of 0.8V to 1.2V.



**Figure 53. DVFS synthesis**

The following steps describe how to create power domains in CPF and perform DVFS synthesis.

First, in CPF:

- Define power domains VDD1, VDD2, and VDD3
- Define the nominal conditions and assign the technology libraries to the nominal conditions
- Define the power mode and attach the nominal conditions to the power domains.
- Define the MMMC condition by using the analysis view to associate the operating corners with the power modes
- Define level shifter rules

Then, during synthesis:

- Read the design RTL
- Elaborate the design
- Read in the CPF file

Following is sample CPF syntax for TOP.cpf:

```
create_power_domain -name VDD1 -default
update_power_domain -name VDD1 -internal_power_net VDD1
create_power_domain -name VDD2 -instances B
update_power_domain -name VDD2 -internal_power_net VDD2
create_power_domain -name VDD3 -instances
update_power_domain -name VDD3 -internal_power_net VDD3
create_level_shifter_rule -name LVLH2L -fromVDD1 -to VDD2
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -fromVDD1 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -fromVDD2 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to

define_library_set -name Vlib1 -libraries Lib1
define_library_set -name Vlib2 -libraries {Lib2 Lib3}
define_library_set -name Vlib3 -libraries Lib4

create_operating_corner -name corner1 -voltage 0.80 -process 1 -temperature 125 -library_set Vlib1
create_operating_corner -name corner2 -voltage 1.0 -process 1 -temperature 125 -library_set Vlib2
create_operating_corner -name corner3 -voltage 1.2 -process 1 -temperature 125 -library_set Vlib3

create_nominal_condition -name Vdd_low -voltage 0.8
update_nominal_condition -name Vdd_low -library_set Vlib1
create_nominal_condition -name Vdd_mid -voltage 1.0
update_nominal_condition -name Vdd_mid -library_set Vlib2
create_nominal_condition -name Vdd_hi -voltage 1.2
update_nominal_condition -name Vdd_hi -library_set Vlib3

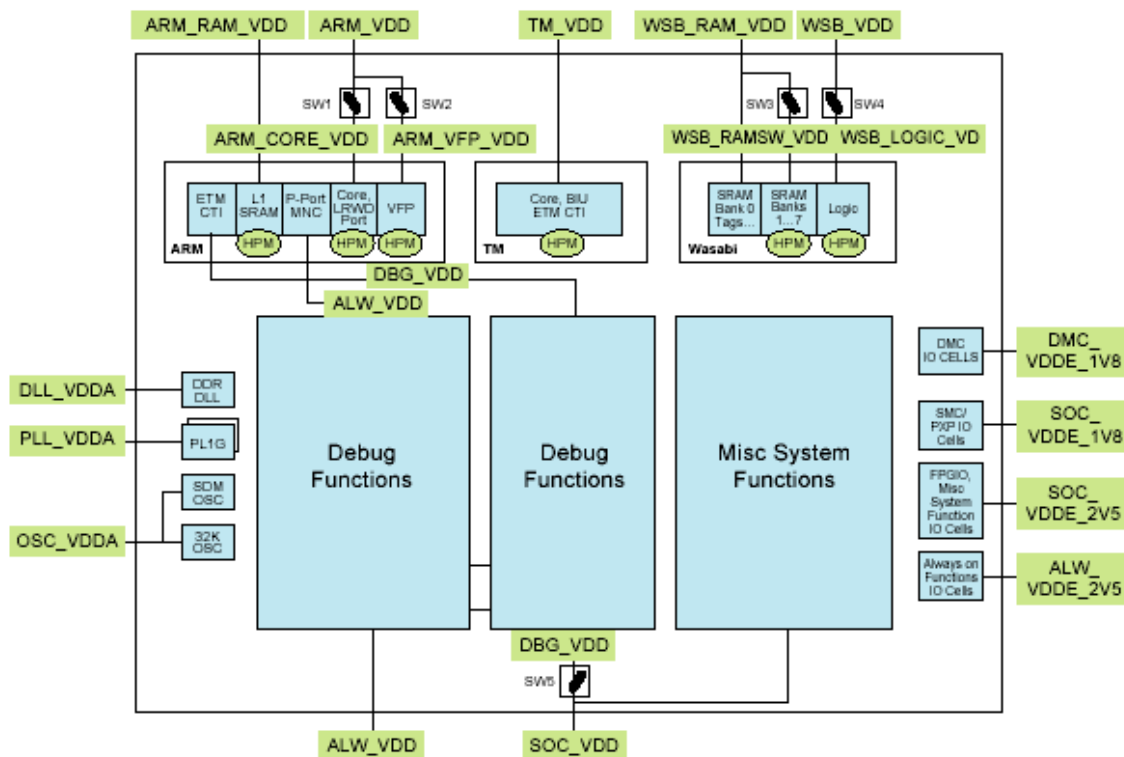
create_power_mode -name PM_base -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_hi} -default
update_power_mode -name PM_base -sdc_files top.sdc
create_power_mode -name PM_scale1 -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_low}
update_power_mode -name PM_scale1 -sdc_files top2.sdc
create_power_mode -name PM_scale2 -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_mid}
update_power_mode -name PM_scale2 -sdc_files top3.sdc

create_analysis_view -name base_view -mode PM_base -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner3}
create_analysis_view -name base_view -mode PM_scale1 -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner1}
create_analysis_view -name base_view -mode PM_scale2 -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner3}
...
```

DVFS is currently used in modern processors such as Intel's XScale, Transmeta's Crusoe, AMD's mobile K6 plus, and ARM 1176.

NXP shared its CPF design experience at CDNLive 2007 [\(Ref. 11\)](#). The NXP platform design is a complex SoC, leveraging a reusable low-power specification, and was implemented using CPF and a CPF-enabled tool flow from RTL to GDS.

As shown in Figure 54, the SoC consists of 11 islands, with 3 voltage scalable logic sections, 3 on-chip switchable domains, 5 off-chip switchable domains, and separate switchable pad ring sections. The three blocks consuming the most power (RISC CPU, VLIW DSP, and L2 System Cache) are controlled using DVFS.



**Figure 54. NXP Platform block diagram. Courtesy NXP, CDN Live 2007 [\(Ref. 11\)](#)**

Because of DVFS power reduction, the number of modes increased, since an “active” block may mean a range of operating voltages and therefore a large number of corners. Associating analysis views to each power mode gave NXP the ability to manage the different constraints and library associated with each operating condition of each power domain for each mode.

NXP successfully taped out this platform SoC on the first pass. It realizes that CPF is a way for low-power design to move from handcrafting to automation and improve turnaround time. Using design tools that understand a common power design intent, with the highest possible level of abstraction, can help compensate

for the increased complexity introduced by designing with multiple supplies, DVFS, and other advanced power management techniques.

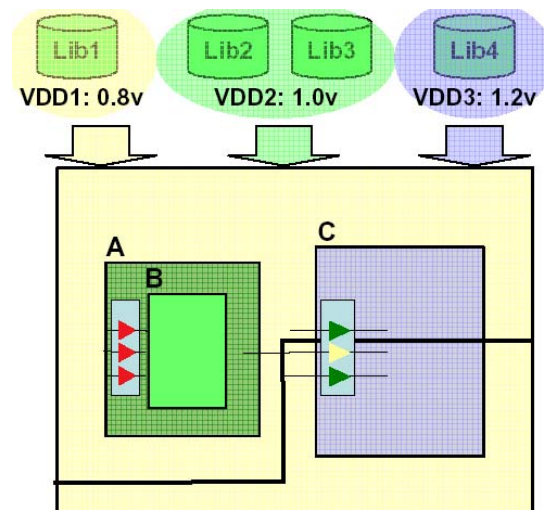
### **Power Shutoff (PSO)**

Power shutoff is the single most efficient way to reduce leakage power. If a block is not used, it is powered down, greatly reducing power.

Synthesis uses the power domain concept to describe switchable blocks (switchable power domain) and always-on portions of the design (always-on power domain). Isolation cells are needed to prevent the unwanted propagation of signals from power-down domain to power-on domains.

The main task of synthesis is adding isolation cells automatically based on CPF. Otherwise, synthesis is not largely affected by PSO unless it needs to insert state retention cells and/or always-on cells. The connection of power switch cells to the power control module happens during the physical implementation flow, when physical information is known.

In Figure 55, there are three power domains. VDD2 is the defined name of block B, the switchable power domain.



**Figure 55. Power shutoff for block B**

The following is a CPF command file showing PSO for block B (called VDD2), including isolation and shutoff:

```

create_power_domain -name VDD1 -default
update_power_domain -name VDD1 -internal_power_net VDD1
create_power_domain -name VDD2 -instances B -shutoff_condition {PSO_EN}
update_power_domain -name VDD2 -internal_power_net VDD2
create_power_domain -name VDD3 -instances
update_power_domain -name VDD3 -internal_power_net VDD3
create_level_shifter_rule -name LVLH2L -from VDD1 -to VDD2
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -from VDD1 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -from VDD2 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_isolation_rule -name ISOLH2H -from VDD2 -to VDD3 -isolation_condition "isoenable"-
isolation_output low
update_isolation_rules -names ISOLH2H -cells ISOLS2 -combine_level_shifting -location to
...

define_library_set -name Vlib1 -libraries Lib1
define_library_set -name Vlib2 -libraries {Lib2 Lib3}
define_library_set -name Vlib3 -libraries Lib4

create_operating_corner -name corner1 -voltage 0.80 -process 1 -temperature 125 -library_set Vlib1
create_operating_corner -name corner2 -voltage 1.0 -process 1 -temperature 125 -library_set Vlib2
create_operating_corner -name corner3 -voltage 1.2 -process 1 -temperature 125 -library_set Vlib3

create_nominal_condition -name Vdd_low -voltage 0.8
update_nominal_condition -name Vdd_low -library_set Vlib1
create_nominal_condition -name Vdd_mid -voltage 1.0
update_nominal_condition -name Vdd_mid -library_set Vlib2
create_nominal_condition -name Vdd_hi -voltage 1.2
update_nominal_condition -name Vdd_hi -library_set Vlib3

create_power_mode -name PM_base -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_hi} -default
update_power_mode -name PM_base -sdc_files top.sdc

create_analysis_view -name base_view -mode PM_base -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner3}
create_analysis_view -name off_view -mode PM_base -domain_corners {VDD1@corner1
VDD3@corner3}
...

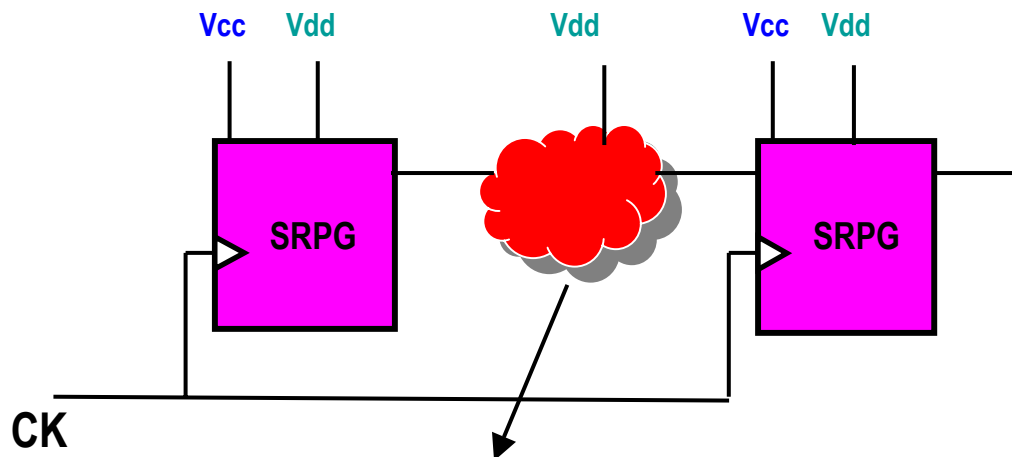
```



## State Retention Power Gating (SRPG)

The use of power-gating state retention cells allows a system to shut down power to certain block(s) in a design, and recover the prior states after a power-up sequence.

To implement power gating, special state retention cells are required to store prior state(s) of the blocks before power-down. The basic flip-flop has been modified in SRPG, and the master latch runs on the same power supply  $V_{dd}$  as combinational logic, while the slave latch runs on the different power supply  $V_{cc}$ . The state of the system will be retained in the flip-flops during power down and all the combinational logic will be turned off during sleep mode.



*Figure 56. SRPG: huge leakage power*

The advantages of SRPG include shutdown leakage savings, which can be independent of process variations. It allows for faster system power-on because the state is preserved in the slave latch.

Disadvantages include increased area and die size; timing penalties such as increased signal and clocking delays; increased routing resources (power routing for  $V_{cc}$  and a power-gating signal tree with on buffers); specialized library models for SRPG cells; additional power overhead in the active mode; and impacts to functional verification, physical integration, and DFT.

Following is a sample CPF syntax for power shutoff with state retention (TOP.cpf):

```

create_power_domain -name VDD1 -default
update_power_domain -name VDD1 -internal_power_net VDD1
create_power_domain -name VDD2 -instances B
update_power_domain -name VDD2 -internal_power_net VDD2
create_power_domain -name VDD3 -instances
update_power_domain -name VDD3 -internal_power_net VDD3
create_level_shifter_rule -name LVLH2L -fromVDD1 -to VDD2
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -fromVDD1 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to
create_level_shifter_rule -name LVLH2L -fromVDD2 -to VDD3
update_level_shifter_rules -names LVLH2L -cells LS12 -location to

create_state_retention_rule -name SRPG1 -domain VDD2 -restore_edge {B_RESTORE_EN}
update_state_retention_rule -names SRPG1 -library_set Vlib2 -cell_type DRFF
...

define_library_set -name Vlib1 -libraries Lib1
define_library_set -name Vlib2 -libraries {Lib2 Lib3}
define_library_set -name Vlib3 -libraries Lib4

create_operating_corner -name corner1 -voltage 0.80 -process 1 -temperature 125 -library_set Vlib1
create_operating_corner -name corner2 -voltage 1.0 -process 1 -temperature 125 -library_set Vlib2
create_operating_corner -name corner3 -voltage 1.2 -process 1 -temperature 125 -library_set Vlib3

create_nominal_condition -name Vdd_low -voltage 0.8
update_nominal_condition -name Vdd_low -library_set Vlib1
create_nominal_condition -name Vdd_mid -voltage 1.0
update_nominal_condition -name Vdd_mid -library_set Vlib2
create_nominal_condition -name Vdd_hi -voltage 1.2
update_nominal_condition -name Vdd_hi -library_set Vlib3

create_power_mode -name PM_base -domain_conditions {VDD1@Vdd_low VDD2@Vdd_mid
VDD3@Vdd_hi} -default
update_power_mode -name PM_base -sdc_files top.sdc

create_analysis_view -name base_view -mode PM_base -domain_corners {VDD1@corner1
VDD2@corner2 VDD3@corner3}
...

```

## Simulation for Power Estimation

Power consumption is dependent on both the physical structures on the chip and the mode of operation. With today's multimode SoCs, determining the correct stimulus to verify average and peak power across a variety of modes is increasingly challenging.

Generally, designers will want to obtain early estimates of power based on available stimulus. For more accurate power estimates, switching activity data is obtained by simulating test cases with real system stimulus. Often, such simulation

is not available until later in the design cycle. Designers need to use the most accurate test bench available at any given point in the design flow and revise their estimate as new stimulus becomes available. If switching activity data is not available from simulation, designers should estimate the switching activity on the chip's primary inputs and apply that estimate within the power analysis tool. Transient switching power can be estimated based on the number of flip-flops, combinatorial gates, and clock speed. By default, RTL Compiler estimates dynamic power using some default switching activity values.

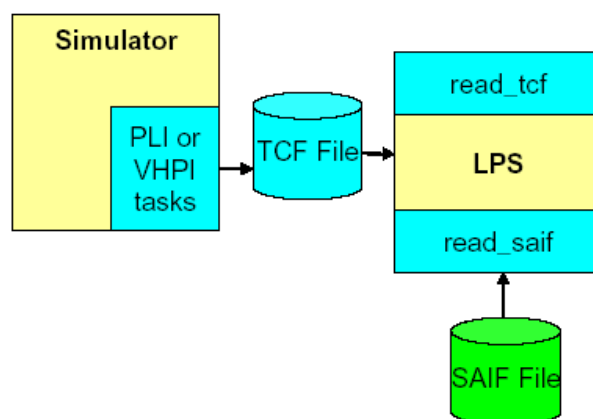
Annotate switching activity using accurate switching activity data when available. To get a more accurate estimate, run simulation of the final netlist to generate a switching activity file in one of the standard formats.

Simulation tools support the switching activity information needed for power optimization and power analysis. This information needs to be provided before running generic synthesis. Switching activity can be annotated into the compiler by loading a .vcd, .saif, or .tcf file.

The toggle count format (.tcf) file contains switching activity in the form of the toggle count information and the probability of the net or pin to be in the logic 1 state. Synthesis tools propagate the switching activities throughout the design.

The functional simulations are Verilog or VHDL simulations. The functional simulation is carried out to generate the toggle count format file (.tcf, .saif, or switching activity) by running the test bench on the RTL or synthesized gate-level netlist.

The .tcf generated by running simulation on the RTL is used as an input for accurate power analysis in synthesis.



**Figure 57. Generating a .tcf file using PLI tasks**

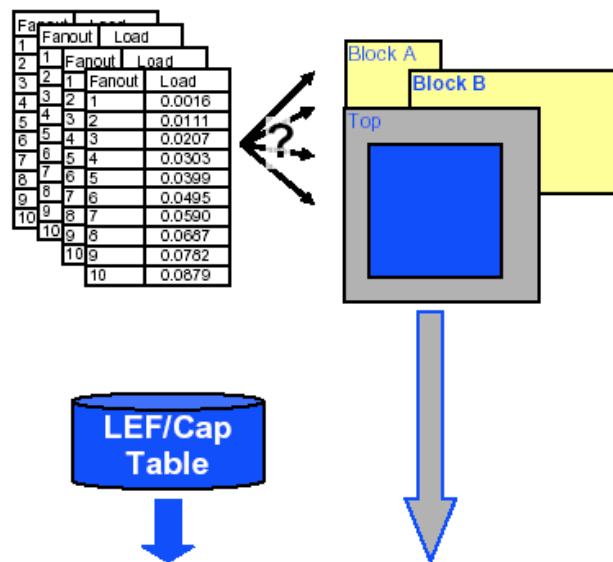
Also, consider the simulation mode when generating switching activities. A zero-delay gate-level simulation will not account for any natural glitching that occurs in

combinatorial logic, and will result in an optimistic power calculation. If gate-level simulation is required for power analysis, use an SDF delay-based gate-level simulation.

Use libraries that represent the worst-case power. Synthesis is done using worst-case timing libraries to optimize for area, timing, and power concurrently, but they do not necessarily represent the worst-case power. Dynamic power is usually the highest in fast conditions, which can be represented by the best-case timing libraries.

Use accurate wire modeling. Every designer knows about the inaccuracies of wire load models when it comes to timing closure. Yet, many design teams use a “zero” wire load model for synthesis, resulting in inaccurate power estimation. Use a reasonable wire load model or one of the “physical based” wire-modeling technologies available in today’s synthesis tools.

Physical layout estimation is a physical modeling technique that bypasses wire loads for RTL synthesis optimization. In place of wire loads, the compiler generates an equation to model the wire delay. Physical layout estimation uses actual design and physical library information and dynamically calculates wire delays for different logic structures in the design. In most cases, physical layout estimation–synthesized designs correlate better with place and route tools.



**Figure 58. Synthesis data flow**

## CFP Synthesis Summary

Many power optimization techniques are handled automatically during synthesis with today's tools. However, the more advanced and powerful techniques are automated by a combination of synthesis and CPF. Exploring power at the architectural level and during synthesis using CPF files, independent of the RTL design, provides the highest leverage and percentage of power reduction possible in the SoC design to implementation flow.

# Low-Power Implementation with CPF

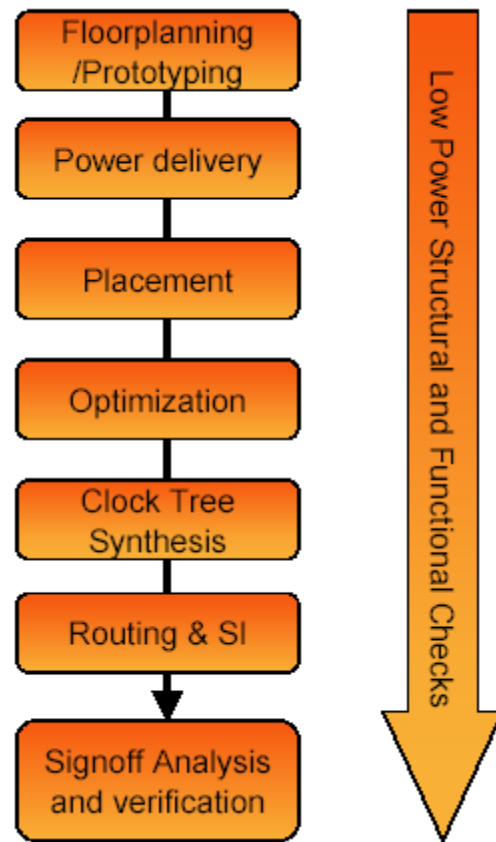
## Introduction to Low-Power Implementation

Low-power implementation must correctly deal with the physical implications and penalties incurred by the wide variety of techniques used for power optimization earlier in the flow. In addition, it automatically performs power optimizations that can only be accurately employed once the physical layout is understood.

### *Implementation Stages*

Low-power implementation consists of multiple steps, automated through CPF and the electronic design automation tool flow:

- Floor planning with multiple power domains
- Power delivery, through power planning and routing
- Insertion of power gating for low-power shutoff
- Placement, including placement of level shifter, isolation, and SRPG cells
- Optimization, including multiple threshold voltage (Multi- $V_{th}$ ) optimization, as well as multiple supply voltage (MSV) optimization
- Clock tree synthesis, and ensuring the clock tree is well balanced and optimized for power
- Efficient routing, because the shorter the route length, the less power is dissipated, while timing and signal integrity must be preserved
- Analysis and verification, or signoff power analysis, to make sure power consumption is consistent with estimation, and that timing and IR drop are under control



**Figure 59. Low-power implementation flow**

The low-power techniques that have an especially high impact on implementation complexity, as previously discussed, include:

- Gate-level optimizations—logic resizing, restructuring, and pin swapping
- Clock gating
- Multi- $V_{th}$  optimization
- MSV
- Power shutoff (including state retention cell usage)
- DVFS
- Back biasing

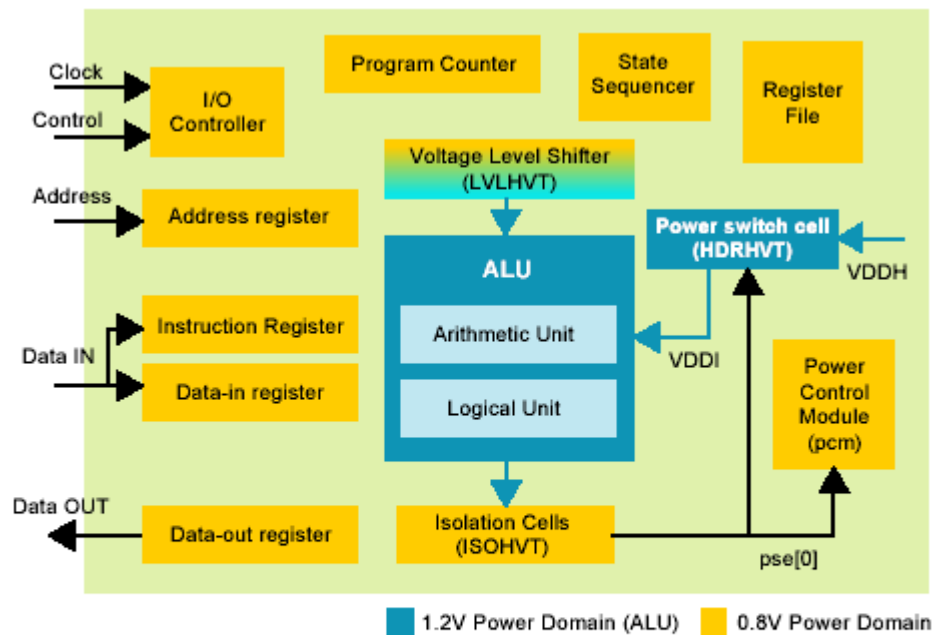
### ***Critical Challenges of Low-Power Implementation***

The success of the SoC design depends on a physical implementation that obeys the consistent power intent from front-end design and verification. Power intent in this case refers to the implementation of power domains according to definitions, isolation/level shifter cell usage, etc. For example, in the front end, which power domains belong to which hierarchical instances is established. This has to be



maintained consistently between front-end design and implementation, which has special impact on the cell identification, place, route, and verification tasks.

The following shows a multi-block design with ALU, I/O, address, instruction and data registers, a state sequencer, and on-chip power control module. It represents an application of multiple power domains (0.8–1.2v) for power optimization, with the concomitant level shifters and isolation cells.



**Power Intent Specification:**

1. ALU is a switchable domain controlled by pse[0] signal from Power Control Module (pcm). When it is ON, it runs at 1.2V
2. The ALU output needs to be isolated when ALU is shut down. The inputs to ALU needs to be shifted in voltage.
3. There is no state saving inside ALU.

**Figure 60. Illustration of power intent**

The CPF for the above design follows:

```

set_design cpu
### Power net definitions ###
create_power_nets -nets VDD -voltage 0.8
create_power_nets -nets VDDH -voltage 1.2
create_ground_nets -nets VSS
create_global_connection -domain ALUP \
-net VDDI -pins VDD
### Low power cell definitions ###
define_power_switch_cell -cells HDRHVT \
-stage_1_enable SLPIN -stage_1_output SLPOUT \
-power VDDH -power_switchable VDDI
define_isolation_cell -cells ISOHVT \
-enable NSLEEP -power VDD -ground VSS
define_level_shifter_cell \
-cells LVLHVT -valid_location from \
-input_voltage_range 0.8 \
-output_voltage_range 1.2 -ground VSS \
-input_power_pin VDD \
-output_power_pin VDDH
### Power domain definitions ###
create_power_domain -name TOP -default
create_power_domain -name ALUP \
-instances ALU \
-shutoff_condition {pcm_inst/pse[0]}
update_power_domain -name ALUP \
-internal_power_net VDDH

### Low power cell creation directives ###
create_power_switch_rule -name PSW_RULE -domain ALUP
create_isolation_rule -name ISO_RULE -from ALUP
-isolation_condition {pcm_inst/pse[0]} -isolation_output high

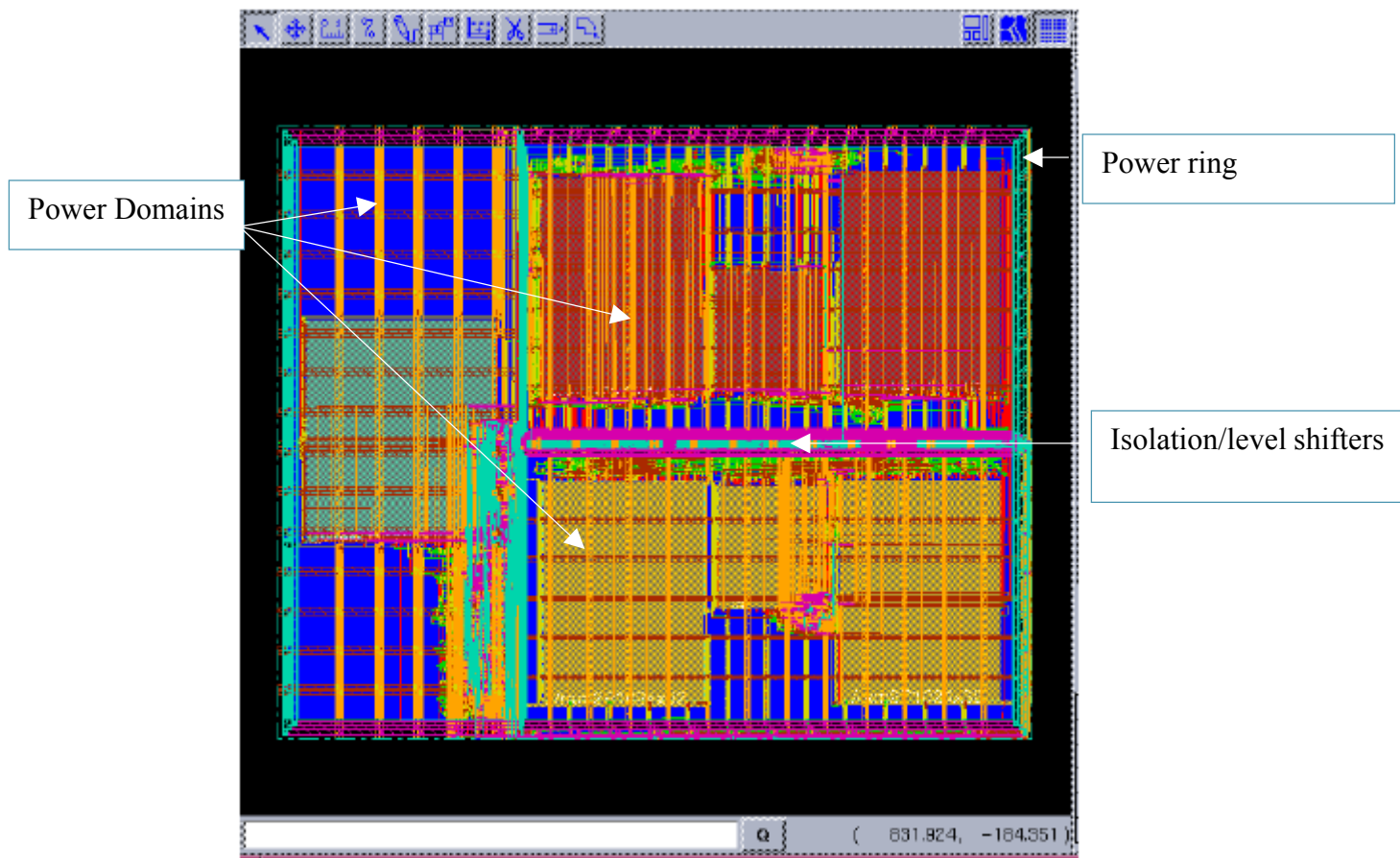
create_level_shifter_rule -name LS_RULE -from TOP -to ALUP

```

The physical designer faces many challenging physical realities when implementing low-power constructs defined in front-end design. For example, how many power switches are needed in order to prevent IR drop from causing timing problems or a catastrophic failure? Is current density an issue on the SoC?

This is also the final opportunity to juggle and optimize timing, area, and power requirements. In the implementation stage, timing, area, and power are translated to physical reality. Cells are now known to have actual placement area; routes have real lengths with associated RLC. Therefore, meeting timing, area, and power requirements becomes a hard requirement, which is an iterative process.

Figure 61 is a snapshot of the SoC physical placement and routing layout from the example above, showing MSV techniques implemented in 65nm.



*Figure 61. Automated power planning in physical placement and routing*

## Gate-Level Optimization in Power-Aware Physical Synthesis

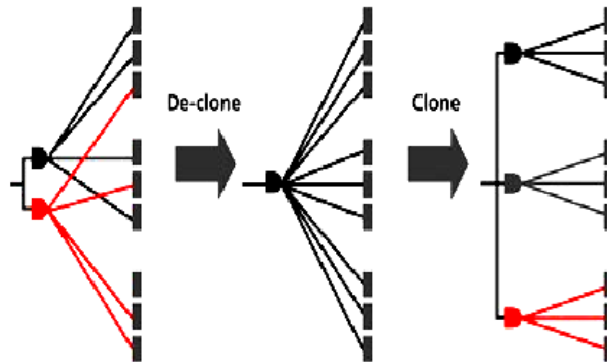
Perhaps the most basic of low-power techniques in the implementation stage is gate-level optimization. This set of techniques includes transistor resizing, restructuring, and pin swapping. These techniques are not unlike those being used in the synthesis stage; the only difference is that in the implementation stage, the designer and the implementation tool have exact knowledge of physical distance and routing distance between cells. This allows more accurate application of resizing, restructuring, and pin swapping for maximum benefit while incurring minimum timing penalty.

## Clock Gating in Power-Aware Physical Synthesis

Today, clock gating to address dynamic power is done in almost all designs, not just low-power designs. The reason is that clock-gating technology in EDA tools has evolved to where it is automated and easy to implement, and doesn't break the methodology.

Clock gating is first defined in the synthesis stage, as discussed in the Low-Power Design chapter, and then optimized in the implementation stage.

In the synthesis stage, clock-gating elements are inserted; however, in the synthesis stage there usually is no exact information on the physical distance between the clock-gating element and the leaf cell. Clock-gating violations usually occur because the clock-gating cell is too far from the leaf cell. During physical implementation, in order to fix clock-gating violations, the clock-gating cell must be physically moved closer to the leaf cell. However, if the clock-gating cells are completely de-cloned, this isn't possible until clock-gating cloning is done.



**Figure 62. Clock cloning/de-cloning**

Conversely, overdoing clock-gating cloning will introduce many clock-gating elements, thereby nullifying the power and area advantage provided by clock gating. The designer is caught between Scylla and Charybdis!

However, in the physical realm, the implementation tool now knows exactly how far the clock-gating cell is from the leaf pin. This enables the tool to correctly clone the clock-gating element to prevent clock-gating timing violations.

Therefore, the correct methodology to deal with clock gating is to de-clone all the way during synthesis, and then selectively clone based on clock-gating timing during the physical implementation stage. This is a process that is automated by the EDA tool during the clock tree synthesis implementation stage.

## **Multi- $V_{th}$ Optimization in Power-Aware Physical Synthesis**

Multi- $V_{th}$  optimization, which addresses leakage power, is also widely used in today's physical implementation designs. Current EDA technology has matured so that multi- $V_{th}$  optimization is automated from RTL through GDS. Basic requirements are different threshold voltage libraries of the same cell's functionality, and a power-aware implementation tool. High- $V_{th}$  cells are low power, but lower performance as well. Low- $V_{th}$  cells consume higher power, but provide

higher performance. Usually the trade-off favors power. For example, by using a high- $V_{th}$  cell instead of low- $V_{th}$  cell, the user can achieve a significant reduction (up to 80 percent) in leakage power with a small impact to timing (around 20 percent).

Different  $V_{th}$  versions of the same functional cell usually have the same footprint, so the cells can be swapped interchangeably and easily during layout. However, the timing impact of using different  $V_{th}$  cells has to be taken into account during cell swapping. The implementation tool also usually handles this analysis automatically.

Multiple threshold voltage swapping usually takes place either in the post clock tree synthesis implementation stage or the post-route stage.

## **Multiple Supply Voltage (MSV) in Power-Aware Physical Synthesis**

MSV implementation is essentially a continuation of MSV synthesis. It is also similar to power shutoff in a number of ways. The tasks involved include:

- Creation of power domains
- Placement and optimization
- Level shifter handling

### ***Creation of Power Domains***

First, during the floor-planning stage, different power domains have to be created, consistent with power domain definitions in the front end. Each power domain has a different set of libraries associated with it for that specific voltage domain, as in the synthesis stage.

### ***Placement and Optimization***

For placement and optimization in a top-down situation where the design is being implemented as a whole, the tool needs to understand that power domain boundaries must be honored. That is, the CPF-aware tool knows that no logic from one power domain can be moved to another power domain.

In addition, during placement and optimization, the tool should be able to use the correct timing libraries set for each of the power domains. For example, when the tool is optimizing the 0.8V power domain, it should use the timing libraries characterized at 0.8V.

Some less-sophisticated implementation tools do not understand the concept of multiple supply voltages, through CPF, and thus MSV design implemented using those tools will need to be implemented bottom-up, which is less efficient and involves more manual engineering effort.

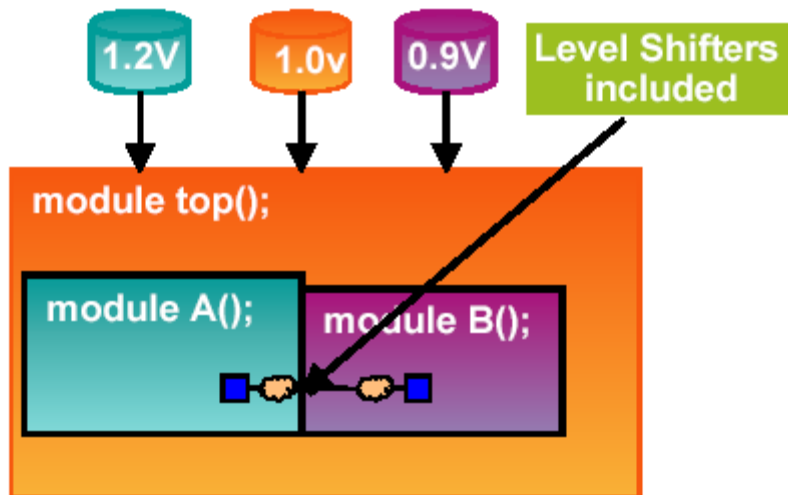


Figure 63. MSV design and level shifters

## Level Shifters

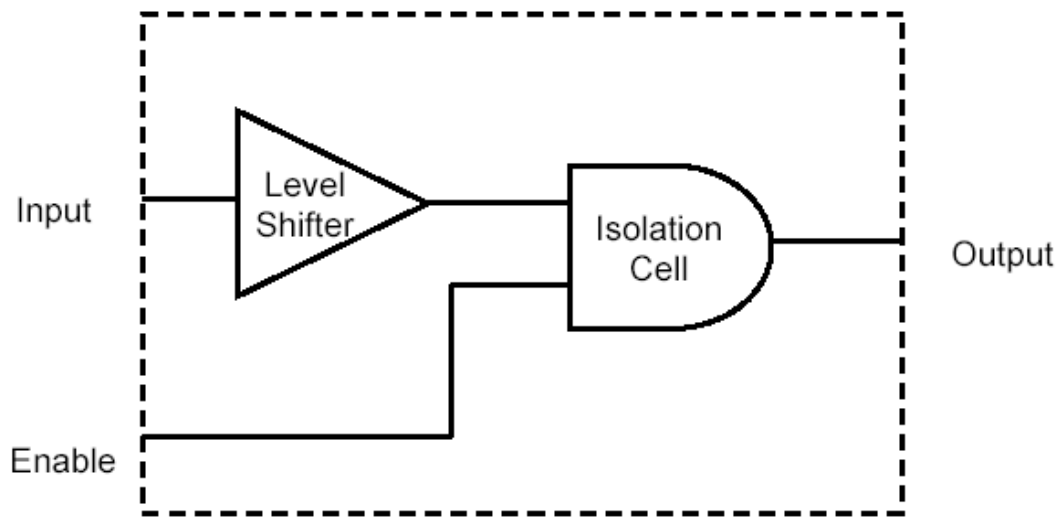
Handling level shifters is another automated task with CPF. Level shifters can be inserted during the synthesis or implementation stage. Every signal that crosses an MSV power domain should have a level shifter attached to it. Although level shifting from a higher-voltage power domain to a lower one is usually optional, level shifting from a lower-voltage power domain to a higher one is mandatory.

A sample CPF for level shifting is shown below:

```
define_level_shifter_cell \
  -cells LVLHVT -valid_location from \
  -input_voltage_range 0.8 \
  -output_voltage_range 1.2 -ground VSS \
  -input_power_pin VDD \
  -output_power_pin VDDH

create_level_shifter_rule -name LS_RULE -from TOP -to ALUP
```

In cases where MSV and PSO are used together, most designers opt for combination level shifter and isolation cells.



**Figure 64. Level shifter/isolation combination cell**

Level shifters are placed in a fashion similar to isolation cells, close to the power domain boundaries. However, level shifters have two power rails:

- *Primary power rail*: usually set at the top and bottom edge of the level shifter
- *Secondary power rail*: usually set at the center horizontal line of the level shifter

The power domain where the level shifter resides depends on which voltage the primary power rail matches. For example, if the primary power rail of the level shifter is a 0.8V rail, that level shifter should be placed in the 0.8V power domain. Therefore, some knowledge about the library is needed in order to decide in which power domain to place the level shifter.

## Challenges in MSV Implementation

### Voltage regulators

One of the main challenges of implementing MSV is the requirement of an on-chip voltage regulator to generate different voltages. A voltage regulator is a complex analog block that generates a different voltage from a given voltage. In some designs, an off-chip voltage regulator may be used, but it is usually done on chip.

### Implications of using lower operating voltages

Theoretically, since power is proportional to voltage squared, by lowering the voltage we should get an exponential decrease in power consumption. In reality, this is not necessarily so, because in the physical world, lower voltage means timing issues and increased transition time, which translates into more power consumption.



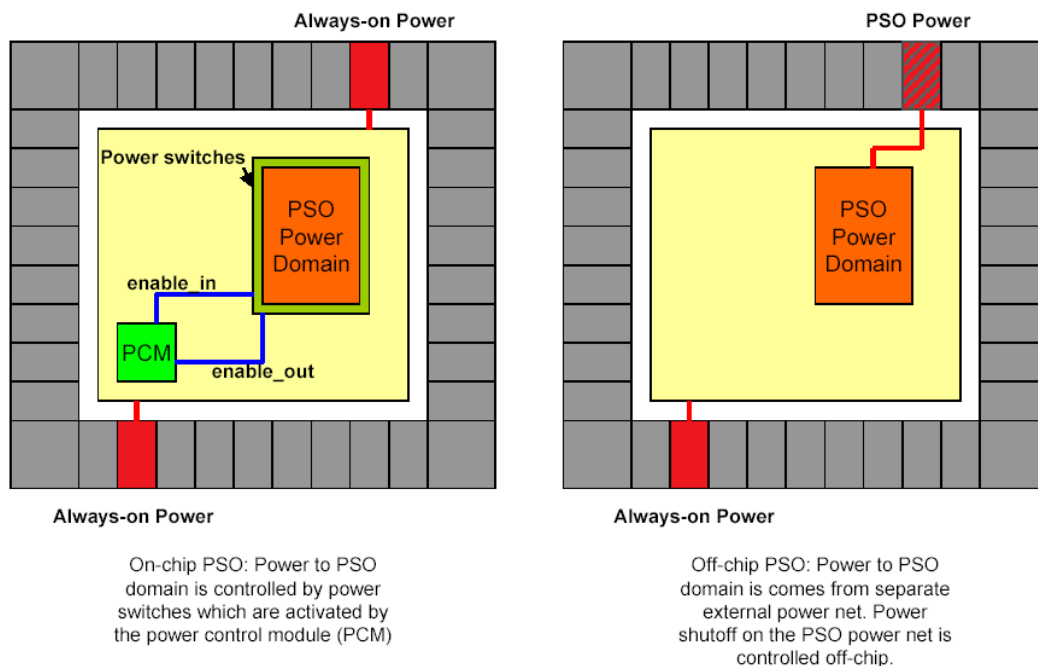
In order to fix timing issues, logic needs to be upsized or inserted, also resulting in more power consumption. Overall, operating at a lower voltage definitely gives power savings, although not as much as theoretically would be possible without reference to timing issues.

## Power Shutoff (PSO) in Power-Aware Physical Synthesis

PSO involves shutting down a part of the chip while the other parts remain functioning, and is a relatively sophisticated low-power technique with many implications for timing and implementation complexity. Nonetheless, PSO is becoming increasingly popular today, not only in mobile electronics but also in tethered electronic systems that are plugged into a power outlet. This is because of the strong low-power benefit and the fact that today's CPF-enabled tools can automate the implementation of PSO with confidence.

Following are the two types of PSO:

- *On-chip power shutoff* means that power switches within the SoC control the power shutoff
- *Off-chip power shutoff* means the power switches are external to the chip



**Figure 65. On-chip PSO vs. off-chip PSO**

PSO (or power gating) can also be either fine- or coarse-grained, referring to the size of each logic block controlled by a single power switch. With fine-grained power gating, power can be shut off to individual blocks or cells without shutting off the power to other blocks—which continue to operate. This can help to reduce active mode leakage power, or leakage during normal operation. With coarse-

grained power gating, power is gated very coarsely, as with a single sleep signal that powers down the entire chip. This reduces leakage only during standby, naturally.

The following table summarizes aspects of each.

	Fine-grained	Coarse-grained
<i>Power gate size</i>	<i>Worst-case switching (30% area)</i>	<i>Actual switching (5% area)</i>
<i>Gate control slew rate</i>	<i>Always-on buffer network</i>	<i>Always-on buffer by abutment</i>
<i>Simultaneous switching capacitance</i>	<i>No issue</i>	<i>Needs to be addressed</i>
<i>Power gate leakage</i>	<i>&gt;30%</i>	<i>&lt;5%</i>

## Physical Implementation Implications of PSO

### Creation of Power Domains

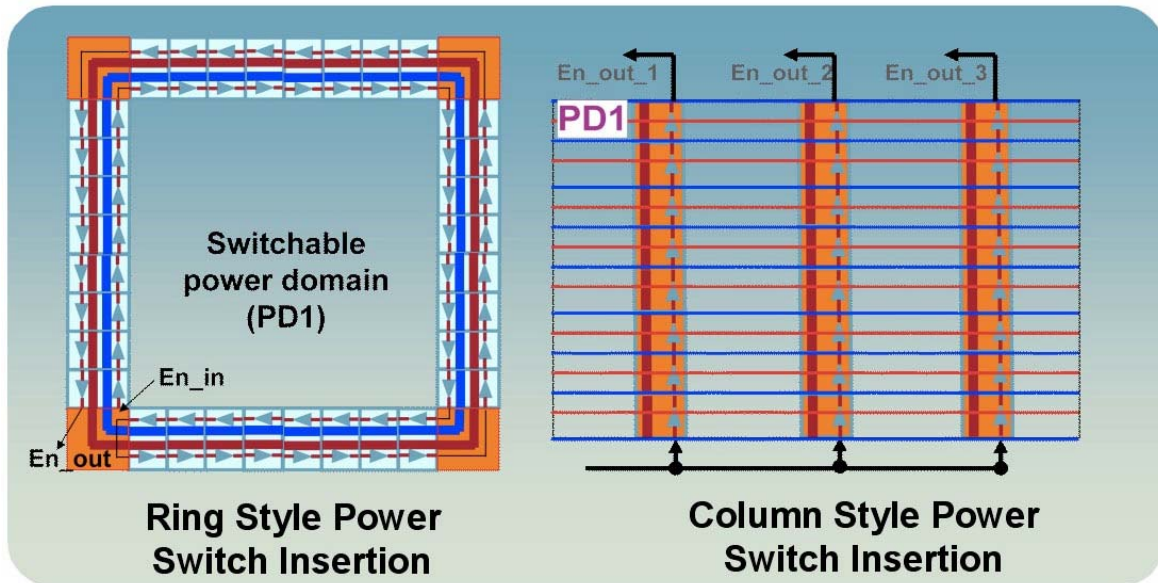
Power domains must be consistent with front-end design power domain definitions. Usually there will be a hierarchical module that is defined as a PSO power domain in the CPF file. This power domain is then implemented such that all the logic or hard macros in the hierarchical module reside in the correct physical area in the power domain, and all the logic or hard macros that don't belong in the hierarchical module reside outside the physical area of the power domain. This is important because the physical area generally defines whether that logic/hard macro is powered by an always-on power net or a PSO power net.

Power domain creation occurs in the floor-planning or physical prototyping stage of the implementation flow. Following is an example of the CPF:

```
create_power_domain -name ALUP \
  -instances ALU \
  -shutoff condition {pcm inst/pse[0]}
```

### Insertion of Power Switch Cells

Insertion of power switch cells (for on-chip PSO) is the next step. Power switch cells can be inserted in a column or a ring fashion.



**Figure 66. Column- versus ring-style power switch insertion**

More advanced, CPF-enabled EDA toolsets will automatically insert the power switch cells for the designer; in less advanced toolsets, the designer has to manually insert these constructs. Power switches are also inserted during the floor planning or prototyping stage of the implementation flow. An example of the CPF for power switches follows:

```
define_power_switch_cell -cells HDRHVT \
    -stage_1_enable SLPIN -stage_1_output SLPOUT \
    -power VDDH -power_switchable VDDI

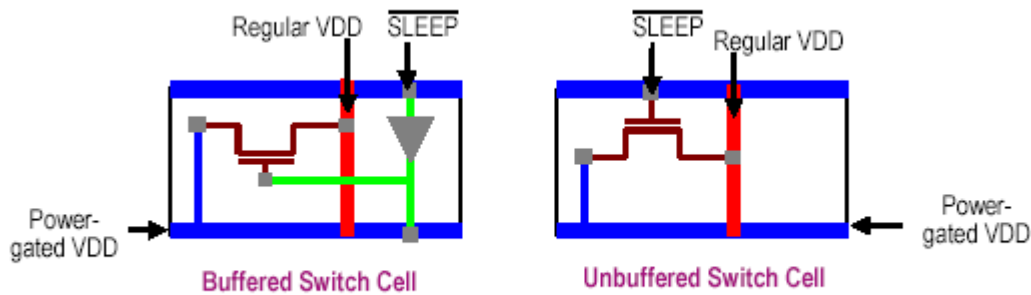
create_power_switch_rule -name \           PSW_RULE -domain ALUP
```

The number and size of the power switches that are inserted depend heavily on the design's physical characteristics. Generally, the larger the PSO power domain area, and the more logic and macros in the PSO power domain area, the more power switches are needed.

The goal is to have the true optimal number of power switches to satisfy IR drop and current density requirements. Too many power switches leads to wasted area, but too few power switches creates excessive IR drop and risks having too much current (rush current) going through each power switch during wakeup.

Some power switches have built-in buffers/delays that accomplish two things: first, control the skew of the enable signal of the power switch; and second, introduce a delay when the enable signal traverses the power switch array.

Figure 67 compares buffered and unbuffered power switches.

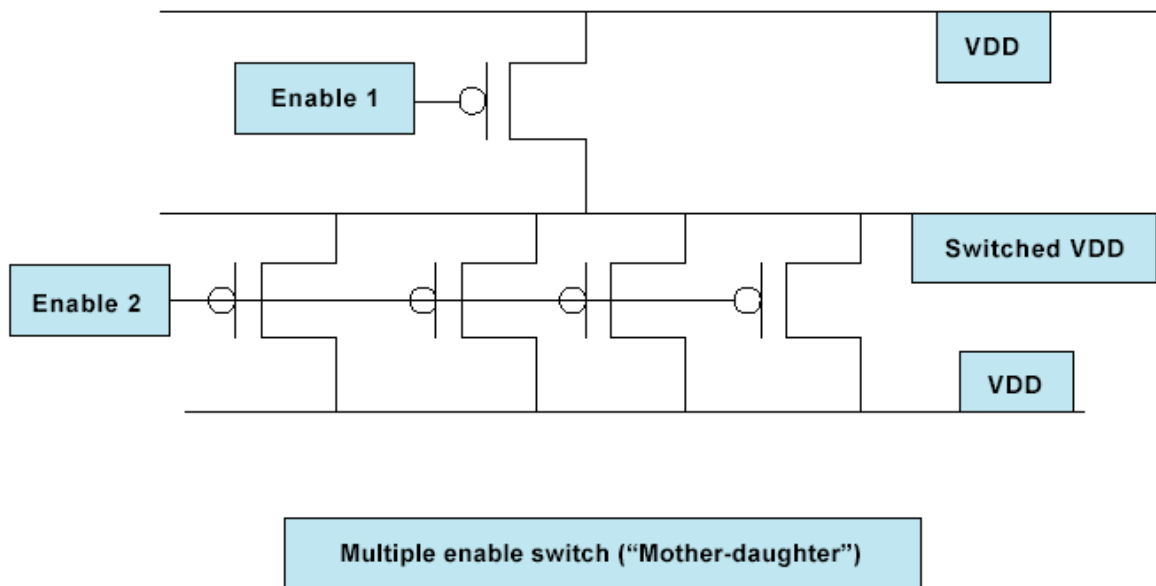


**Figure 67. Buffered vs. unbuffered power switches**

It may be desirable to introduce a delay, because turning on the PSO power domain causes a large current to be drawn by the domain, causing a current spike or rush current. Introducing a delay between the times when each power switch turns on will spread out the turn-on time of the PSO domain, thereby reducing the current spike. Another method for reducing the current spike is to turn on the power within the domain in stages over time.

It is also desirable to design the power switches in groups of cells and turn them on and off one group at a time. This way, the last group of power switches at the end of the shutoff sequence, or the first group of power switches at the beginning of the power-on sequence, will handle the large current instead of a single power switch.

In many designs, switches are used in a configuration called “mother-daughter” pair. These switches have multiple enable pins; typically, the smaller switch is turned on first to get the voltage up to 95 percent, then the bigger switch is turned on to reduce the IR drop. Figure 68 illustrates the configuration of such a switch.



**Figure 68. Mother-daughter pair**

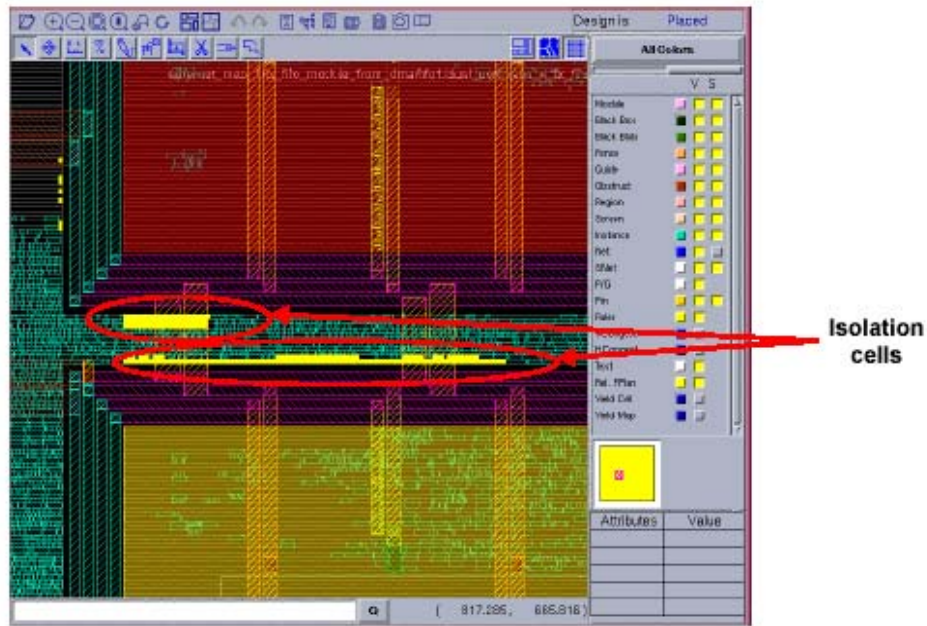
## Isolation Cell Handling

As we have seen earlier, isolation cells can be inserted by the synthesis tool early in the design, if the synthesis tool understands the concept of PSO as it is supported in CPF. The physical implementation tool may also insert isolation cells. Isolation cells should be inserted into the netlist in the early floor-planning stage. Following is a CPF example for isolation cells:

```
define_isolation_cell -cells ISOHVT \
  -enable NSLEEP -power VDD -ground VSS

create_isolation_rule -name ISO_RULE -from ALUP \
  -isolation_condition {pcm_inst/pse[0]} -isolation_output high
```

Isolation cells are placed as close to the PSO domain as possible, but usually reside in the always-on domain. Figure 69 shows this physical layout.



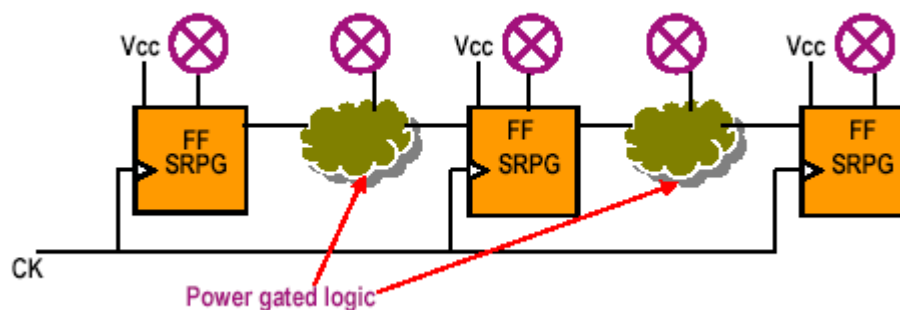
**Figure 69. Isolation cell placement**

Again, sophisticated, standards-based EDA tools are available to handle this automatically, while other EDA tools require the designer to manually create regions for isolation cells to be inserted—an error-prone process.

Common problems that may occur while inserting isolation cells include placing the isolation cells in the wrong power domain or hooking up the isolation power supply to the switchable power supply instead of the always-on power supply. These are catastrophic issues!

## State Retention Register Handling

For SRPG, regular registers in PSO domains are transformed or swapped into state retention registers during synthesis.



**Figure 70. State retention register scheme**

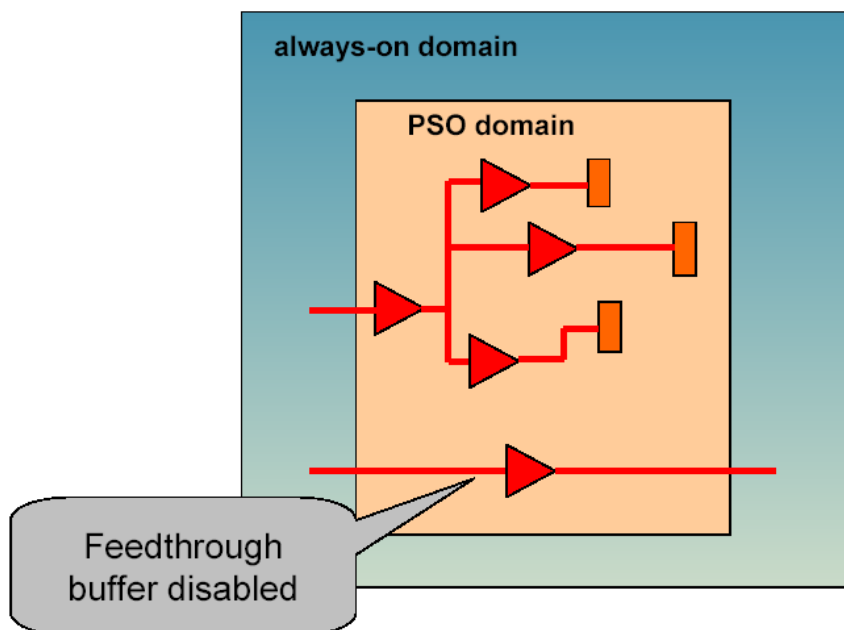
State retention registers require two types of power supplies: a switchable power supply and an always-on power supply. This introduces some complications and penalties in power routing area requirements. The physical designer, or implementation tool, must allocate extra area to accommodate this additional power routing.

### Always-on Buffering

Always-on buffering is required because certain nets in the power shutoff domain have to remain on at all times; for example, control signals for SRPG registers that feed through nets.

Always-on buffering is also handled in physical implementation.

Figure 71 shows an always-on domain and a PSO domain. In this case, since the feedthrough buffer resides in the PSO domain, it would be powered down and disabled. So the buffer must be an always-on cell.

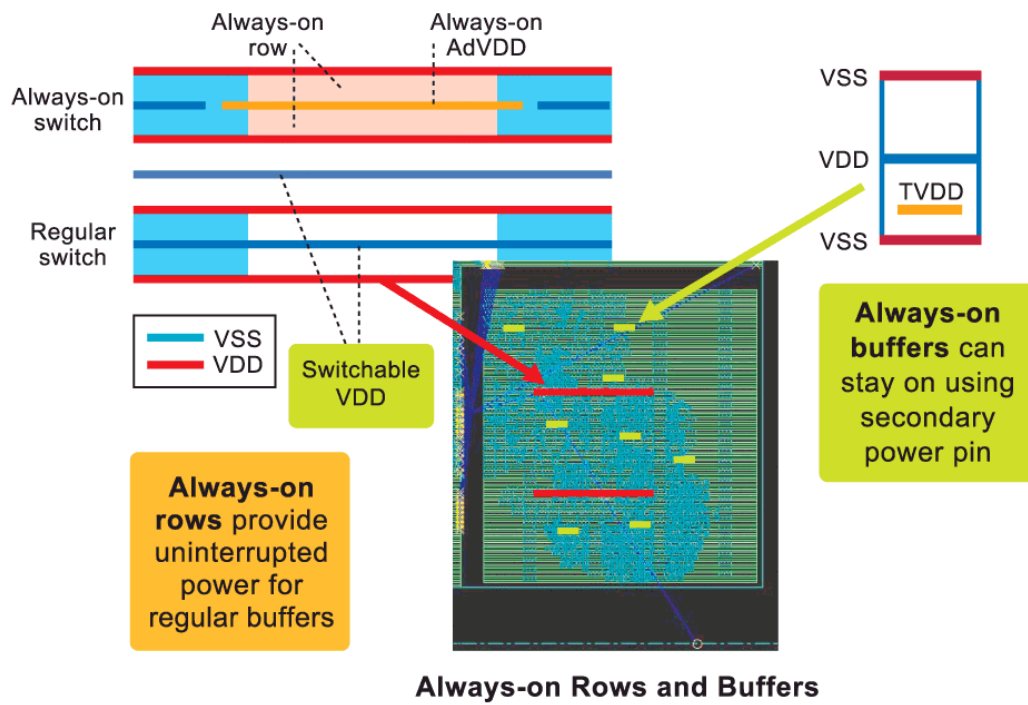


**Figure 71. Always-on and PSO domains**

As shown in Figure 72, both always-on rows and always-on buffers are supported.

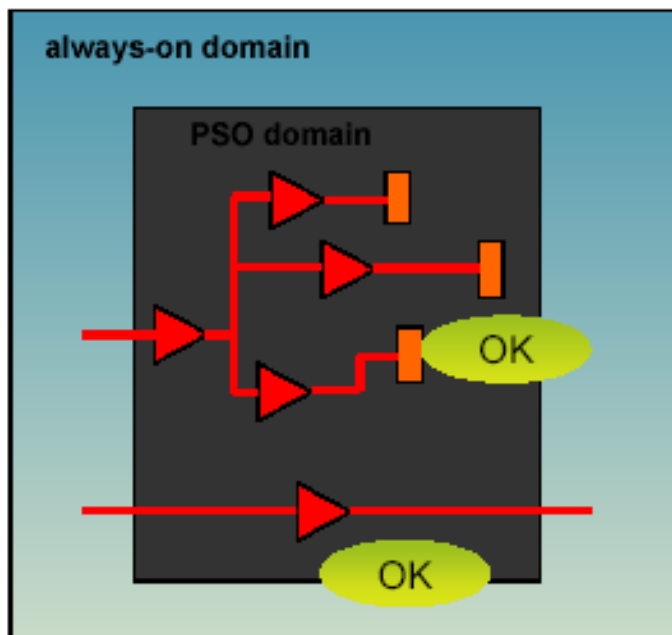
- *Always-on rows* provide uninterrupted power for regular buffers
- *Always-on buffers* can stay on using a secondary power pin





**Figure 72. Always-on rows and buffers**

With always-on buffering support, always-on nets can be implemented correctly.



**Figure 73. Transformed always-on and PSO domains**



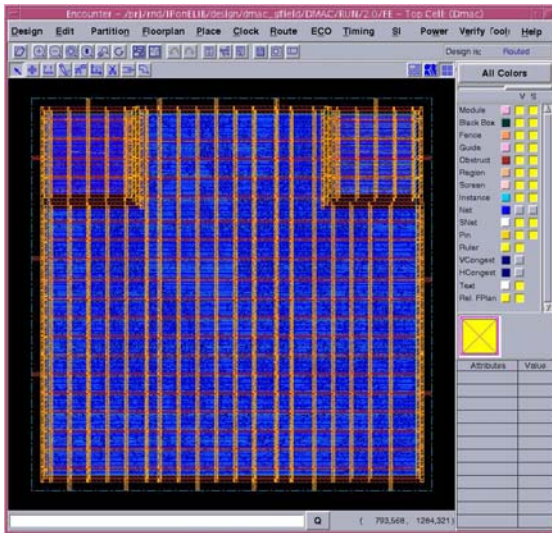
## Dynamic Voltage/Frequency Scaling (DVFS) Implementation

In the implementation stage, DVFS is accomplished using a combination of MSV and multimode/multi-corner (MMMC) techniques. Utilizing power domains is a requirement for implementing DVFS designs. In addition, these power domain definitions must be consistent with front-end definitions of power domains, which again are automated with CPF.

DVFS differs from MSV in that with DVFS, a single power domain may operate at different modes, where each mode has a different supply voltage and operating frequency.

In implementation with DVFS, the challenges are very similar to DVFS in synthesis: juggling different operating voltages (with their assigned, different timing libraries) and different operating frequencies (different timing constraint files). In more advanced EDA tools, these different combinations are optimized in parallel, automating the process. Although this may result in longer run times to achieve design closure than with traditional, non-DVFS designs, the power benefits are worthwhile.

For example, the design below shows DVFS techniques implemented in the layout. In the baseline or active mode of operation, all blocks operate at 125MHz and 1.08V. In slow mode, one block operates at 66MHz and 0.9V, which conserves power. In standby, two of the blocks are powered down completely.



Mode	Core	Drowsy	Dull
Baseline	1.08V 125MHz	1.08V 125MHz	1.08V 125MHz
Slow	1.08V 125MHz	1.08V 125MHz	0.9V 66MHz
Standby	0.0V	1.08V 125MHz	0.0V

**Figure 74. Three modes of operation with DVFS**

## Substrate Biasing Implementation

Substrate biasing, also known as back biasing, involves biasing the voltage of the body (bulk) of the transistors. The PMOS bulk is biased to a voltage higher than  $V_{dd}$ , and the NMOS bulk is biased to a voltage lower than  $V_{ss}$ . This reduces the leakage current of the transistors. For single-well technology, the bulk of the PMOS is connected to the n-well and the bulk of the NMOS is connected to the p-substrate. For dual-well technology, the bulk of the NMOS is connected to a p-well.

### Charge Pumps

Depending on the library, substrate biasing can be done for the PMOS, NMOS, or both. To bias the bulk of the NMOS and PMOS of the standard cells, voltages are created by charge pumps, which are custom blocks that output VDDbias and VSSbias voltages.

These charge pumps, which are custom macros about the size of PLLs, provide VDDbias and VSSbias. These voltages then need to be distributed across the parts of the chip that utilize substrate biasing. There are two methods for distributing the bias voltages to standard cells:

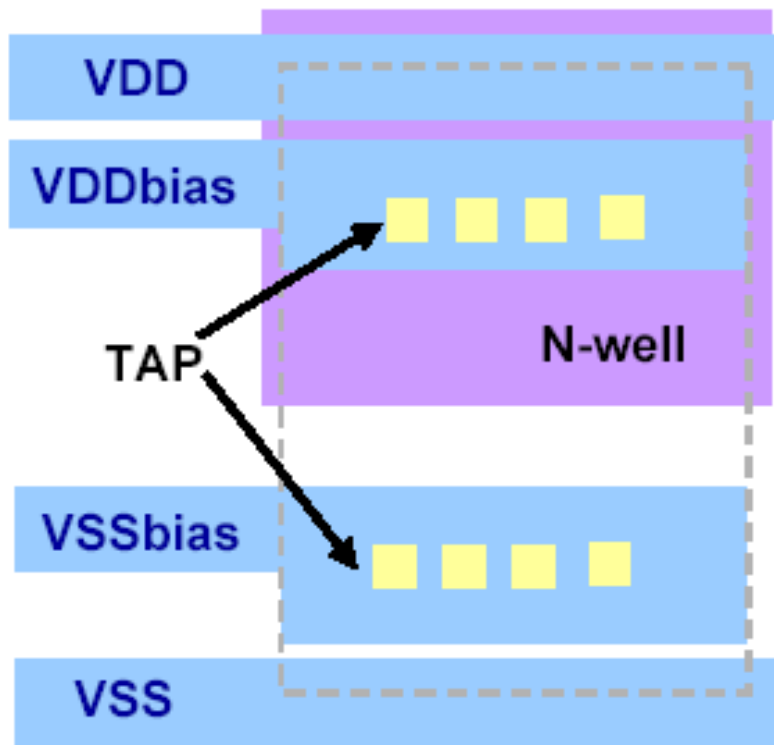
- Using well-tap cells (body-bias cells)
- In-cell taps, having VDDbias and VSSbias pins for each standard cell, then tapping those pins to n-well and p-sub, respectively

### Well-Tap or Body-Bias Cells

Well-tap or body-bias cells tap VDDbias and VSSbias to n-well and p-sub, respectively. Theoretically, each standard cell row must have at least one well-tap cell. In reality, multiple body-bias or well-tap cells are needed per standard cell row to prevent latch-up. Designers usually have a rule of one tap cell placed in a standard cell row per every certain distance, at regular intervals.

Adding well-tap cells actually saves area, because compared with the second method listed below, the only area increase is for the well-tap cells (which are smaller than the average 1x inverter).

Figure 75 shows a typical body-bias cell. It looks similar to a normal non-bias cell, except for two differences: The n-well is tapped to VDDbias instead of  $V_{dd}$ , and the p-sub is tapped to VSSbias instead of  $V_{ss}$ . Placing this cell at multiple points in every standard cell row will tap the n-well and p-sub of that row to VDDbias and VSSbias, respectively.



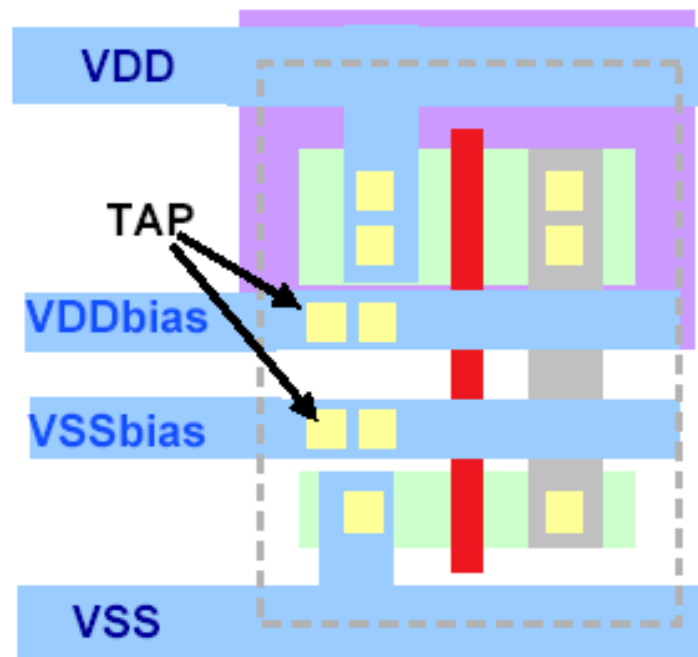
*Figure 75. Well-tap or body-bias cells*

### ***In-Cell Taps***

In-cell taps means having VDDbias and VSSbias pins for each standard cell, then tapping those pins to the n-well and p-sub, respectively. Extra pins are used to connect VDDbias and/or VSSbias to n-well and p-substrate, respectively, in each standard cell.

This method provides a consistent bias voltage level to the n-well and p-sub, but uses more area, since each standard cell has to reserve area for the bias voltage pins as well as the tap area. It also takes up a significant amount of routing resources, due to the need for routing every VDDbias and VSSbias pin to the bias voltage sources.

Figure 76 shows a standard cell that employs VDDbias and VSSbias pins. Here, the separate body-bias cell is not needed, because the taps to n-well and p-sub are embedded in the standard cells. Each standard cell has an extra VDDbias and VSSbias pin, which is connected to metal shapes. The metal shapes are then tapped to n-well and p-sub.



**Figure 76. In-cell tap approach**

### **Potential Issues with Substrate Biasing**

Designers who choose to utilize substrate biasing may run into two potential issues, involving p-substrate separation and bias voltage distribution.

#### **P-Substrate Separation**

For single-well technologies, the entire chip silicon *is* the p-substrate. That is, except for the parts of the chip that have been made into the n-well, the entire chip die is essentially the p-sub. That means if the designer chooses to bias the p-substrate, the entire substrate of the chip would be biased. This is rarely desirable, because usually certain parts of the chip (for example, any analog blocks) should not be biased.

This is not a problem for n-well biasing, since the n-well of the chip is easily separated.

This is also not a problem for dual-well technologies, which have a p-well and n-well. Therefore, the p-well can be separated from the rest of the chip, just like the n-well.

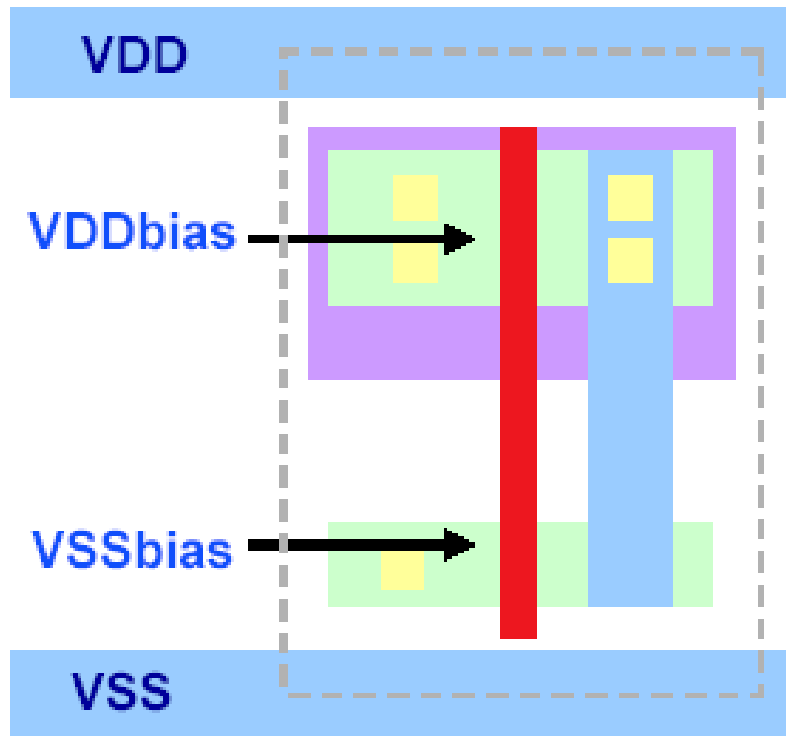
## Bias Voltage Distribution

Regardless of the bias voltage distribution method, the bias voltage nets (VDDbias and VSSbias) still have to be routed from the charge pump to the well-tap cells or standard cells. Most EDA tools today do not have special functionality for substrate biasing. Therefore, the designer might run into issues while routing the bias voltage distribution nets.

More important, these distribution nets take up a significant amount of routing resources and might adversely affect the routability of the design.

## Diffusion Biasing

An alternative to substrate biasing is diffusion biasing, which bypasses the substrate separation issue. In this technique, the diffusion of the transistor is biased instead of the bulk (see Figure 77).



*Figure 77. Diffusion biasing*

Note that as processes shrink, substrate biasing is predicted to be overshadowed by power shutdown. This is because the power-saving returns for substrate biasing are diminishing with smaller processes, thereby making PSO a more attractive choice.

## CPF Implementation Summary

The implementation phase for low-power devices brings about its own complexities and challenges, but these can be solved with the correct knowledge, standards, tools, and methodology.

- Power intent consistency can be solved using a standard power format such as Si2's Common Power Format (CPF)
- Physical handling of low-power constructs (e.g., isolation cells, level shifters, power switches, etc.) are automatically handled in advanced EDA implementation tools

However, the designer still needs to have conceptual knowledge of these low-power constructs.

As low-power design emerges and becomes more automated based on CPF, juggling power, performance, and area can be seen as just a progression of design and implementation. Trade-offs for power simply add another axis for the design space.

While traditional flows were a trade-off between timing and area, designers now face the challenge of power as another constraint in designs going forward.

## CPF User Experience



### ARC Energy PRO: Technology for Active Power Management

© ARC International 2008.

ARC International is the world leader in configurable processor technology. ARC licenses configurable CPU/DSP processors and multimedia subsystems, which enable customers to design products that give them a strategic advantage in today's highly competitive consumer electronics marketplace. Using ARC's patented ARCHitect processor configurator with ARC's configurable subsystems and cores, designers can develop highly differentiated SoCs. These SoCs consume less power, are less expensive to produce, and provide protection from cloning, offering distinct advantages over non-configurable, fixed-architecture alternatives.

#### Overview of ARC Energy PRO

ARC Energy PRO offers technology for active power management that reduces power by as much as four-fold through an integrated hardware, software, and EDA flow based on the Common Power Format (CPF). It is ideal for battery-operated portable applications such as WiMAX, digital radio, medical devices, etc.

ARC currently offers two processor cores based on Energy PRO, and the technology will be an integral part of future processor cores, multimedia subsystems, and their applications.

#### The Power Struggle

Traditional low-power design challenges involved increasing functionality, minimizing costs of packaging and cooling, and improving reliability. However, increasingly, the emerging low-power design challenge is extending battery life.

In the last three decades, SoC computational power increased by more than 4 orders of magnitude, while battery capacity has increased by only about 4x; this trend will continue. The solution is to address power across all phases of the product design with CPF-enabled flows.

## Designing Low-Power Solutions

### Configuring for Low Power

To serve markets such as wireless, networking, consumer, multimedia, and storage, configurability and extensibility are key.

#### Wired & Wireless Networking



#### Consumer/Multimedia



#### Storage



**Figure 78. Target applications**

Configuration of the design for each target application means minimum design size, with no silicon wastage, and the lowest-power design, with no inactive functional blocks. Extensibility means easy addition of specialized functions and more opportunities for designers to contribute further product differentiation.

### Power Management Techniques

Energy PRO power management techniques include traditional techniques such as fine-grained clock gating and user-driven multi- $V_{th}$  optimization. Advanced techniques address power issues in both active and inactive modes of operation.

For inactive modes and blocks:

- Reduce power when on “standby” for long periods.
- Provide techniques to reduce latency on restart.
- Gate the clocks at the highest possible level.
- Power down the core of the design.

During active modes:

- Target power reduction when operational.
- Techniques must have no impact on functionality.
- Functional latency may be impacted.
- Gate the clocks for a function when it is not in use.
- Reduce the voltage and frequency for non-compute-intensive operations.



Coordinated SoC-wide power management also provides an interface for extending these techniques to the rest of the SoC.

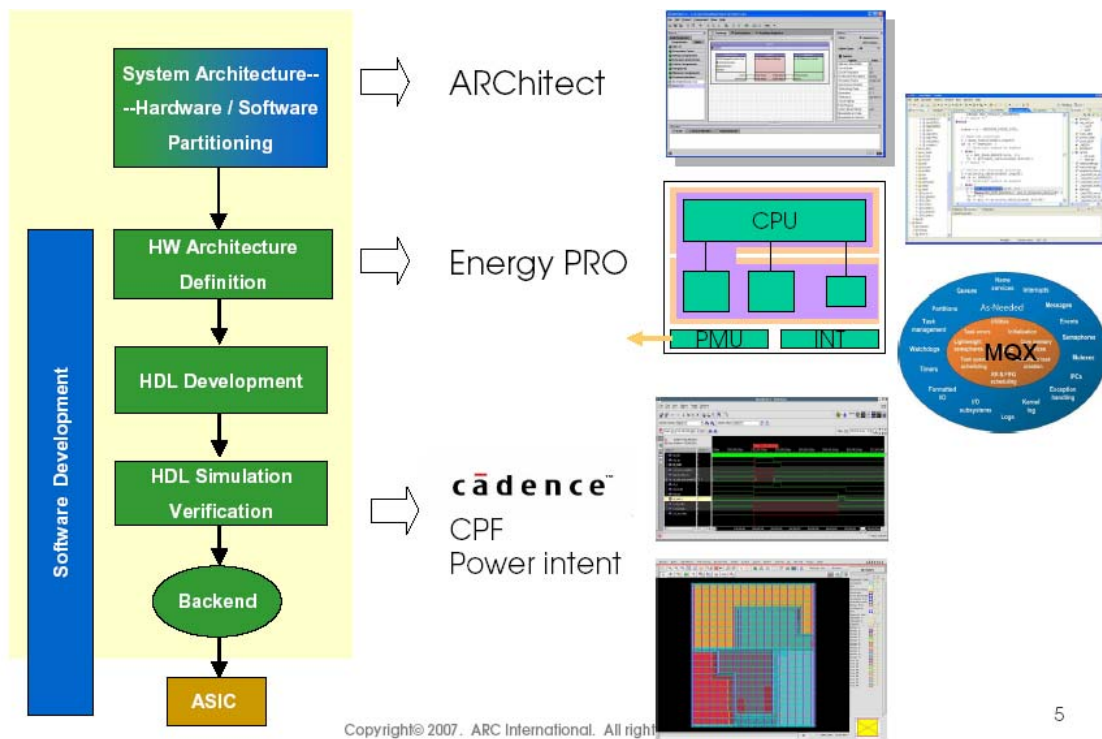
## Energy PRO Software Components

Key features of Energy PRO are invoked under software control, such as switching power modes and scaling voltage and frequency (DVFS.)

The ARCompact ISA was enhanced to support Energy PRO. ARC-provided software provides access to Energy PRO features that can be used by both the operating system and directly by applications.

ARC's MQX-EP: Energy PRO-aware RTOS provides an applications interface to Energy PRO; records power management activity; and can intelligently adjust settings based on thread requirements and workload profile.

## Energy PRO Design Flow



**Figure 79. ARC Energy PRO design flow**

In the design flow, simple low-power techniques can be incorporated at discrete points: fine-grained clock gating during synthesis and multi- $V_{th}$  optimization during synthesis and/or layout.

However, advanced techniques are complex and affect multiple tools throughout the design flow. This is where CPF is key.

- Accurate simulation of power-down modes
- Insertion of isolation cells and level shifters during synthesis
- Timing optimization with special power cells
- Placement of voltage regions and cells into the correct regions
- Verification of power intent

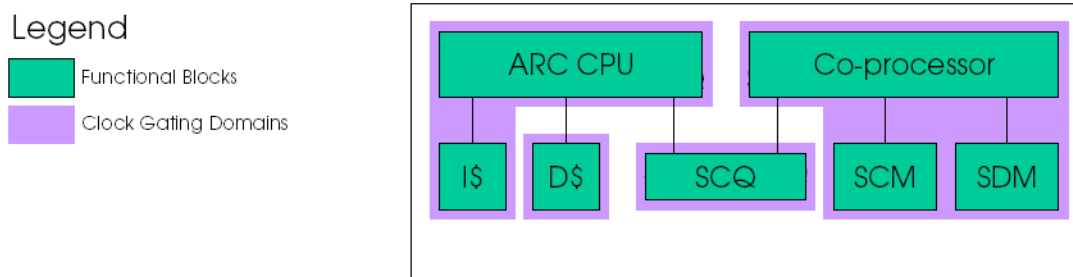
A project was undertaken in partnership with Cadence and Virage Logic to develop a low-power reference flow, based on CPF, deploying advanced power-saving features for ARC processors.

Objectives for the project were as follows:

- Using a sample design, capture the power intent in the form of a CPF file.
- Develop a low-power reference flow for the sample design.
- Make the power intent data configurable across ARC processor cores.
- Make the design flow configurable for use with ARC cores.

### ***Project Subsystem: ARC CPU with Co-processor***

The project subsystem is an ARC CPU with a co-processor that can process large data streams. The seven functional blocks in this design are shown below. The diagram also shows the four different domains for clock-gating power management.



**Figure 80. Project subsystem**

When processing high-bit-rate data streams, both the ARC CPU and the co-processor run flat out for high performance. When processing a lower-bit-rate data stream, the subsystem can be run at a lower frequency. For generic processing, the co-processor can be inactive.

This architecture lends itself to several advanced power management techniques, including power shutoff (PSO or power gating) and voltage scaling (DVS).

Power Intent: Power Shutoff

First, let’s explore PSO and its architecture. The following diagram shows the two different power domains (CORE and SIMD) relevant to PSO, with control signals, power switches, and isolation. Note that a new block of always-on logic has been defined for a total of eight blocks.

The three modes of operation, PDOS, PDO, and PD2—combined with the two power domains—are summarized in the table.

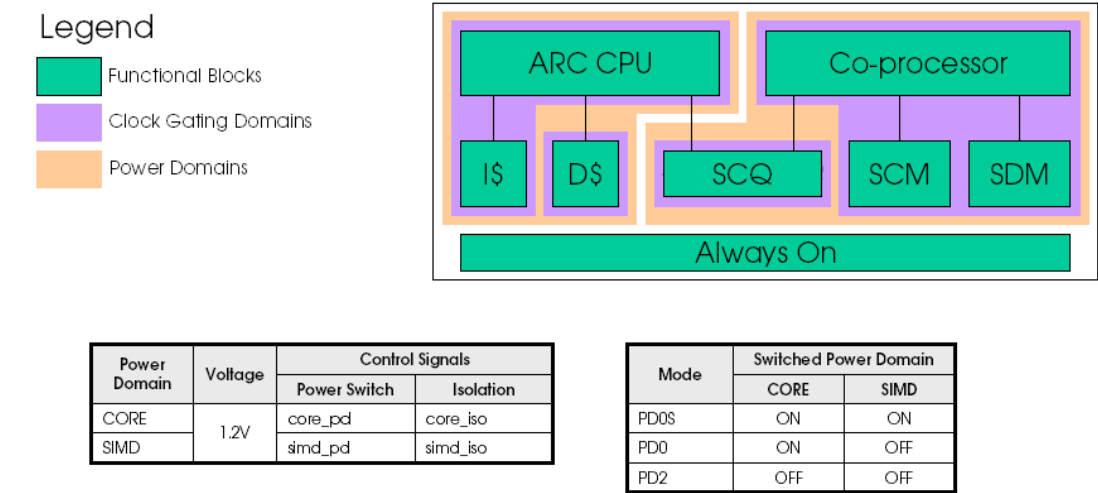
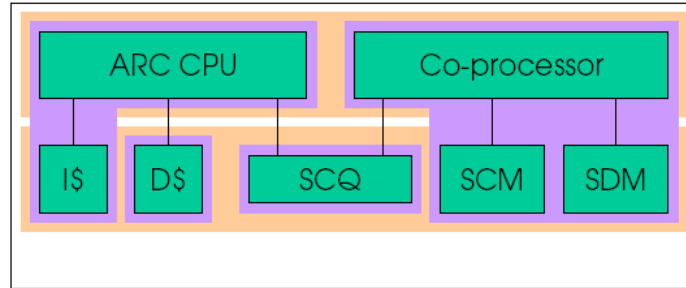
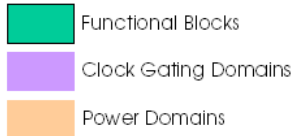


Figure 81. Project subsystem with PSO

Power Intent: Dynamic Voltage Scaling

Dynamic voltage scaling adds additional complexity. It introduces two differently defined power domains appropriate for DVFS, as shown in the block diagram. The three performance modes, with associated voltages, and their control signals are shown in the table on the left. The table to the right shows, by mode of operation or performance mode, the status of the two switched power domains, RAM and CORE.

## Legend



Performance Mode	Voltage	Control Signals
HIGH	1.2V	ram_high, core_high
NOM	1.0V	ram_nom, core_nom
LOW	0.8V	ram_low, core_low

Performance Mode	Switched Power Domain	
	RAM	CORE
C0	HIGH	HIGH
C1	HIGH	NOM
C2	NOM	LOW
C3	LOW	LOW

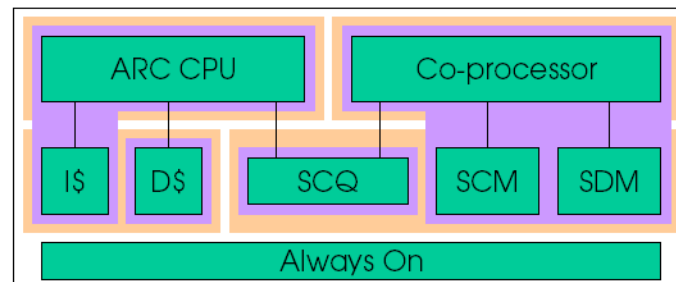
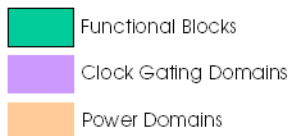
**Figure 82. Project subsystem with DVFS**

## Power Intent: Combined

Combining the two power management techniques, there are now four power domains, layering four clock-gating domains, defined for the eight blocks in the system.

The tables describe the voltage levels, the switched power domains, and the mode-dependent behavior of the low-power architecture.

## Legend



Name	Voltage
HIGH	1.2V
NOM	1.0V
LOW	0.8V

Perf. Mode	Switched Power Domain	
	RAM	CORE
C0	HIGH	HIGH
C1	HIGH	NOM
C2	NOM	LOW
C3	LOW	LOW

Mode	Switched Power Domain							
	CACHE/SIMD_RAM*				CORE/SIMD*			
	C0	C1	C2	C3	C0	C1	C2	C3
PD0	H	H	N	L	H	N	L	L
PD1	H	H	N	L	H	N	L	L
PD2	L	L	L	L	0	0	0	0
PD3	0	0	0	0	0	0	0	0

\* SIMD may be powered down independently

**Figure 83. Project subsystem with PSO and DVFS**

## Power Savings

The following tables summarize the benefits of the CPF-enabled low-power architecture. Power saved by DVFS during low-bit-rate data streams vs. high-bit-rate data streams is over 50 percent. DVFS contributes almost 50 percent for generic processing at high and lower frequencies.

PSO potentially saves even more—depending on the end design application—during standby, wait, and other power-down modes.

Power @ TSMC 90G	Dynamic Power(mW)			Leakage Power(mW)		
	Cell	RAM	Total	Cell	RAM	Total
Test subsystem	22.6	18.4	41	1.3	3.5	4.8

Power Saving Mode	Power (mW)
High bit-rate data stream @166MHz, 1.0V	45.8
Generic processing , co-processor off @166MHz, 1.0V	28.4
Low bit-rate data stream @100MHz, 0.7V	17.9
Generic processing, co-processor off @100MHz, 0.7V	14.7
No function being performed, clocks running	11.7
Clock Gated off at source	4.8
Complete core power down	0.4

**Figure 84. Power savings**

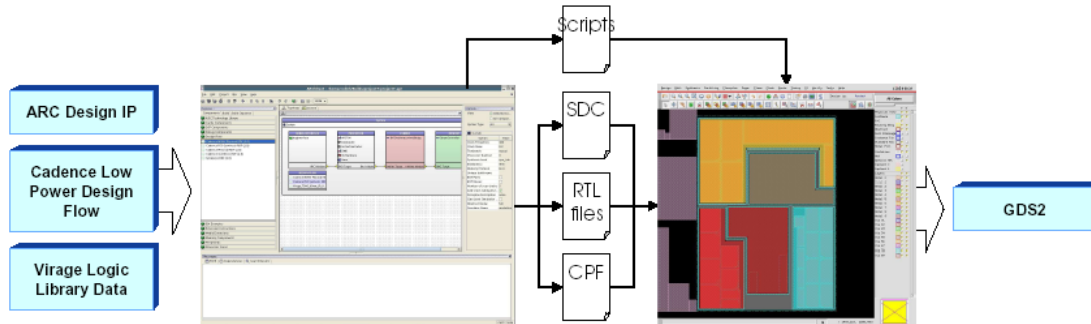
## Design Flow Implications

ARC believes that the power intent of the design needs to be clearly understood throughout all the design flow stages, and CPF-enabled design tools are key for RTL simulation, power analysis, synthesis, formal verification, and place and route.

Design and verification with CPF identifies and prevents challenging problems with isolation cells and level shifting. During clock gating, the flow ensures that powered-down blocks no longer receive a clock signal. And to prevent rogue RTL, when a signal goes from one module to another, it ensures that there are no “simple” operations happening in an unpowered domain.

## Reference Design Flow

ARC has now released a design flow for the new cores, wherein CPF describes the power intent and ensures consistent implementation across all tools in the design flow.



**Figure 85. ARC Reference Design Flow**

ARChitect configures the design in the context of the Cadence Design flow and library data. Library data, including specialist low-power cells, have been supplied by Virage Logic.

## Conclusion

ARC Energy PRO represents a new active power management technology that reduces power by as much as four-fold. Its end-to-end, fully verified power management solution reduces time to market for advanced SoC designs and is ideal for battery-operated portable applications.

The technology and flow are based on CPF and integrated with the Cadence Low-Power solution to ensure accurate implementation at all flow stages.

The Energy PRO technology will be included in future ARC processor cores and multimedia subsystems spanning a breadth of design applications and markets.

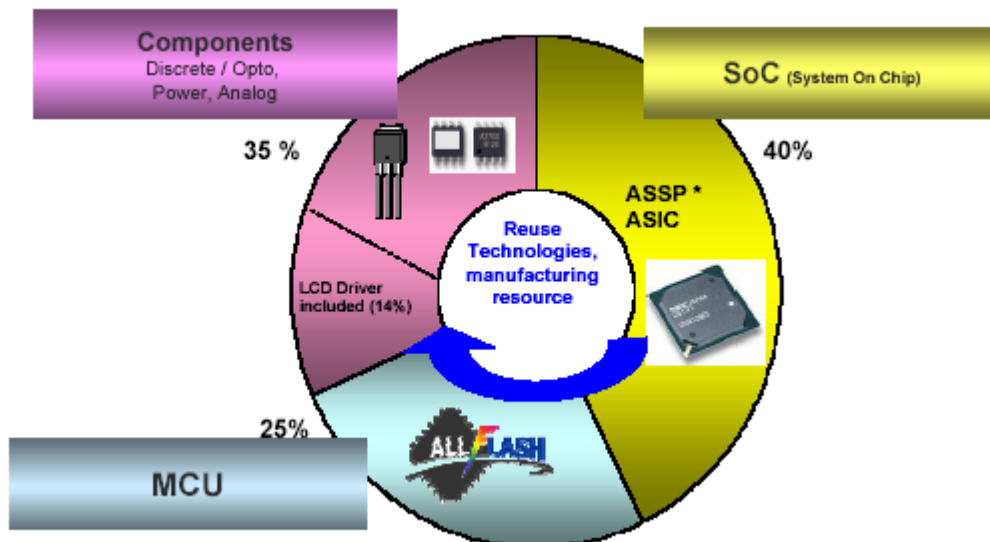
## NEC Electronics: Integrating Power Awareness in SoC Design with CPF

*By Toshiyuki Saito, Senior Manager, Design Engineering Development Division, NEC Electronics.  
© NEC Electronics Corporation 2008.*

NEC Electronics Corporation specializes in semiconductor products encompassing advanced technology solutions.

There are three pillars of the NEC Electronics business, and the goal is to create leading edge products in each of these focused domains:

1. Microcontrollers (MCU), leading the market with strength in application support, software development environment, embedded FLASH and architectures.
2. Discrete and IC, including LCD drivers and optoelectronics components, power components and analog components.
3. System on Chip (SoC) platform solutions with a competitive edge in advanced process technology, design environment, IP cores and libraries, and software.



**Figure 86. Three pillars of NEC Electronics business**

Low power in all its aspects is critical to NEC Electronics, as evidenced by their strong corporate commitment to Green environmental targets to reduce CO<sub>2</sub>; early support for the European Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS Directive), the EnergyStar program in the US, and similar directives under the Kyoto Protocol from the Kyoto Japan Convention. NEC Electronics provides semiconductor devices with increasingly advanced functions and high performance that help customers build greener products, and has created the UltimateLowPower™ program, which is shown below.

UltimateLowPower™ stands for a new low-power design environment, as well as process technology for low power. It includes exploring and driving fundamental technology, methodologies, and efficiencies in device, design, and process.

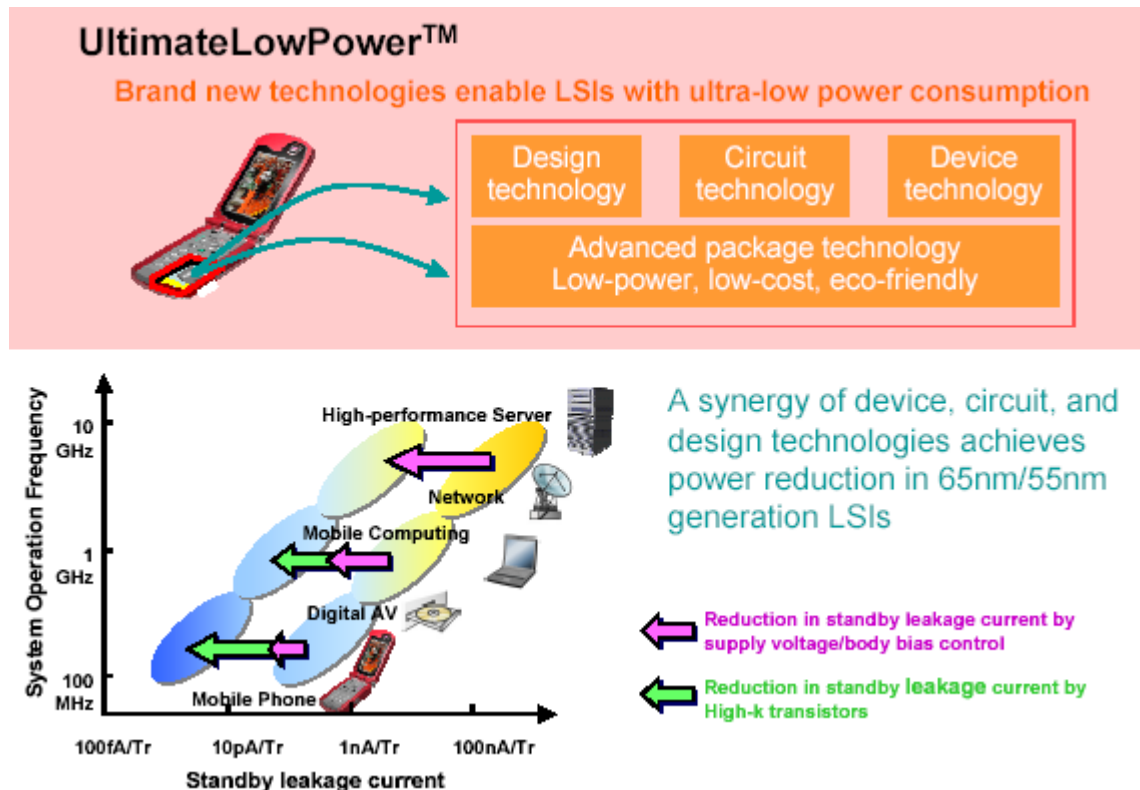


Figure 87. NEC Electronics UltimateLowPower, 2007 ([Ref. 12](#))

### NEC Electronics and CPF

Although NEC Electronics has done many low-power designs, in the past the design flow was tedious and troublesome. Many advanced power reduction techniques were used, but they were difficult to implement with the existing design flow.



And NEC Electronics is not alone: In the absence of a power standard, designers have been left to their own devices when describing low-power concerns across the design and implementation flow. As a result, to avoid problems and risks, most designers will be reluctant to adopt advanced low-power design techniques.

This explains NEC Electronics' strong interest in standards, especially with processes at 65nm and below providing significant motivation. Simplification of the low-power design flow leads to improved efficiency, reduced costs, improved quality, and a concomitant competitive advantage.

NEC Electronics was a founding member of PFI ([Ref. 13](#)), and began working with Cadence on CPF in April 2006. To date:

- A proof point project was designed to verify the CPF methodology with a test chip
- The methodology verified with CPF reflects the methodology today at NEC Electronics
- NEC Electronics provided over 300 requirements and inputs to the CPF standard
- 
- The rest of this chapter describes general low-power technology trends, followed by an example of recent low-power design success in NEC Electronics: a 65nm cell phone SoC. Next, it introduces the proof point project using CPF to improve the low-power design environment in NEC Electronics. Finally, it discusses the necessary spectrum of activities to promote CFP-based design, and provides a perspective of future low-power design progress.

### **Why Low Power?**

With consumers increasingly demanding feature-rich devices, which expend more power and generate more heat, there are significant business implications of low power consumption. Power reduction can bring competitive success across several axes:

- Battery life is critical for the success of mobile systems such as cell phones, digital still cameras, and handheld electronics. The chip developed for the latest NEC cell phone is discussed in detail later in this chapter.
- Heat suppression is important for wired systems such as servers, personal computers, set-top boxes, DVD recorders and decoders, graphics interface chips, and the digital TV market. The motivation for power reduction is cost: avoiding the cost of cooling systems and fans and their lack of reliability.
- Recently, automotive applications—such as keyless entry, safety, security, GPS and entertainment systems—also require dedicated low power efforts especially to minimize standby power.

Low-power design helps to secure the cost competitiveness of the SoC and of its downstream product applications across all markets. Benefits include:

- Packaging cost reduction: for example, saving \$1-4 per part in a low-cost plastic package
- Development costs with an efficient low-power flow are minimized, as well a turnaround time for SoC designs

As mentioned earlier, low-power design also contributes to preserving our global environment and is strongly demanded by governments worldwide. ([Ref. 14](#))

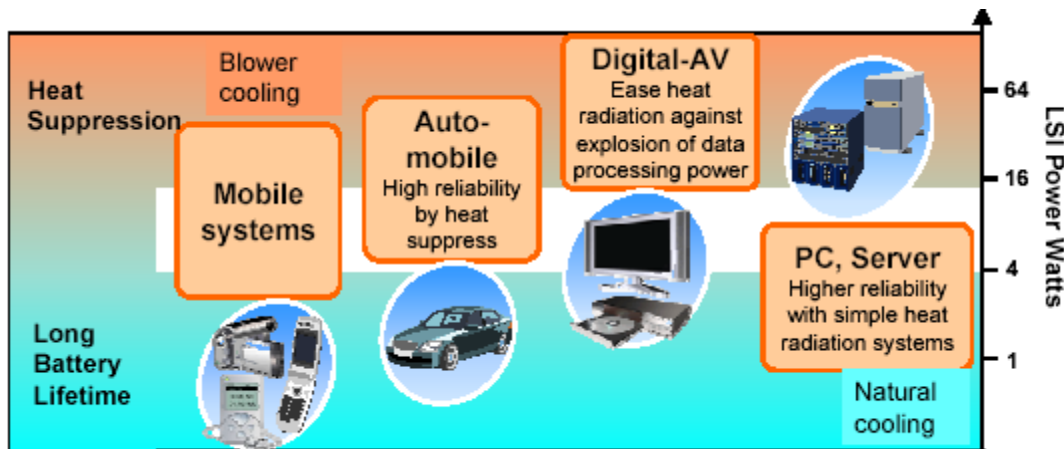


Figure 88. Low power consequences for various electronic products

### Trends in Power Consumption

Now let's look at the trends in power consumption as designers move to advanced process nodes.

Power consumption of a chip can be expressed as the following well-known formula:

$$\text{Power} = \text{Active Power} + \text{Leakage Power}$$

$$= \sum C * \text{Freq} * V_{dd}^2 + N_{Tr} * I_{off}(V_t) * V_{dd}$$

Where C is the total capacitive load of switching in frequency (Freq);  $V_{dd}$  is supply voltage;  $N_{Tr}$  is the number of transistors; and  $I_{off}$  is the off-leakage current of a transistor, which is a strong function of threshold voltage  $V_t$ .

Active and leakage power can be restated, and trends and relationships simplified, as described below.

For every process generation change, the number of gates in a chip increases by 2x, and the clock speed increases by 1.3x. However, the switching power of the unit gate decreases by 0.7x and the leakage power of unit gate is kept constant.

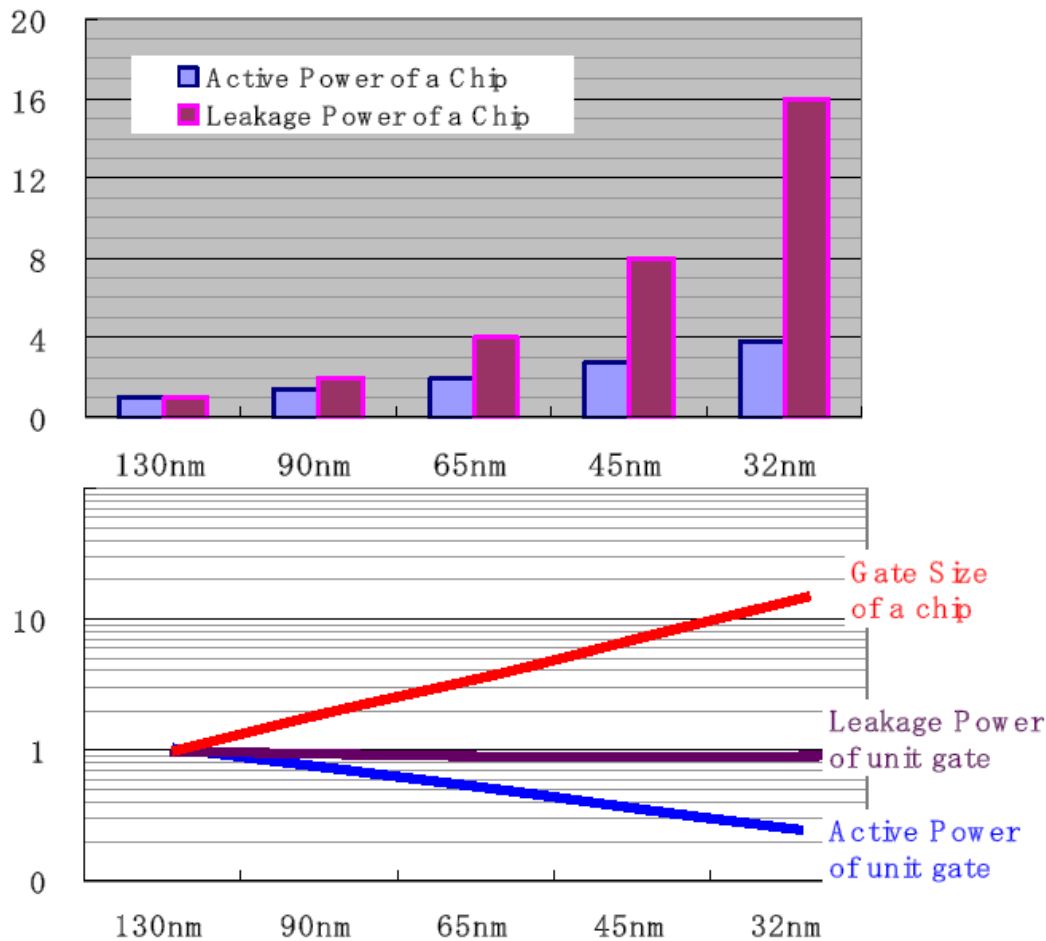
This is due to process miniaturization, in which switching load capacitance is decreased and off-leakage current is controlled by transistor device engineering.

Hence, the power consumption increases as follows:

$$\begin{aligned} \text{Active Power} &= \sum C * \text{Freq} * V_{dd}^2 \\ &= (\text{No. of gates}) * (\text{Clock speed}) * (\text{SwitchingPower of UnitGate}) \\ &= 2.0x * 1.3x * 0.7x = \sim 1.8x \end{aligned}$$

$$\begin{aligned} \text{Leakage Power} &= N_{tr} * I_{off}(V_t) * V_{dd} \\ &= (\text{No. of gates}) * (\text{LeakagePower of UnitGate}) \\ &= 2.0x * 1.0x = \sim 2.0x \end{aligned}$$

Active power and leakage power increases 80 percent and 100 percent respectively. ([Ref. 15](#))



**Figure 89. Trends In LSI power consumption**

The solution involves employing more advanced power management techniques throughout the design and implementation phases. For this, we need a comprehensive approach to low power.

### **Comprehensive Approach to Low Power**

Low power demands a very comprehensive technology flow. NEC Electronics' goals are to cover a solution both as wide as possible and as deep as possible. Physical design alone is not the solution, as other elements of the design chain have to be considered as well.

### **NEC Electronics' Successfully Deployed Low Power Techniques**

Low-power technology in NEC Electronics takes a synergistic approach among process, circuit and design innovation, as shown below.

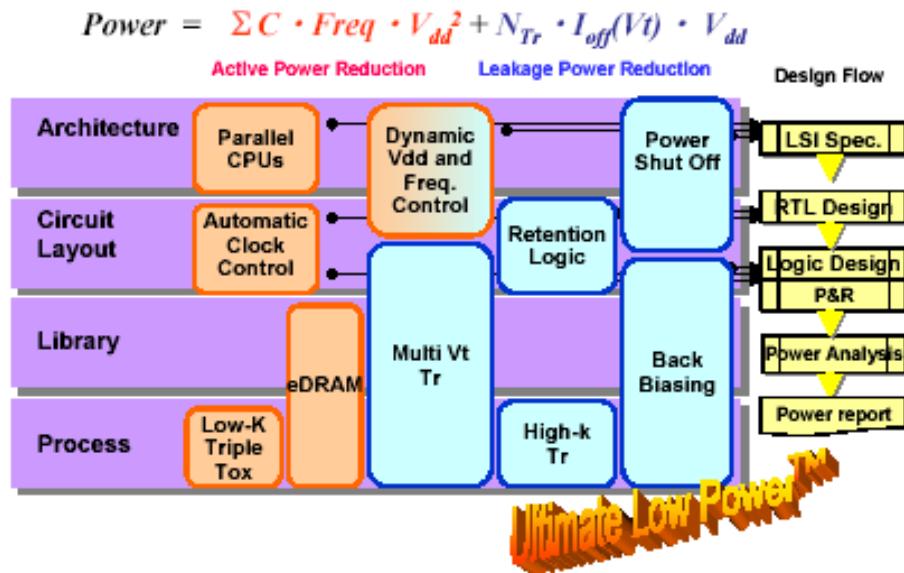


Figure 90. Low power technology in NEC Electronics

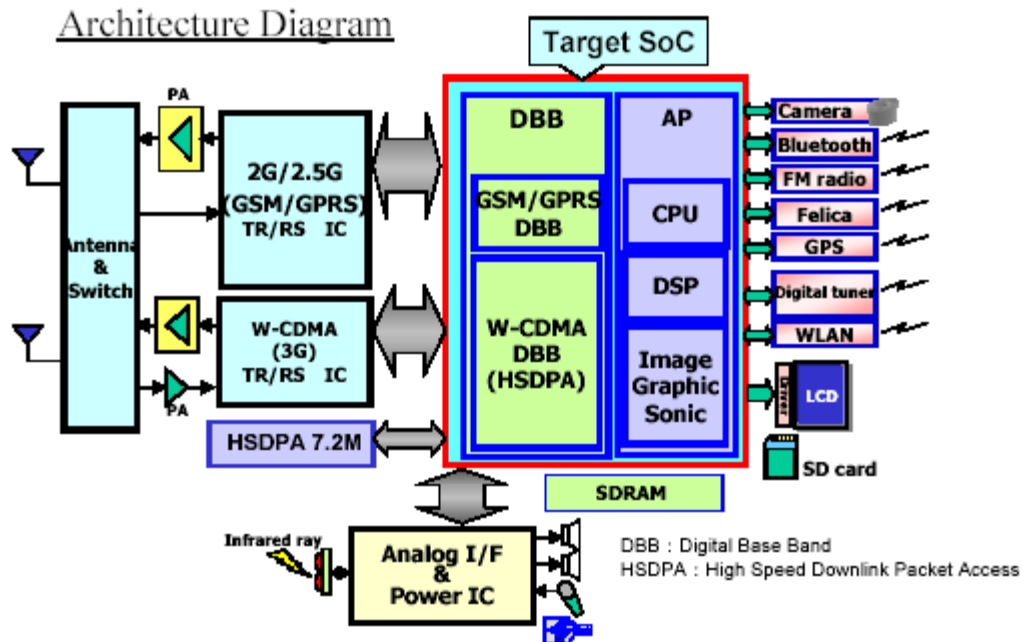
The most popular power management techniques used include multi-Vt and clock gating, which are very common because they have a low penalty regarding design complexity, area and timing, and because EDA tool automation for these techniques has been achieved. Power shutoff (PSO), dynamic frequency / voltage scaling (DVFS), and back biasing are advanced techniques which carry more penalties and have been more difficult to implement ([Ref. 16](#)).

### Example of Mobile Phone System SoC

The following example SoC represents a new mobile cell phone chip designed at 65nm and used in the NEC cell phone, and others. This fundamental architecture was originated as early as 2003, and has undergone evolution, enhancements, new standards, and process migration.

### Architecture

The following architecture diagram viewpoint shows the system architecture, including all the different modes of operation, application and baseband functionality, combined and packed into a single complex SoC.



*Figure 91. Architecture of NEC Electronics 65nm cell phone SoC*

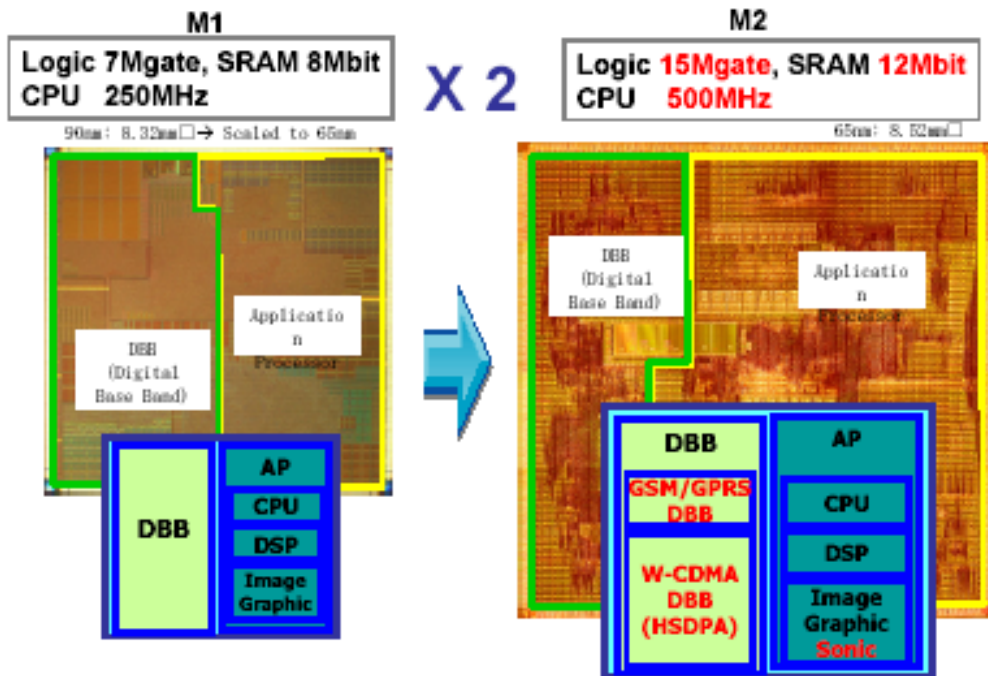
This chip is an example of the complexity and high level of integration that challenges low power goals; especially in the wireless market where battery life is paramount!

## Two SoC Implementations

The following diagram shows two physical layouts for the SoC:

- M1 with 7 million gates, 250MHz CPU, and 8Mbit SRAM was designed in a 90nm process
- M2 is twice the chip, with 15 million gates, 500MHz CPU, and 12Mbit SRAM was designed in a 65nm process

It is obvious that the targets keep on getting more aggressive.



*Figure 92. Two implementations of NEC Electronics 65nm cell phone SoC*

## Power Consumption Results

Power results indicated that if the M1 design had been implemented without advanced power management, the power results would have been completely unacceptable for both active and leakage power: over 2 times the power.

However, by deploying advanced reduction techniques including dynamic clock controls, multiple power domains with power shutoff, back bias, and multi-Vt, M2 delivered twice the performance with the same power specification as the M1 chip. The techniques reduced active power over 50% and leakage by over 60%.

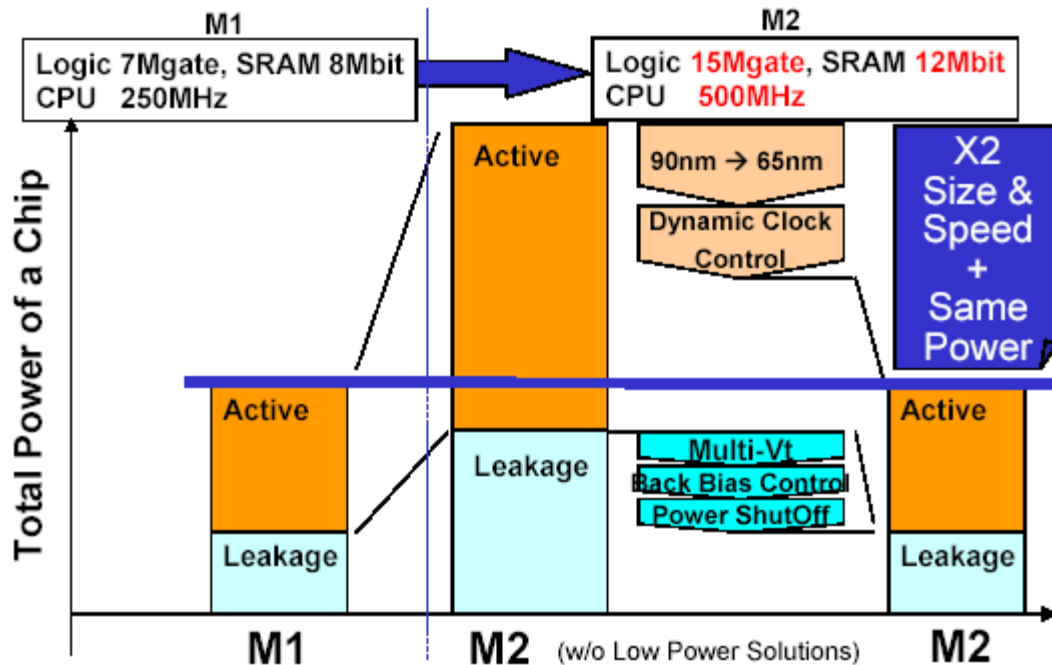


Figure 93. Power results for two implementations of NEC Electronics 65nm cell phone SoC

### Power Forward Initiative and CPF Expectations

The M2 design described in the previous section was accomplished without CPF. The EDA tools used in the flow included Cadence software, especially Cadence Encounter for physical implementation, synthesis from Synopsys, simulators from a variety of EDA companies, some NEC Electronics proprietary in-house tools, utilities to handle MSV, and many side files.

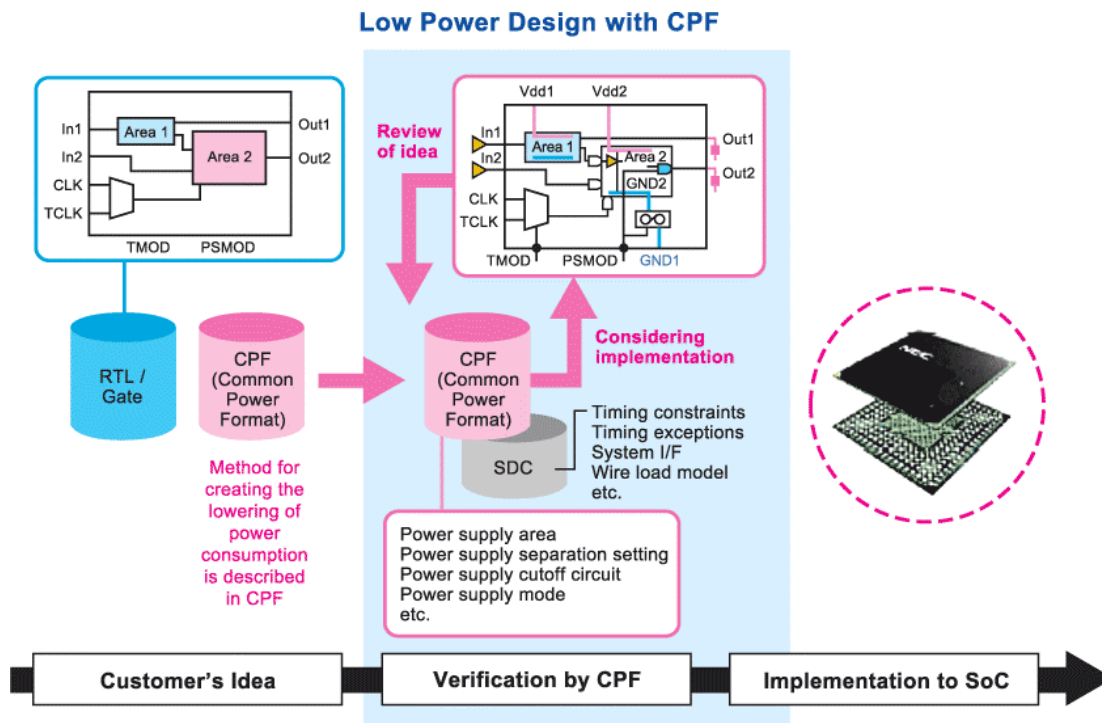
So, NEC Electronics sees the value added by CPF standard in improving the flow, especially with respect to high-level power verification capabilities.

NEC Electronics' original goals and expectations for CPF included:

- Significant productivity gain in high-level design and verification
- Design cost savings through a simplified low-power SoC design flow
- Quality improvement using the Common Power specification by all designers

To promote and to confirm these goals, NEC Electronics joined the Power Forward Initiative and created a joint Proof Point Project with Cadence, since CPF was the first power format available, in early 2006.





**Figure 94. Low-power design with CPF**

### **NEC Electronics CPF Proof Point Project: NEC-PPP**

The NEC Electronics Proof Point Project (NEC-PPP) with CPF successfully completed intensive validation of the CPF standard and the CPF-based flow, for major low-power methodologies within NEC Electronics.

### **Productivity Gain Example of Power Control Simulation**

One of the remarkable benefits from using CPF has been shown in power control simulation. The following is a real example of our 65nm design experience describing how power shutoff is implemented, with a traditional flow and with a CPF-enabled flow.

In a traditional flow for power shutoff simulation, designers must set unknown or "x" values for all necessary boundary pins of power shutoff domains, depending on what power mode they want to verify. It is not only tedious to write thousands of lines of test bench, designers also tend to lose quality of design, since it is difficult to develop a high enough number of corner cases to improve simulation coverage. The same simulation can be done using CPF just by adding simple power intent descriptions.

CPF simplifies both the description of low-power intent, and the verification, reducing test bench complexity significantly. The runtime and disk usage are also much more reasonable.

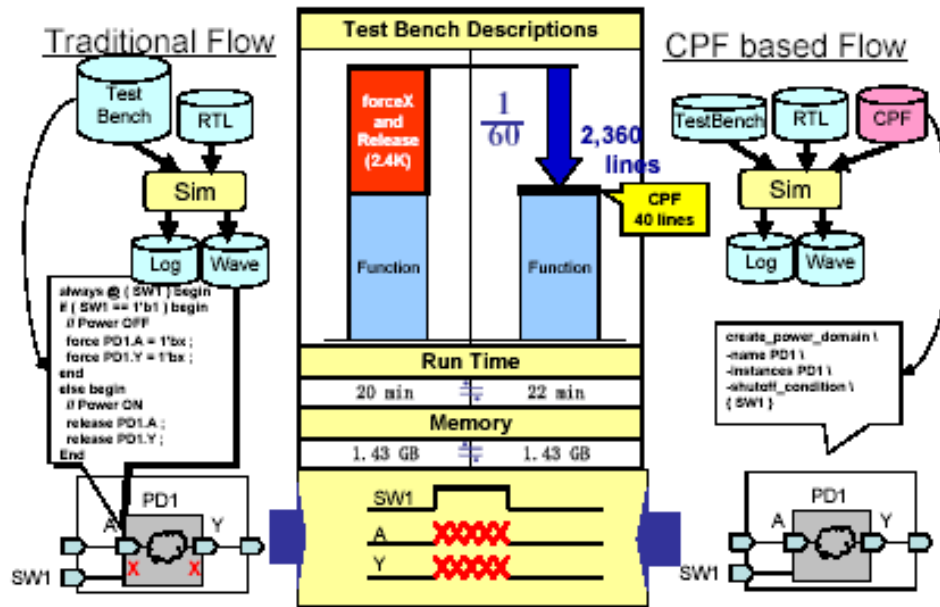


Figure 95. Power shutoff example

## NEC-PPP Test Chip

The NEC Electronics Proof Point Project (NEC-PPP) consisted of a test chip designed to provide exhaustive testing of the CPF standard, as well as its actual implementation in the tools flow, far more comprehensively than any representative real SoC could test. The test chip was carefully designed to exercise all major low-power design techniques used at NEC Electronics. These, of course, covered all techniques used in the cell phone chip described earlier. Since the NEC-PPP covered a very wide spectrum of design phases and has a high number of check points for the flow, and also because CPF-based design tools were not yet mature, many design flow iterations were expected during the NEC-PPP term. Therefore, the test chip was created to be as small as possible.

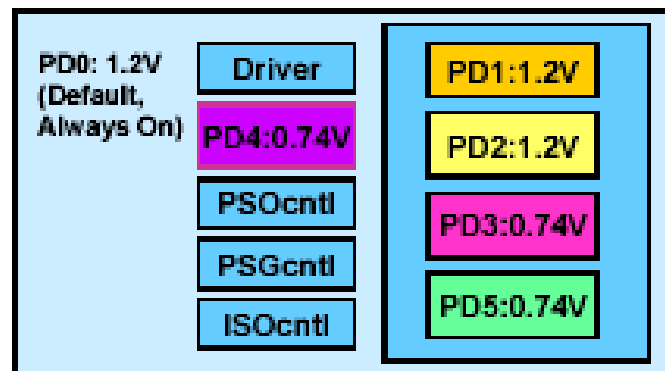
This small, but intelligently contrived replica circuit chip exercised all the corners of test cases for all low-power techniques, as used in real designs, but checked them in all their possible combinations, for necessary and sufficient testing. CPF descriptions of power intent included:

- Multi-Supply-Voltage (MSV)
- Power Shut Off (PSO)
- State Retention Logic (SRL)
- Variable Voltage Library (VVL)
- Clock Tree Gating (CTG)
- Adaptive Back Bias (ABB)

Another goal was to check that all the tools supported the semantics of CPF in the same way, making it easy to verify all these capabilities and to see the actual functionality of the tools on the design.

The test case was designed at 65nm, 1.2v, and combined:

- 5 power modes
- 6 power domains



*Figure 96. Test design and power structure*

It included the minimum number of components to check as many combinations of low-power functions as possible.

Power Mode	Power Domain					
	PD0	PD1	PD2	PD3	PD4	PD5
PM1	1.2V	1.2V	1.2V	0.74V	0.74V	0.74V
PM2	1.2V	PSO	1.2V	0.74V	0.74V	0.74V
PM3	1.2V	1.2V	PSO	0.74V	0.74V	0.74V
PM4	1.2V	1.2V	1.2V	PSO	0.74V	0.74V
PM5	1.2V	PSO	PSO	PSO	0.74V	PSO

*Figure 97. Power domains and modes*

## NEC-PP Check Points

NEC-PPP check points included:

- 146 criterion items for design flow validation
- 386 checkpoints to verify all criteria

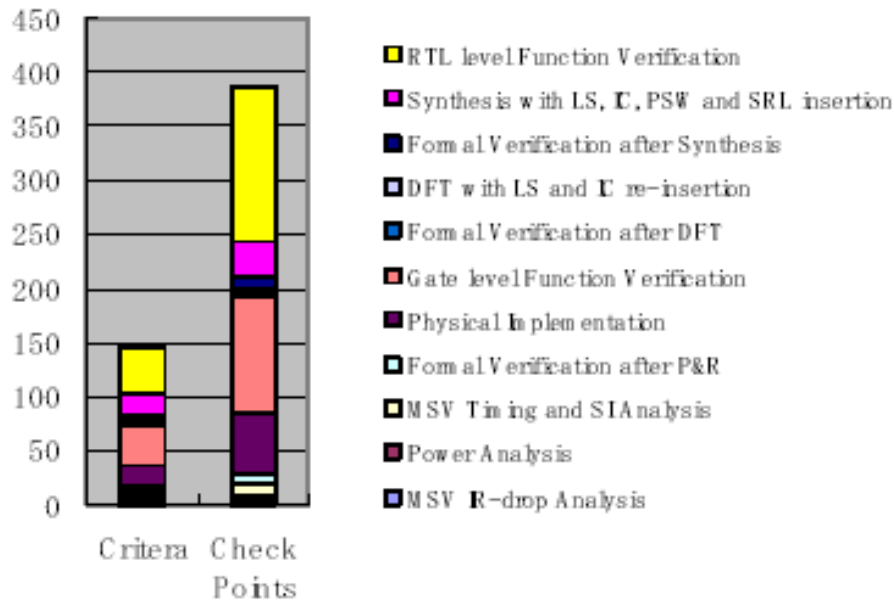


Figure 98. NEC-PPP check points

## NEC-PPP Results

All 386 check points were evaluated, successfully.

Only 23 issues were raised for CPF 1.0, in contrast with 116 detected for CPF 0.5. CPF semantics were clarified as well. Some issues will influence future CPF v1.x evolution. 60 tool & library issues were detected, and resolved, mostly CPF semantics interpretations.

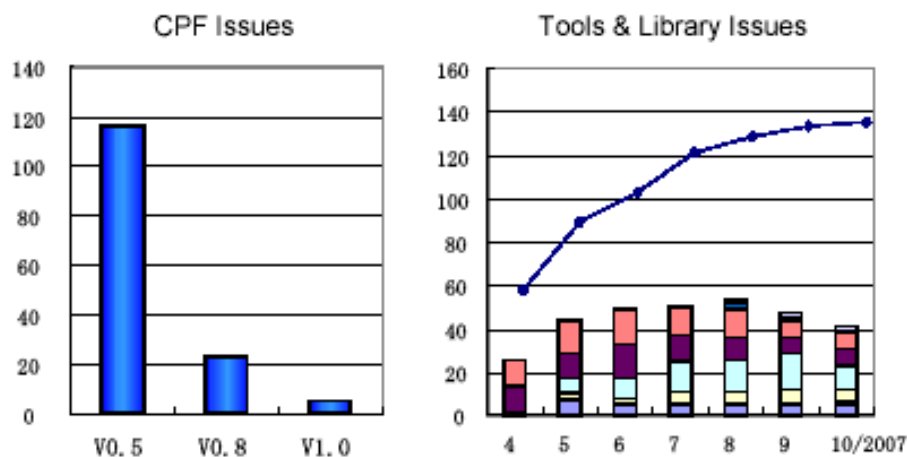


Figure 99. Learning from NEC-PPP

Examples of NEC-PPP CPF semantics and tool interpretation contributions:

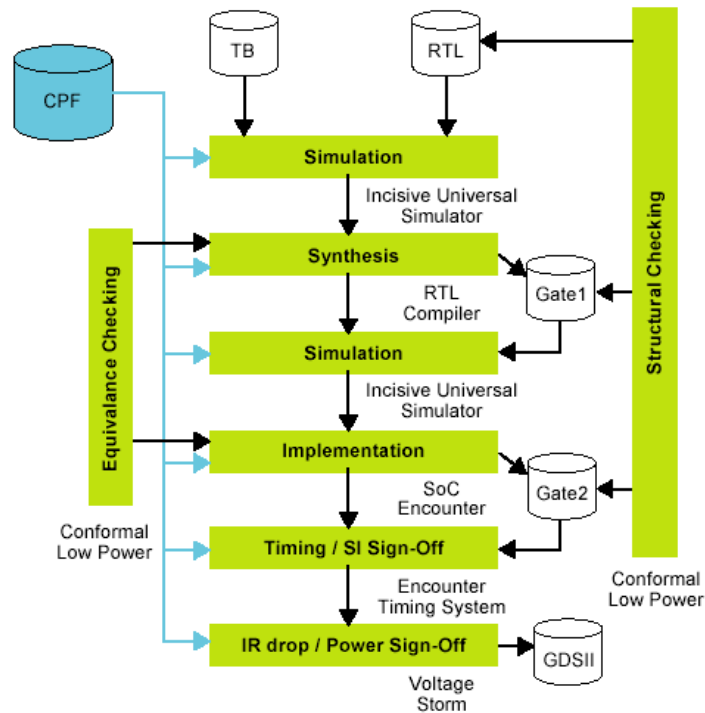
<i>For always-on cells</i>
▪ <i>Tool behavior and CPF semantics were inconsistent</i>
▪ <i>Hierarchical macro instantiation issue was resolved</i>
<i>Feed-through net handling</i>
▪ <i>IUS translated wrong X from feed-through net in off domain</i>
▪ <i>Improved feed-through handling: Feed-through nets can be modeled as buffer or wire</i>
<i>Combo cell description</i>
▪ <i>Simple description for combo cell, combined isolation and level shifter functions</i>
<i>CPF file verification capability</i>
▪ <i>Since writing CPF without human error is not always easy, CPF description check without netlist was requested</i>
▪ <i>Now, Conformal Low Power has checking utility (lint-like capability)</i>

### **NEC-PPP Outcomes**

The proof point project NEC-PPP concluded successfully with the following outcomes:

- Ensured support for major low-power methodologies at NEC Electronics
- Completed fundamental validation for flat design with CPF-based tools
- Contributed significantly to increased maturity of CPF and the CPF-based design flow
- Power control RTL and gate simulation based on CPF was verified successfully
- MSMV Conformal checking has enough capability to support virtual low-power cells in RTL
- MSMV physical implementation flow shows comparable quality of results with previous design flow (without CPF)
- MSMV/MCMM signoff timing and SI analysis has been tested

This level of success allows NEC Electronics to deploy a CPF-based flow for practical use in 2007.

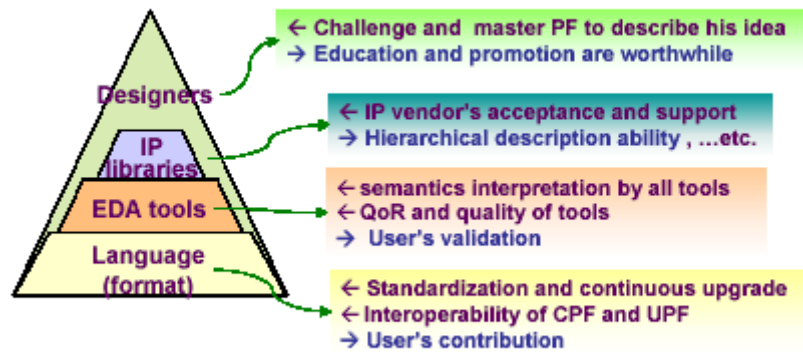


**Figure 100. CPF-Based flow for major low-power methodologies in NEC Electronics**

- Also, the NEC-PPP test chip is now used in Cadence regression test suites for CPF-enabled tools.

## CPF Adoption

For every new standard, including CPF, it is necessary to promote acceptance and adoption by a wider design ecosystem. To this end, many types of players, EDA vendors, IP and library providers, and designers, need to contribute respective responsibility and effort.



**Figure 101. Spectrum of adoption**

The CPF language (or format) is one of the best, but simply defining a format does not help in practical SoC design. An excellent tool-set and well-integrated design methodology are critically important, so for all EDA tool vendors the following efforts are recommended:

- Adopt a common R&D target of holistic low-power design systems
- Assure interoperability between multiple vendor tools: a critical issue. The success of CPF requires the buy-in of many EDA vendors, to drive the momentum towards general EDA acceptance
- Ensure that all tools interpret the CPF format in the same way

For IP vendors and libraries:

- It is requested to support IP cores with new format as well
- A CPF-based reference methodology for commercially major cores, including how to implement low-power cores in efficient way and how to use a core in chip design under the tradeoff between optimum power and other design factors
- Advanced IP users expect more specific descriptions and knowledge which can be prepared only by the IP vendor. This will enable users to adopt IP cores optimally

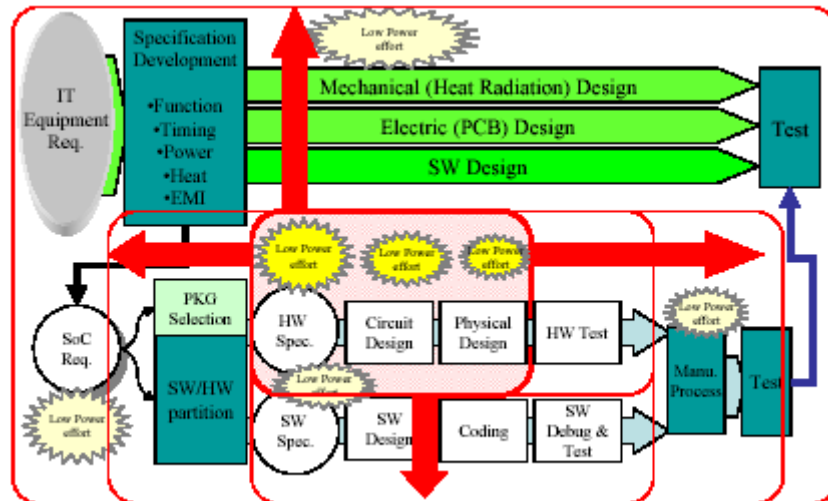
For designers:

- Mastering new language solutions format to describe ideas is essential
- Education and training in CPF should be prepared by responsible CPF drivers, without expensive fees

### ***A Perspective on Future Holistic Approach to Low Power***

Low power demands a very comprehensive technology for SoC design, as seen in previous sections.

An ideal low-power design methodology, however, should consider an even wider scope, covering the entire system through hardware and software design phases. In all design phases, there are various opportunities to promote low-power efforts.



**Figure 102. System specification with holistic low-power approach**

The above chart shows the new product definition flow from a setmaker who has a development idea for an electronic system. The initial product requirements lead to specification development, covering function, timing, power, heat and EMI specifications. Typically, the systems house (NEC or a customer) creates a specification, decomposes this into flows, then partitions and develops the software design. If an SoC is required, the systems designer requires a specification of the package and software/hardware specifications from NEC Electronics.

Low power should be considered at the time of mechanical, electronics hardware, and software partitioning, through the detailed implementation. But as the shaded area shows, the focus of today's tools is on low power in the hardware implementation design enabled with CPF.

From the original equipment idea, all the way to the end, there are opportunities for power reduction. The extension of low-power design to the testing interface (shown by the right arrow), to the package design (left arrow), to software design (down arrow), and ultimately to whole system optimization (up arrow) including mechanical and PCB designs, is left as an arena for future improvement.

A holistic approach is required for the future.



## Summary

Power is immensely important to competitive success. Low-power design requires comprehensive optimization with many design trade-offs, and a wide design space must be considered. The Common Power Format (CPF) holistically combines architecture design with physical implementation, and as such, CPF provides remarkable benefit in low-power design.

Low-power design success with CPF includes:

- Significant productivity gain in high-level design and verification.
- Design cost savings with a simplified low-power SoC Design flow.

The NEC Electronics proof point project made a significant contribution to the maturity of the CPF design flow.

NEC Electronics has already started to deploy CPF in a production environment, in 2007, because it believes that a competitive advantage in low power can attract many new business opportunities, and the Common Power Format is a part of that advantage.

## Acknowledgements

*The author would like to thank Takashi Nakayama for sharing the 65nm design experience, and all PFI members at NEC Electronics and Cadence for their remarkable contribution to NEC-PPP.*

---

*Toshiyuki Saito serves as a senior manager of the Design Engineering Development Division at NEC Electronics Corporation. He has managed many foundation technology development projects for advanced LSI design, including the UltimateLowPower™ design technology for 65nm mobile SoCs. He received his M.S. degree in computational physics of materials from Kanagawa University in 1984, joined NEC Corporation, and moved to NEC Electronics Corporation in 2002 when the company was established through a spinoff of the semiconductor business operations from NEC Corporation.*

## FUJITSU: CPF in the Low-Power Design Reference Flow

By Tsutomu Nakamori, Manager of Low Power Technology Development at Fujitsu.  
© Fujitsu 2008.

Headquartered in Tokyo, Fujitsu specializes in semiconductors, computers (supercomputers, personal computers, servers), telecommunications, and services. Established in 1935 under the name Fuji Tsūshinki Seizō, Fujitsu today employs around 161,000 people and has 500 subsidiary companies.

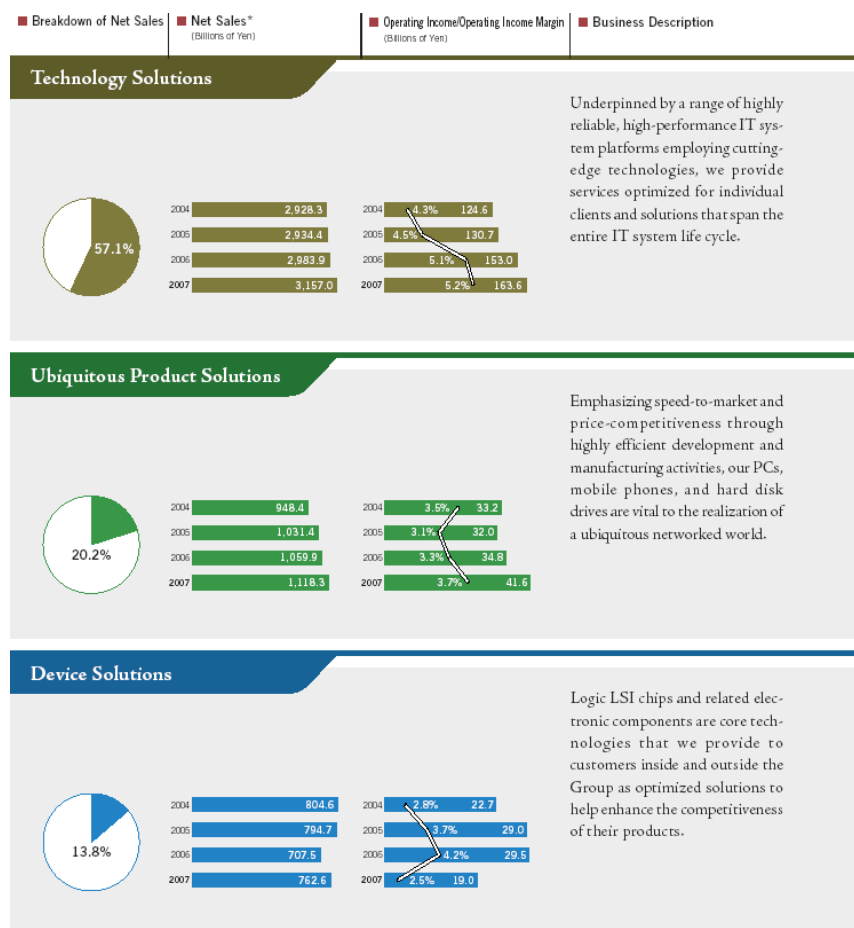


Figure 103. Three key business units (Ref. 17)

Fujitsu's device solutions segment made up about 15.8% of total sales in the first fiscal half of 2007 (April-September), amounting to ¥397.9B (US\$3.46B). Fujitsu has announced that the LSI business will become Fujitsu Microelectronics, Limited, a wholly owned subsidiary, in late March 2008.

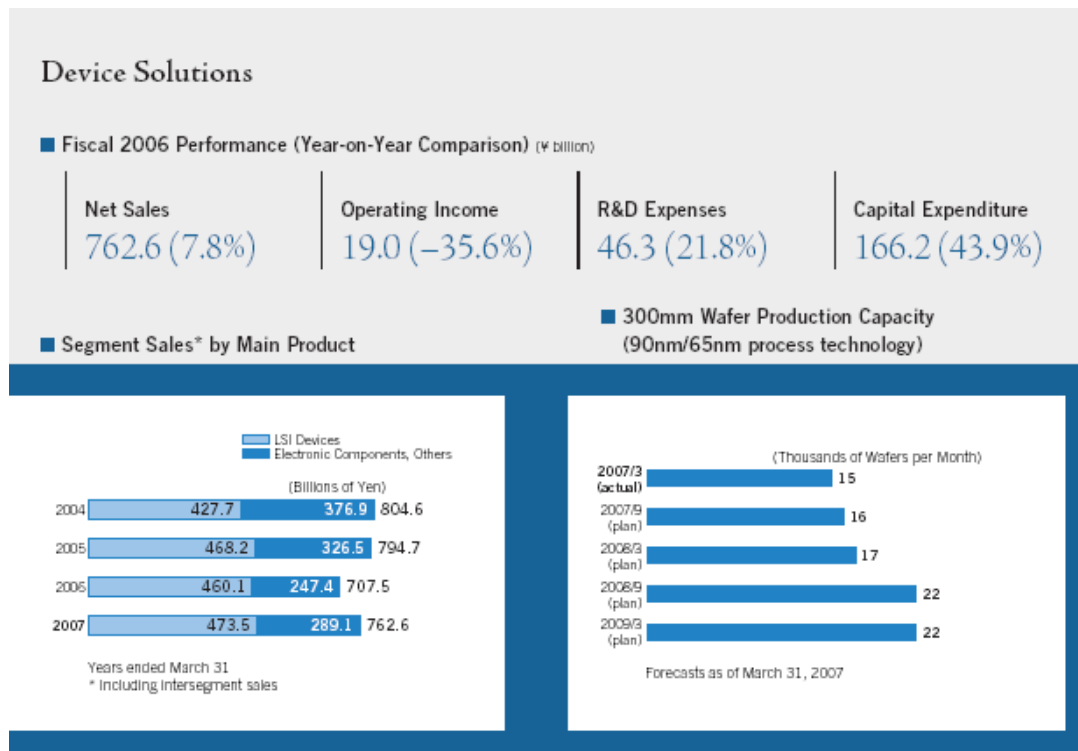
Fujitsu Microelectronics provides high-quality, reliable semiconductor products and services for the networking (metro, enterprise, access and wireless), automotive, consumer, industrial, security and other markets. To meet customer requirements of the complexities of deep submicron process technology and compressed time-to-market schedules, Fujitsu has developed an Integrated Device Manufacturing business model. IDM provides specific services, ranging from flexible design methodologies to a comprehensive set of IP macros, as part of development alliances tailored to customer needs.



**Figure 104. Examples of Fujitsu Electronics Devices offerings ([Ref. 17](#))**

Fujitsu MicroElectronics and Electronics Devices group includes offerings of:

- ASICs, including standard cell, embedded array and gate array
- IP macros
- MPU/MCUs and development environments
- ASSPs, including WiMAX, IDB-1394, communications ICs, video ICs, power management ICs
- System memory, including Flash, FRAM, FCRAM
- Media devices
- Electromechanical components
- Optical components
- Wafer foundry services
- Packages



**Figure 105. Key parameters for the Device Solutions business (Ref.17)**

Both designers within Fujitsu and Fujitsu customers for ASIC, IP, MPU/MCU and ASSPs agree: power is becoming the predominant challenge across a variety of electronics products.

Low power is also important to the Stage V Environmental Protection Program within Fujitsu, which includes improving the environmental value of products and services by increasing the number of Super Green products:

- Increasing the number of Super Green products with top-class environmental characteristics by over 20% by the end of 2009
- Achieving an environmental efficiency factor of 2x relative to products in fiscal 2005, across all business units

### **Fujitsu and CPF**

Because of the importance of low-power design to Fujitsu Microelectronics and its customers, and recognizing the necessity and efficiency of CPF, Fujitsu was a founding member of the Power Forward Initiative. Fujitsu both contributed to the formation of the CPF specification, and was an early adopter of the CPF flow.

- In January 2007, Fujitsu started to build a CPF-based flow
- Fujitsu mounted a proof point project chip design as a test before incorporating the standard into its recommended Reference Design Flow (RDF) 3.0

- In June 2007, Fujitsu taped out the complex, real-world low-power chip using CPF. The SoC, discussed below, worked completely with no respin, validating the reliability of the CPF-based design flow
- Fujitsu announced their RDF 3.0 ASIC/ASSP design flow in July 2007. It is the first CPF-based flow in the world

Fujitsu is also a member of STARC, which participates in the Si2 Low Power Coalition, promoting CPF. STARC announced in January 2008 that it has seen up to 40% power reduction achieved in the CPF-based design flow, PRIDE 1.5. ([Ref. 18](#))

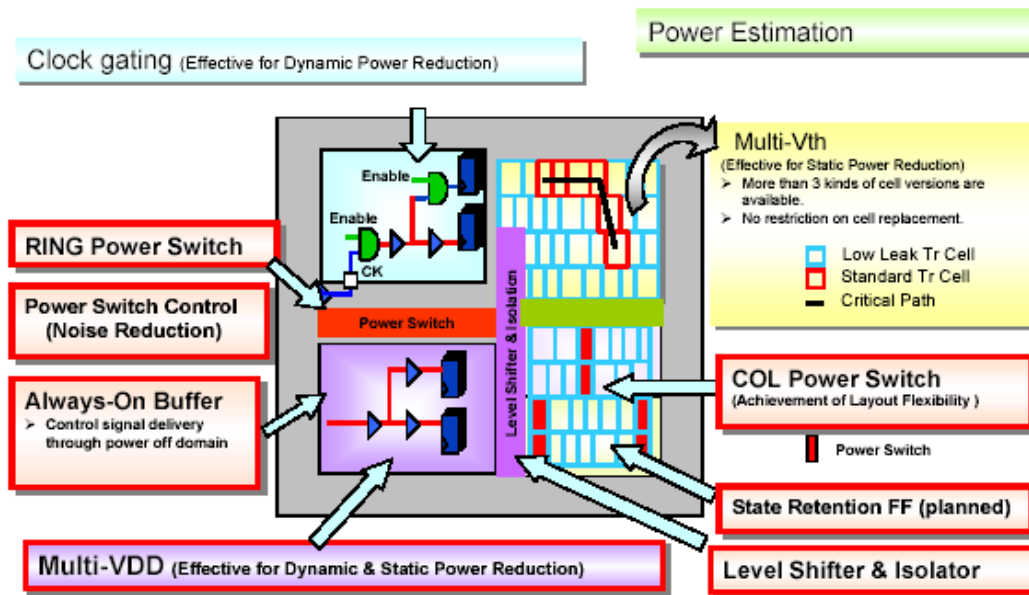
### **Low-Power Design Techniques Used by Fujitsu**

Fujitsu uses a variety of design techniques for power reduction across their spectrum of product offerings. These approaches and techniques have both benefits and penalties in the design and implementation of the SoC.

A table lists the common power management techniques and their impacts:

Power Management Approach / Technique	Power Impact	Penalty
<i>Power estimation for early power analysis and architectural exploration</i>	<i>May be dynamic and static reduction</i>	<i>Time</i>
<i>Multi-V<sub>t</sub> with &gt; 3 cell libraries</i>	<i>Static power</i>	<i>Low; automated</i>
<i>Clock gating</i>	<i>Dynamic power</i>	<i>Low; automated</i>
<i>Multiple supply voltages (MSV)</i>		<i>Complexity, timing, area</i>
<i>Dynamic voltage / frequency scaling (DVFS)</i>		<i>Complexity, timing, area</i>
<i>Power shutoff (PSO)</i>	<i>Static power</i>	<i>Complexity, timing, area</i>
<i>Adaptive voltage scaling (ASV)</i>		<i>Complexity, timing, area</i>

The following block diagram illustrates some of the simple, and advanced, power management techniques commonly used by Fujitsu. As you can see, power management can introduce complexity by requiring additional power and control signal routing, clock gating schemes, power switches, isolation cells, level shifters, state retention cells, and always-on buffers.



**Figure 106. Power reduction techniques in a challenging low-power design**

### **Low-Power Test Chip Developed with CPF**

The low-power test chip developed by Fujitsu represents a complex, multi-processor SoC for mobile applications.

It was designed in a 90nm process, with 7 layers of metallization and was fabricated at the Mie Plant, one of Fujitsu's fabrication plants located in the Mie prefecture in central Japan.

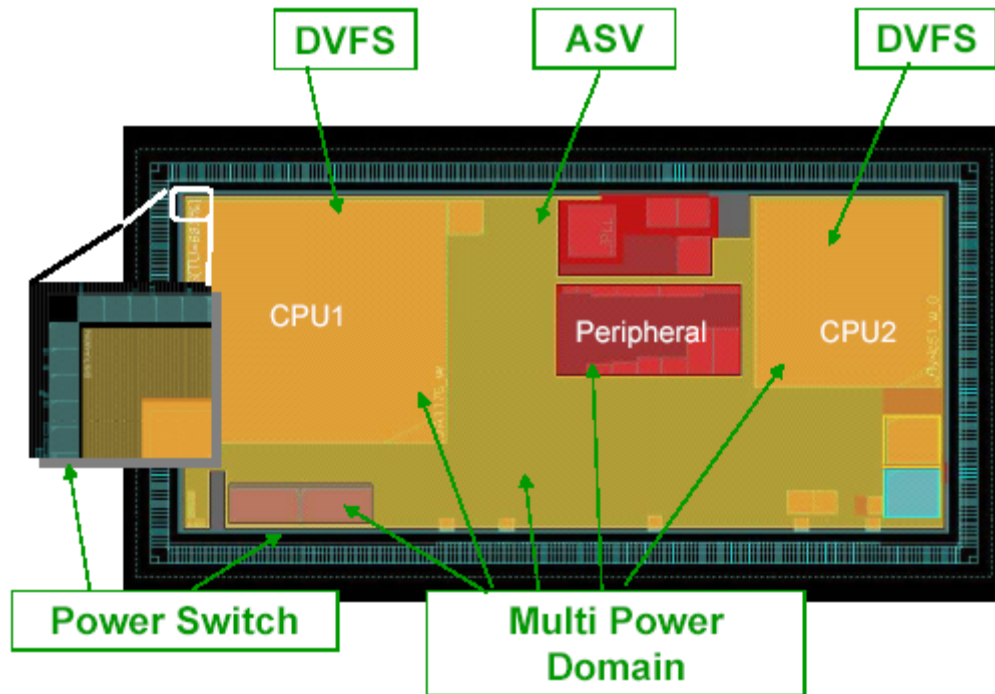


**Mie Plant for Advanced Logic LSI Production**  
The Mie Plant produces LSI devices on 300mm wafers. Fab No.1 has been onstream since April 2005, and Fab No. 2, which utilizes 65nm process technology, became operational in April 2007.

**Figure 107. One of Fujitsu's fabs for 65nm**

The design includes 940K instances for a rough gate count of 4M gates, and 1.7Mbytes of memory. It offers audiovisual peripherals, and a monitor to improve noise. It contains both the ARM 11 core (CPU1 in the below diagram) with Intelligent Energy Manager (IEM) and a Fujitsu CPU core, the FRV (CPU2.)

This advanced low-power design incorporates 11 power domains and 19 different power modes: a real challenge for an automated flow from power intent through verification, synthesis, test synthesis and physical implementation!



**Figure 108. Implementation of the CPF low-power test chip**

### **Low-Power Design Flow with CPF**

The low-power design flow upon which this SoC was developed, and which is incorporated in the Reference Design Flow, is diagrammed below.

The flow included:

- Cadence RTL Compiler, as well as other synthesis tools
- Cadence Incisive Unified Simulator and Conformal-LP for verification
- Fujitsu intelligent power switch capability, which estimates the requirements for power switches before implementation with attention to noise reduction. The tool inserts the right number of power switches to reduce noise while preserving performance, mitigating the effects of rush currents. This software includes analysis and parameter tables which are based on Fujitsu's knowledge of process/voltage parameters, switching times, cell and gate requirements, and empirical results from previous chips
- Cadence SoC Encounter for power-aware physical implementation - This includes an automatic always-on switch insertion capability, which was developed in cooperation between Fujitsu and Cadence and is now part of Encounter
- A Fujitsu proprietary power switch insertion tool, which supports non-rectilinear shaped physical power domains, with an easy graphical user interface (GUI) and full integration with Encounter



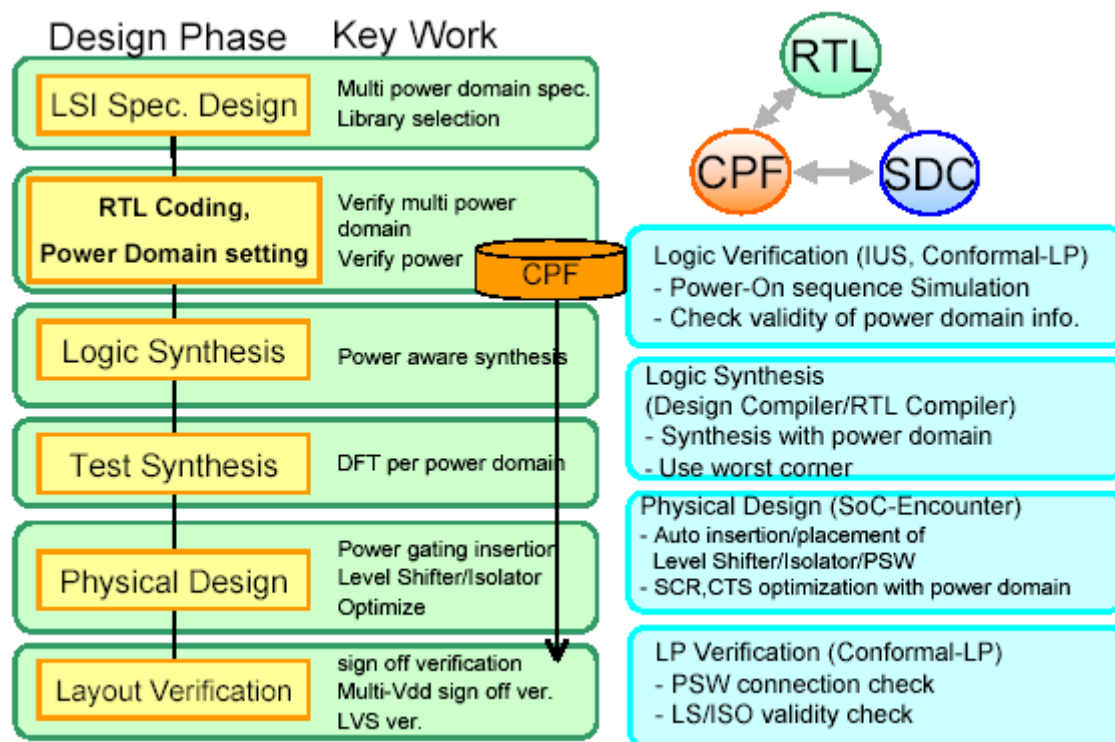


Figure 109. CPF low-power test chip flow

### Review of Low-Power Test Chip Design

The complex test chip, with 940K instances, 11 power domains and 19 different power modes, taped out successfully in June of 2007, validating both CPF and the tools flow, satisfying the objective of the Fujitsu proof point project.

Power savings realized for this SoC included 35% operating power reduction, with standby power reduced by a factor of 100-1000.

Low-power techniques included:

- Multi-Vt with > 3 cell libraries
- Clock gating
- Multiple supply voltages (MSV)
- Dynamic voltage / frequency scaling (DVFS)
- Power shutoff (PSO)
- Adaptive voltage scaling (ASV)

Verification summary:

- CPF can be applied to the design flow

- Modification of RTL is not necessary
- On-chip power gating and multi-supply, multi-voltage (MSMV) power-reduction techniques can be easily implemented
- Power shut-off states can be handled in RTL simulation
- Level shifters, isolation cells, power switch and always-on buffers are automatically inserted
- Design can be verified with low-power design rules

Statistics for the advanced low-power design show that power reduction techniques with CPF produce excellent results for area, which translates into cost savings, preservation of performance, and superior engineering productivity:

Design Parameter	Without CPF	With CPF
Area penalty / cost of silicon	Varies widely depending on engineering expertise.	The area penalty, including all the low -power techniques, was less than 2%.
Timing / performance	Risk of performance impact.	There was no significant impact on timing design or performance.
Productivity	Months of additional engineering effort for manual implementation of low-power techniques, verification; still high risk.	Design cycle was extended by only 2 to 4 weeks (mainly logic design and verification) to incorporate all the power management techniques. Working silicon!

So, RDF offers a large savings over trying to implement power reduction manually without CPF!

### ***Fujitsu Reference Design Flow 3.0: Low Power with CPF***

In technical alignment with this success, Fujitsu prepared its RDF 3.0 incorporating the CPF standard.

The following diagram shows the high-level design methodology.

## Supporting Multi-VDD and Power Gating

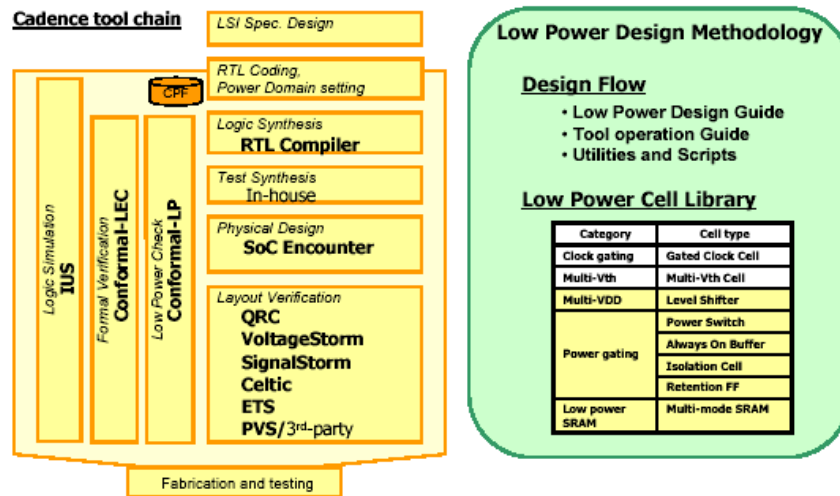


Figure 110. High-level viewpoint: Fujitsu RDF 3.0 with CPF

Now, let's take a closer look at each stage of the design and implementation flow used for the chip, and recommended in the Fujitsu Reference Design Flow.

### Front End Design

The following diagram shows the detail of front end design utilizing multiple low-power design techniques, starting from RTL and including synthesis, verification of both RTL and CPF, and test synthesis.

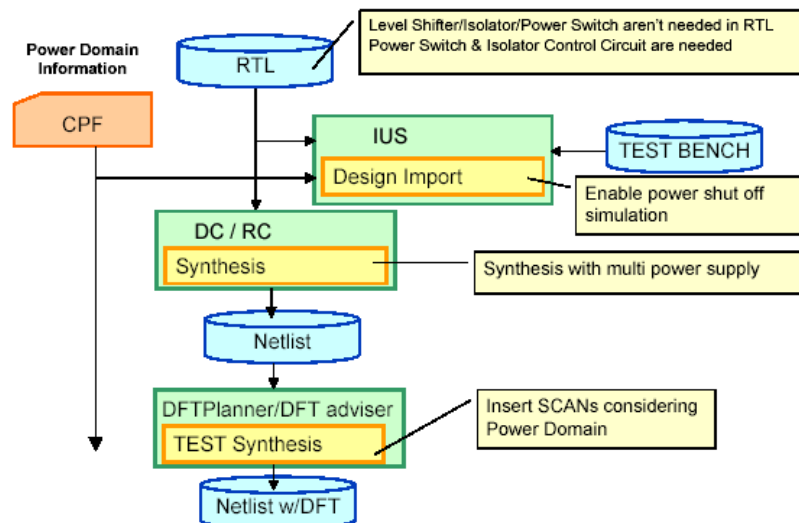


Figure 111. Front-end design flow with CPF in the Fujitsu RDF 3.0

## Floor Plan

The following diagram illustrates how SoC Encounter imports the design and CPF, performs floor planning, defines multiple power domains, implements multi-Vt, and inserts power gating elements such as level shifters, isolation cells, power switches, and power routing for the complex design.

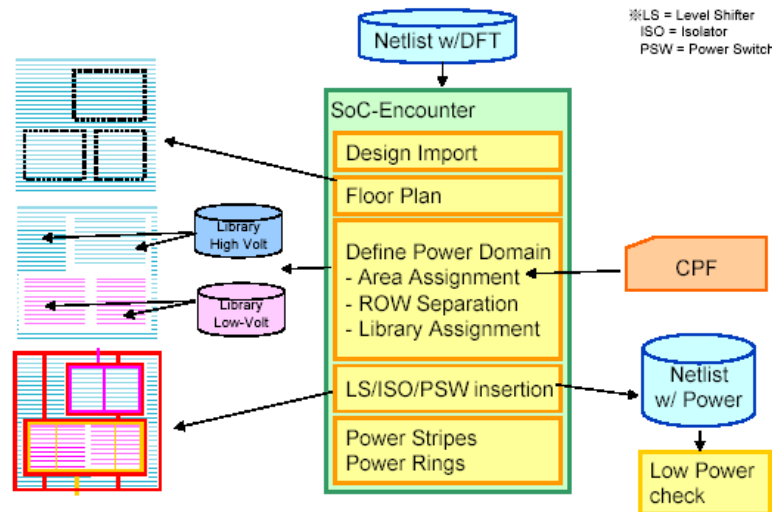


Figure 112. Floorplanning with CPF in the Fujitsu RDF 3.0

## Low Power Check

The following diagram shows how Cadence Conformal checks the low-power design intent and how it is implemented in RTL and CPF files, identifying any issues with power switches, level shifters, and isolation cells.

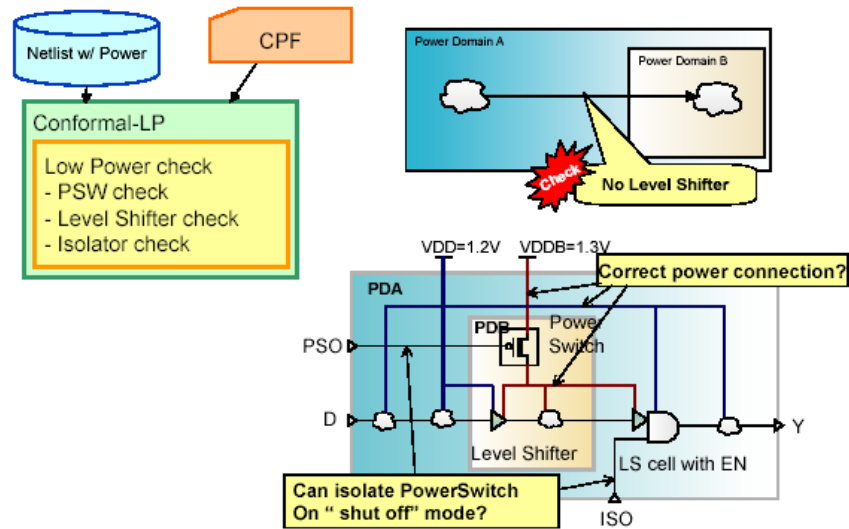


Figure 113. Checks with CPF in the Fujitsu RDF 3.0

### Placement and Clock Tree Synthesis (CTS)

The following diagram pertains to placement of cells for level shifting and isolation, plus power routing for multiple domains, and optimized clock tree synthesis with always-on buffer automatic insertion, based on CPF.

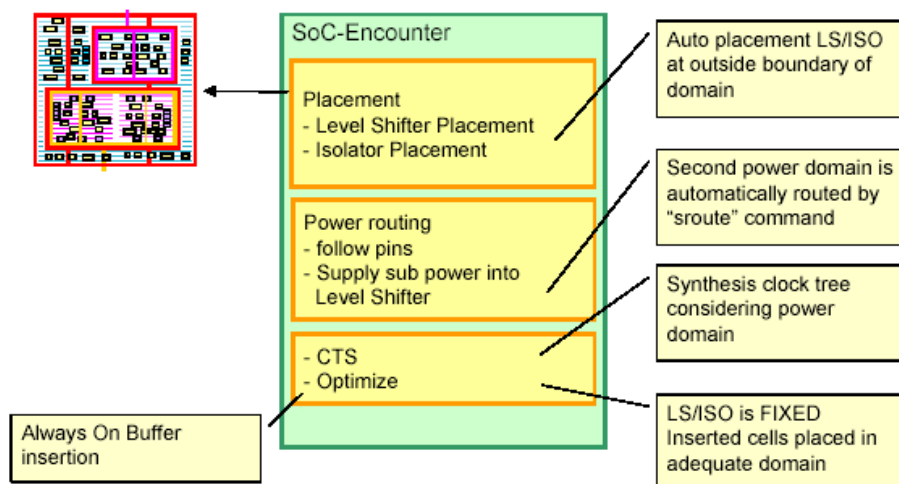
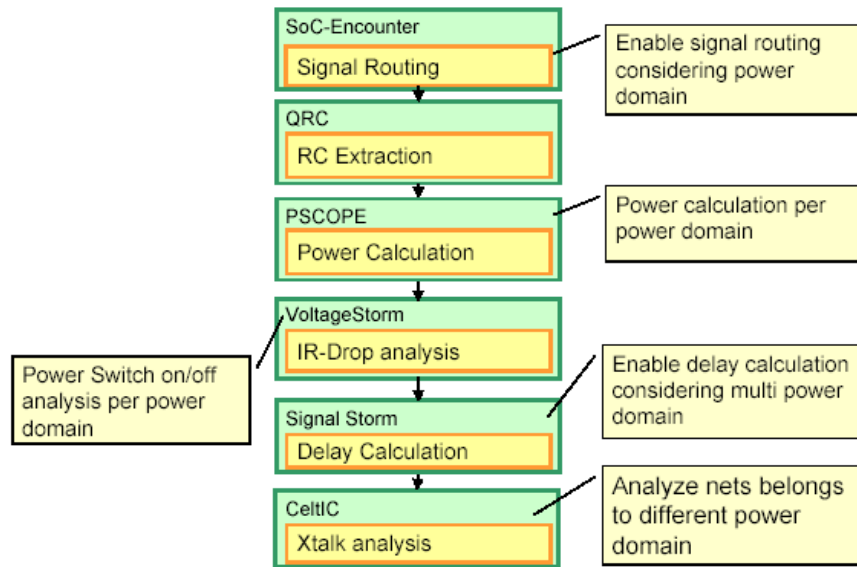


Figure 114. Placement and clock tree synthesis with CPF in the Fujitsu RDF 3.0

## Routing and Analysis

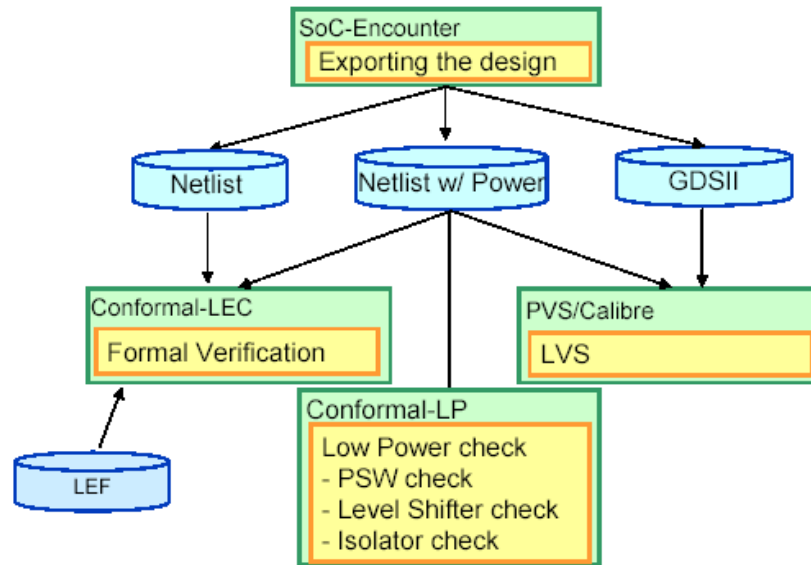
The following diagram shows how signal routing is performed, with accurate RC extraction and power calculation based on the physical layout. Accurate delay calculation across multiple power domains is performed, along with noise analysis, IR drop and cross-talk: all aware of the power domains and low-power techniques described with CPF and carried throughout the flow.



**Figure 115. Routing and analysis with CPF in the Fujitsu RDF 3.0**

## Physical Verification

The following diagram illustrates the final verification steps, based on exporting the design information from SoC Encounter and utilizing Conformal for a variety of formal verification and low-power checks. The flow also includes accurate device parameter extraction.



**Figure 116. Physical verification with CPF in the Fujitsu RDF 3.0**

### **Fujitsu's CPF Low Power RDF Methodology**

The design flow using CPF is currently available from Fujitsu and includes the following features and differentiated technology.

#### **Features**

- Low-power cells have been prepared. Available low-power cells and IP include:
- Power switches, isolation cells, always-on buffers, level shifters
- Power management unit
- SRAMs with low leakage standby mode
- Multi- $V_{DD}$  and on-chip power gating is available
- Fujitsu original power switch insertion tool is part of the flow: flexible and easy to use with GUI
- Automatic always-on buffer insertion was developed in conjunction with Cadence and is incorporated into Encounter
- Fujitsu power switch noise reduction utility is available: straightforward power switch optimization to minimize power noise while preserving timing
- CPF description guidelines for ASIC/ASSP customers is available from Fujitsu ([Ref. 19](#))
- Also CPF hand off guideline document is available to help accelerate the handoff between Fujitsu and its ASIC/ASSP design customers ([Ref. 20](#))

#### **Expectations for CPF Evolution**

Fujitsu, in conjunction with Cadence and other members of the Power Forward Initiative, is committed to extending the low-power flow. In addition to current

capabilities, for emerging technologies, early-stage power architecture exploration is an interesting area with a potentially powerful impact on power reduction.

## **Summary**

Fujitsu has an initiative to continue to strengthen the ASIC/ASSP and standard product businesses. In particular, we will enhance our lineup of distinctive products for mobile devices built on low-leak, energy efficient technologies. ([Ref.17](#))

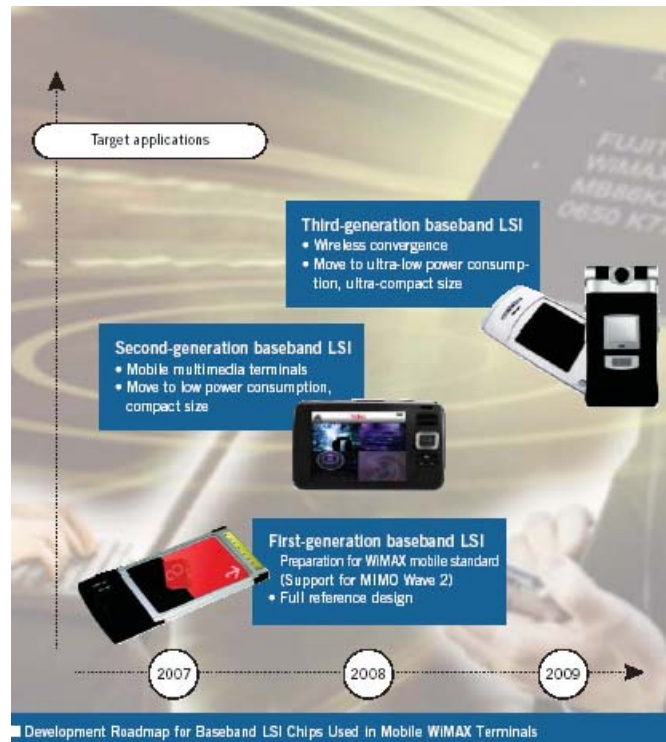
## **Status of CPF-based Design Flow**

- Fujitsu was a founding member of the Power Forward Initiative. Fujitsu both contributed to the formation of the CPF specification, and was an early adopter of the CPF flow
- In January 2007, Fujitsu started to build a CPF-based flow, translating the CPF low-power design flow into the production-worthy Fujitsu RDF 3.0
- Fujitsu mounted a proof point project chip design as a test before incorporating the standard into its recommended Reference Design Flow (RDF) 3.0
- In June 2007, the mobile applications chip previously discussed was fabricated and tested with results of 100% functionality, no re-spins. Operating power was reduced by 35%, and standby power reduced by a factor of 100-1000
- Fujitsu announced its RDF 3.0 ASIC/ASSP design flow in July 2007. It is the first CPF-based flow in the world
- Several additional designs have already started with the RDF 3.0 flow

## **Fujitsu Proprietary Designs and CPF**

Fujitsu Microelectronics will use CPF internally for SoC design across multiple design groups and products. Fujitsu itself represents 30-40% of the ASIC/ASSP designs done by the Microelectronics group. RDF 3.0 will be of particular benefit in the very power sensitive markets and technologies such as the WIMAX product lines.





**Figure 117. Fujitsu WIMAX technology roadmap**

## Fujitsu ASIC/ ASSP Customers and CPF

Fujitsu has also adopted CPF in their Reference Design Flow 3.0 for the benefit of their customers. The vast majority of Fujitsu Microelectronics customers are concerned about power, and many customers, especially in Japan and Asia, have expressed strong interest in using our low-power techniques and design flow.

Today, over 30% of customers are adopting the following reduction techniques now, and this is expected to ramp up in the future:

- Multi-Vt
- Clock gating
- Multiple supply voltages (MSV)
- Dynamic voltage/frequency scaling (DVFS)
- Power shutoff (PSO)
- Adaptive voltage scaling (ASV)

CPF accelerates adoption of advanced techniques like power shutoff, multi-supply voltages, dynamic voltage and frequency scaling, in an automated fashion, with low risk and high engineering productivity.

Fujitsu expects the low-power design flow based on CPF to offer competitive advantage to their customers for devices, IP, design services and electronic products.

---

*Tsutomu Nakamori is the Manager of Low Power Technology at Fujitsu, in the Technology Development Division, in Kawasaki, Japan. Nakamori has worked on the development of SoC design methodology since 1995, and low-power technology since 2005. He is also the chairman of the Power Format Study Working Group within the Japan Electronics and Information Technology Industries Association (JEITA, [www.jeita.or.jp](http://www.jeita.or.jp)). Nakamori joined Fujitsu in 1980.*

## NXP User Experience: Complex SoC Implementation with CPF

*By Herve Menager, Architect, SoC Design Technology, NXP Semiconductors.*

Founded by Philips, NXP is a top-10 semiconductor company creating semiconductors, system solutions and software that deliver better sensory experiences in mobile phones, personal media players, TVs, set-top boxes, identification applications, cars and a wide range of other electronic devices.

<i>Established:</i>	<i>2006 (formerly a division of Royal Philips Electronics) 50+ years of experience in semiconductors</i>
<i>Headquarters:</i>	<i>Eindhoven, The Netherlands</i>
<i>President &amp; CEO:</i>	<i>Frans van Houten</i>
<i>Business Units:</i>	<i>Mobile &amp; Personal Home Automotive &amp; Identification Multimarket Semiconductors NXP Software</i>
<i>Net sales:</i>	<i>€ 4.96 billion in 2006</i>
<i>Sales by region:</i>	<i>22% China 16% Netherlands 12% Singapore 10% USA 7% Taiwan 7% South Korea 5% Germany 21% Other</i>
<i>Research &amp; Development:</i>	<i>Investment of about € 950 million 7,500 engineers 25,000+ patents 26 R&amp;D centers located in 12 countries Participation in over 75 standardization bodies &amp; consortia. Strong links with universities.</i>
<i>Employees:</i>	<i>Approximately 37,000 people in more than 20 countries: 37% Europe 37% Asia 21% Greater China 5% Americas</i>
<i>Manufacturing facilities:</i>	<i>15 manufacturing sites worldwide: 7 wafer fabs: Caen Fishkill Hamburg Hazel Jilin Nijmegen Singapore 8 test and assembly sites: Bangkok Cabuyao Calamba Guangdong Hong Kong Kaohsiung Seremban Suzhou</i>
<i>Customers:</i>	<i>50+ direct customers accounting for approximately 70% of sales. Customers include Apple, Bosch, Dell, Ericsson, Flextronics, Foxconn, Nokia, Philips, Samsung, Siemens and Sony.  30,000+ customers reached via NXP's semiconductor distributor partners, including Arrow, Avnet, Future, SAC and WPI.</i>

**Figure 118. NXP facts and figures**

Worldwide leadership positions include the following, by business unit:

<i>Mobile &amp; Personal</i>	<i>NXP provides complete entry-level to high-end system solutions for mobile phones that enable handset manufacturers to rapidly deliver highly featured and reliable products to the market. NXP's solutions cover a wide range of current and upcoming telecom standards — EDGE, 3G and 4G — and seamlessly share content through a wide range of connectivity interfaces, such as Bluetooth and (wireless)USB, and even allow mobile payments using Near Field Communications (NFC).</i>	<i>More than 250 million Nexperia cellular system solutions shipped #1 in mobile phone speakers #1 in FM radio ICs for portable applications #1 in USB for mobile and portable applications #2 3G RF Leading the market with several industry firsts in TD-SCDMA development</i>
<i>Home</i>	<i>NXP's Nexperia-based Home system solutions and audio/video components enable manufacturers to offer consumers more digital content via a better viewing and listening experience. The Home business unit innovates embedded multimedia features and next-generation, connected multimedia appliances for a connected living experience — making it easier than ever to enjoy and share multimedia content, anytime and in every room.</i>	<i>1 out of 2 TVs worldwide contains an NXP Chip 4 in 10 PC TVs use NXP silicon tuners #1 in TV reception tuners #1 in RF front end modules for digital terrestrial set top boxes Created Nexperia PNX5100, the world's first video postprocessor with Motion Accurate Picture Processing technology</i>
<i>Identification</i>	<i>NXP's contactless technologies are designed to track inventory, improve logistics and protect people's information-driven lives. NXP technologies can be found in everything from Radio Frequency Identification (RFID) tags that authenticate medicines, to e-ticketing systems that cut commute times and e-passports that fight identity theft and increase border security. In particular, Near Field Communication (NFC), a technology NXP co-developed, gives instant yet completely secure access to entertainment, information and services.</i>	<i>#1 in NFC (Near Field Communication) #1 in RFID (Radio Frequency Identification) solutions: over 2 billion ICs shipped #1 in e-passports with 80% of the world's e-passports using NXP ICs 80% of all electronic tickets in public transport use NXP ICs</i>
<i>Automotive</i>	<i>NXP's Nexperia-based processors for automotive offer the same incredible sights and sounds the consumer expects at home, with seamless connectivity to personal media players. NXP's in-vehicle networking technologies like FlexRay make cars more responsive and safer to drive while the RF-based car access solutions are helping to put car thieves out of business.</i>	<i>#1 in car radio tuners #1 in Digital Signal Processors for car radios #1 in automotive networking #1 in system solutions for automotive immobilizers and keyless entry/go</i>
<i>Multimarket Semiconductors</i>	<i>NXP has one of the largest portfolios of multimarket semiconductors in the industry, from basic building blocks like timers and amplifiers to sophisticated ICs that improve media processing, wireless connectivity and broadband communications. These are designed to save space, extend battery life, enable customized solutions tailored to customers' needs, and make it easy to implement last-minute changes.</i>	<i>#1 in 32-bit ARM-based microcontrollers #1 in I<sup>2</sup>C-logic and industrial UARTs 1 out of 2 laptops uses NXP's GreenChip power supply controller #2 in small signal discretes and standard logic worldwide</i>
<i>NXP Software</i>	<i>NXP Software is a fully independent and leading provider of innovative multimedia</i>	<i>Independent Software Vendor for mobile multimedia software solutions</i>

	<i>software solutions focused on enhancing the User Experience, reducing cost and improving time to market for device makers.</i>	<i>More than 250 million devices use LifeVibes software</i>
--	---	---

**Figure 119. NXP business units**

## **Low Power is Critical to NXP**

For NXP designs across all business units, at the device and system level, total power is important— both operating (or active) power and leakage power.

## **Low Operating Power is Important**

For cost-sensitive battery-operated devices with no standby mode, the convergence of computing, communication and entertainment increases the complexity of SoCs, and require higher-level silicon integration. Yet in spite of these challenges, the market expects and demands longer battery life. Also, cost of goods is a critical concern, and exotic heat-dissipating packages are costly. ([Ref. 21](#))

Home consumers who want electronic products that enhance the user environment insist on reduced noise (which means no fans) and cool-running products (again, requiring lower power dissipation.)

To meet these requirements, NXP is addressing low operating power at all design levels: transistor, logic, RTL, interconnect, architectural, and system.

## **Low Leakage Power is Important**

For handheld devices with stand-by requirements, technology and the market combine to create the “Perfect Storm.” Customers want smaller, cooler mobile devices at lower cost. This again leads to both a dramatic increase in functionality and complexity and higher demands on standby battery lifetimes.

Achieving the high levels of silicon integration required for these devices means using advanced processes, but unfortunately, advanced processes have inherently higher leakage current. This creates a challenge that must be addressed by both process and design.

Leakage power can be addressed through choice of process, library options, transistor thresholds, design techniques, and other solutions.

## **NXP and CPF**

NXP early recognized the need for an industry initiative to improve low-power flows, and began work on the Common Power Format (CPF) in early 2006. NXP was a founding member of the 26-company Power Forward Initiative (PFI) to drive

direction and standardization of CPF. NXP was also an early member of the 18-company Low-Power Coalition (LPC) under Si2, which approved CPF as an Si2 standard in early 2007. ([Ref. 22](#))

In the larger sense of power and energy consumption and its impact on our environment, NXP also believes in taking corporate social responsibility, and has taken concrete steps and set clear goals in environmental impact for SoCs and electronic products.

### ***Low-power implementation trends***

Previously, at less aggressive complexity and process nodes, NXP SoC designs avoided undue risk and complexity by primarily using the simple, reliable available power management techniques, which were cleanly supported by existing individual tools. So, among other techniques, for dynamic power, designers:

- Reduced power dissipation sources when not needed
- Gated clocks
- Minimized switching capacitances
- Used synchronous circuits such as handshake protocols

And for leakage power, the approach involved:

- Used multi-Vt synthesis and optimization at the physical level

Still, these common power reduction techniques were not enough to meet our power goals.

More recently, we introduced aggressive, state-of-the-art techniques to control active and leakage power.

For dynamic power, to meet both chip performance requirements and operating power goals, NXP designers used:

- Voltage islands (MSV)
- Dynamic voltage and frequency scaling
- Adaptive voltage and frequency scaling

For leakage power:

- Suppressing current when not needed through power shutoff modes

At the design level, however, without CPF, these advanced techniques can increase risk due to manual intervention in the design, reduce engineering productivity, increase complexity, slow time to market, and create timing and area problems. ([Ref. 23](#))

They are not only intrusive on the functionality of the SoC but also impact the entire design flow—from synthesis through verification and physical implementation.

Without CPF—with manual intervention in flows—NXP’s past SoC designs identified the following challenges and limitations in MSV SoCs: ([Ref. 24](#))

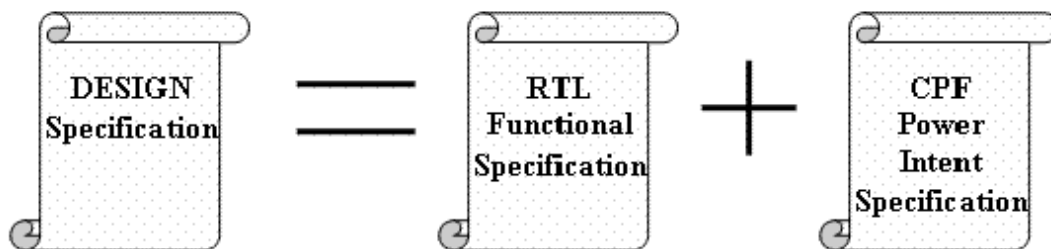
- No placeholders for power and ground nets
- No way to describe power specifications and constraints: power information is sometimes available as a paper specification, but often exists only in the SoC architect's mind, as it is not usually explicit in functional descriptions
- Recurrent specification of the same power intent for each tool in the design flow
- No flow to verify power modes and power sequences in functional simulation
- Increase of STA sign-off cases
- Vast increase of SDF simulation cases
- No reusability of IP with multiple power domains in SoCs
- Tremendous increase of implementation throughput time due to lack of automation
- Complex signal distribution
- Complex power grids
- Design for test (DFT) complexity

Specifying intent for automated implementation and verification is very complex, and the total problem is more than the sum of its parts.

### ***Why a Common Power Format?***

CPF solved these problems by delivering a power intent specification, separate from the functional specification. In CPF shared with RTL, both design intent and low-power intent are captured in the design specification as a power intent and functional specification pair.

CPF facilitates a golden reference design specification, with separate low-power intent information, such that early exploration of different power architectures can be done and power behavior may be changed.



***Figure 120. Functional and power intent***



Since CPF allows the same low-power intent to be shared across all design tools, and from the start of the design through implementation, it offers the following opportunities to reduce power while preserving design and implementation productivity.

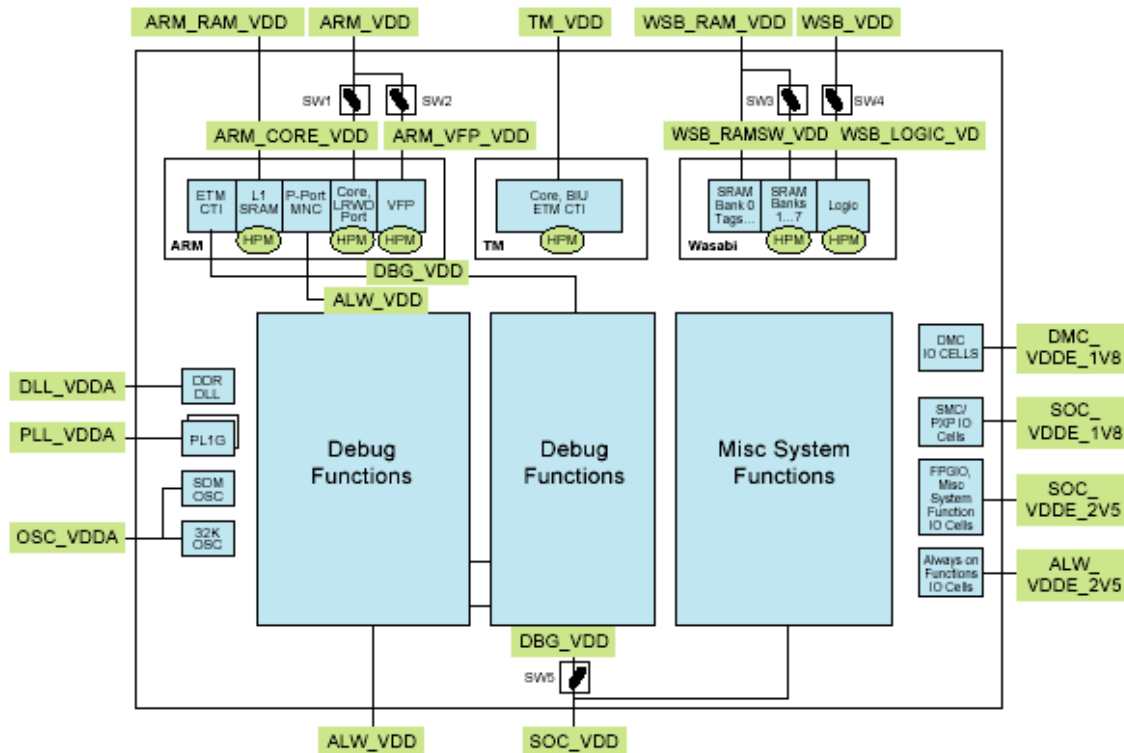
- Scalable solutions
- The ability to capture power network intent independently and throughout the tool flow
- New tool functionality with an integrated flow
  - Low-power design at RTL
  - Verification early in the design flow
  - Implementation based on the common power intent that was verified earlier
  - Validation of the power intent modeled at early design stage and the actual implementation of the power intent
  - DFT
- Support for advanced techniques
  - Voltage islands (MSV)
  - Dynamic voltage and frequency scaling (DVFS)
  - Adaptive voltage and frequency scaling (AVFS)
  - Power shutoff (PSO)
  - Low-power IP required by advanced techniques: level shifters, isolation clamps, on-chip switches, state retention cells
- IP and design reuse and portability

The following SoC design example illustrates how NXP implemented several of these power reduction techniques with a CPF-enabled flow, and summarizes the user experience and results NXP obtained.

### ***CPF in Action on a Complex SoC Platform***

NXP developed a complex SoC that challenged the current architecture and implementation flow, as a CPF proof point project, and has regularly reported to the Power Forward Initiative on the status and progress over the last year. The CPF standard published by Si2 ([Ref. 23](#)) was the industry's first power format to have tool support, with power intent captured in CPF and functionality in RTL description. This allowed the simple migration of a non-power-aware RTL design to a power-aware RTL design.





**Figure 121. Complex platform SoC (Ref. 21,25)**

This complex MSV NXP SoC incorporates 11 voltage islands.

- There are 3 voltage-scalable logic sections, 3 on-chip switchable domains, 5 off-chip switchable domains and separate switchable pad ring sections
- The three major power consumers (RISC CPU, VLIW DSP and L2 System Cache) are controlled using DVFS
- High-bandwidth expansion ports enable the platform to be extended, for example, with graphics or cellular modem subsystems

The die size of the chip is 42mm<sup>2</sup> and it was fabricated in a 65nm CMOS process. (Ref. 21)

### Power Network Intent

Before CPF, and for designs without advanced power management, power and ground were traditionally defined and implemented in the layout phase, as they had no functional impact on the chip (other than being necessary!)

Now, power gating, to minimize leakage current, is making power and ground nets partly functional, since the behavior of the device depends on their state (clamping value) and level (performance.) The number of voltage islands in SoC and IP designs has increased complexity considerably.

But neither RTL nor the logical views for basic library elements (leaf cells) have implicit representation of these nets, and special handling and global connection in the back-end phase is tedious and error-prone.

With CPF, power and ground nets can be specified as part of a design's low-power description with a standardized placeholder. Power intent for the power and ground network is modeled above the physical level abstraction of the design.

The power domain partitioning of the SoC design is shown in this short extract of the top level CPF:

```
set_cpf_version 1.0
set_design e2_core
create_power_domain -name VALW -default -instances \
  {u_cl_per/u_cl_per_valw_*...}
create_power_domain -name VARM_CORE \
  -instances {u_cl_arm/u_cl_arm_varm_1...} \
  -shutoff_condition {/u_cl_per/.../arm_vocore_switch_ena}
```

At the top level, two power domains are created.

- Power domain VALW is the default power domain and is always on.
- Power domain VARM\_CORE is a switchable power domain with an associated shut-off condition. An expression specifies the condition under which the power domain will be switched off.

The RTL designers can use CPF to describe the power-up and power-down behavior, and need not understand the details of how the power domain will eventually be implemented. CPF semantics for power domains furnish the power behavior for each instance, so all instances belonging to the same power domain share the same power characteristics such as voltage, on and off, etc. Power domain semantics model the power and ground network and its connections to the instance power and ground pins.

This facilitates power-aware verification and simulation of the power-up and power-down behavior of the design.

Later in the flow, at physical implementation, designers can associate a power and ground net for each power domain, and only then are power nets actually created and associated to the correct power domains.

```
create_power_nets -nets ALW_VDD
update_power_domain -name VALW_domain -
internal_power_net ALW_VDD
```

This approach allowed NXP designers to separate the power intent from the implementation, simplifying the task of verification engineers in validating the power mode and state transitions.

### **CPF Added Value in Verification**

The NXP SoC, with 11 power islands, is representative of an increasing number of designs implemented with power shutoff, including multiple voltage islands which are temporarily powered down to reduce leakage power without affecting the functionality of the rest of the design.

Power shutoff dramatically increases the complexity of design verification, and must be addressed at the beginning of the design cycle.

Key issues that must be addressed during verification include, for power shutoff: Should an entire block be shut off, or just portions of the block? For isolation: Has logic been added which, when a block is powered down, prevents the propagation of unknown signals to the rest of the design? Are the right values forced on the inputs driven by power-down logic for this block to operate properly? For state retention: Are the values of key registers stored prior to power being shutoff? During initialization: How is a block initialized to a known state after power is restored?

### **Checking MSV Elements**

With CPF, it is now possible to identify missing level shifters and clamps, and verify power intent in the context of RTL simulation:

- Operation of the power down modes
- Clamping to the proper value(s)
- Preventing deadlock in control networks that have a number of power domain crossings
- Prevention of misplaced timeout mechanisms
- Retention, save and restore cycles
- System recovery at power-on

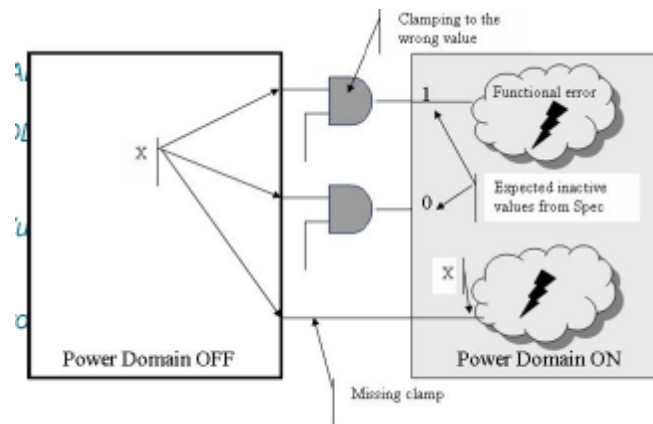
### **CPF and Simulation of Power Modes**

Before CPF-enabled flows, NXP previously verified power-down modes either using proprietary PLI/API based scripts, or by simulating with additional special standard cell libraries which modeled power state functional dependencies. These methods required manual recurrent specification of the power intent and were not easily scalable.

Software rules the verification environment of the NXP design. The approach to test was to develop self-checking test cases to drive a central power mode controller, which controls the individual power-up and power-down sequences

across various power domains of the chip. The power test cases were implemented as self-checking software running on either of the embedded CPU cores.

Incisive Unified Simulator (IUS) read in the CPF and simulated without changes to RTL. The simulator monitored power shutoff conditions with the potential to corrupt a power domain when triggered (Figure 122).



**Figure 122. Interface verification for power switching**

## Level Shifters and Clamps

In the NXP SoC, some isolation cells were inserted in RTL as opposed to using the CPF-enabled tool support. However, the insertion of isolation cells in RTL isn't possible for all paths. Since the infrastructure for production test is generated during, or following the logic synthesis process, any paths traversing power domains created for production test are not present in the RTL. So, any isolation cells in these paths must be inserted during, or after design for test (DFT) insertion. The design team used the CPF design description to insert these isolation cells during the physical implementation phase, and verified the functional integrity in simulation post-layout.

Signals driven from a powered-down block must be clamped, and floating inputs to downstream blocks which remain on must also be clamped to the proper logical values for the powered-on blocks to operate properly. Defining the proper isolation cell value requires detailed knowledge of the inactive state for each IP's input driven by a powered-down block. In the past, these values were stored in spreadsheets or other placeholders, but can now be captured in CPF when known.

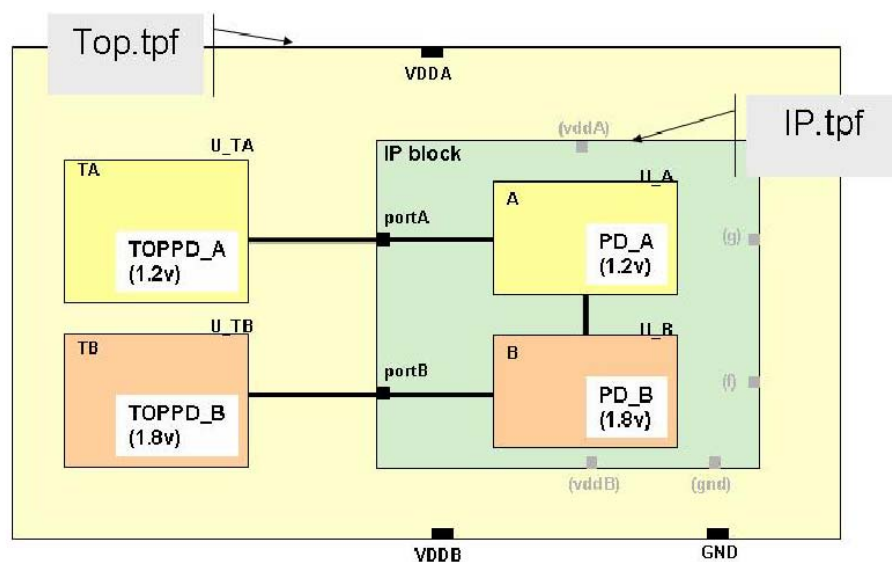
There are also challenges in identifying incorrect functional behavior in communication spanning power islands. The control network in the NXP design

enables communication between IP cells in a number of power domains, so this control network has a number of power domain crossings. If communication is attempted to an IP cell that is powered down and unable to respond, this creates the risk of a deadlock on the control network. To overcome this potential deadlock, the control network implements a timeout mechanism, which aborts the transaction if one of the parties doesn't respond. CPF-enabled simulation was proved very useful in detecting that the implementation of the timeout mechanism had been incorrectly placed in a powered-down domain, thereby disabling the timeout function itself.

NXP's experience was that the power on/off awareness and enhanced capabilities of Incisive Unified Simulator (IUS), with the CPF description of the design and HDL constructs in both Verilog and VHDL, allowed the design team to verify a range of power modes and uncovered a number of issues that would have been difficult to detect in previous designs.

### ***Hierarchical Support for IP and Design Re-Use***

To leverage IP and design re-use in an advanced, power-managed design, both tools and the format must support hierarchical usage of power intent descriptions. The CPF design flow consisting of both power intent specification and functional specification helps define a hierarchical precedence mechanism.



**Figure 123: Top-level design incorporating IP**

For bottom-up reuse, power design intent has been developed along with the functional implementation of an IP.

For soft IP, it must be reusable for the integration of the IP without having to rewrite the intent specification of the entire SoC. In the case of hard IP, the power intent must be derived from the IP implementation, and this description must also be usable in order to give IP visibility from the chip level for integration.

For top-down constraint of lower-level IP implementation, the chip-level power design intent is created. Lower-level blocks must have their power design constraints derived from this chip-level description. The chip-level context must also be visible during IP implementation, so that IP implementation is done with the knowledge of the power domains, including both external boundary level power domains and state conditions.

All of this can be done with CPF while staying at the abstract level, without doing manual design or floor planning, and without specifying the IP instance by instance.

### **Scalable Implementation**

In the past, ad hoc manual approaches to low-power design lacked a holistic view and increased design and implementation time. NXP had in some cases experienced 2X productivity drop for the back-end implementation phase, with the manual approach. This productivity penalty was due to lack of tool functionality and the lack of scalability of implementation of voltage islands, for instance:

- Interface logic, whether isolation gates for power switching or level shifters for voltage scaling, introduces verification challenges. Checks must be run to verify proper isolation, proper connectivity to the right power domains, proper partitioning of the netlist, and proper behavior of the interface
- Level shifters, which are standard cells operating with two voltage supplies, create a constraint for the layout implementation
- Always-on logic resulting from buffering of control logic for retention or global nets in power down blocks requires proper connection of their supplies
- Voltage islands and on-chip switches create a challenge for power distribution and limit floorplan alternatives and flexibility. More effort is necessary for connecting power sources to the voltage domains
- Communication between voltage islands may create logical paths spanning power domain boundaries, increasing the number of corners and modes, and the number of STA runs

To alleviate these problems, CPF-based tools that understand the same power design intent can automate many of the manual tasks. Two examples of improvements introduced by CPF follow:

### **Power Logic Insertion**

CPF describes rules governing interfaces between different power domains by adding isolation rules and/or level shifter rules only once. The CPF specification

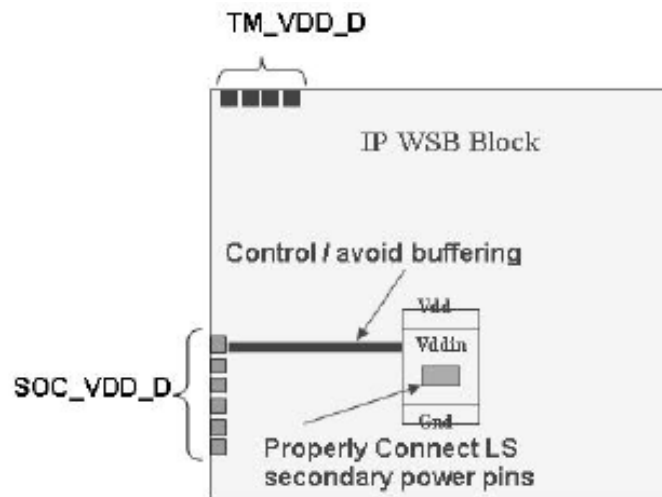
below defines from / to rules for signal interaction between power domains at a high level of abstraction, instead of requiring designers to describe them in an instance-specific way.

```
create_isolation_rule --name SOC_VDD_domain_to_Others\  
  -from SOC_VDD_domain \  
  -to {ALW_VDD_domain WSB_VDD_domain TM_VDD_domain  
      ARM_CORE_VDD_domain ARM_RAM_VDD_domain  
      ARM_VFP_VDD_domain}\  
  -isolation_output low\  
  -isolation_condition  
    {lu_e2_pinmux/e2_core_inst/u_cl_per/u_cl_per_valw_2/ip_pmc_1_vsoc_clamp_ena_n}\  
  -exclude $chiptlet_inputs  
  
update_isolation_rules --names rule_SOC_VDD_domain_to_Others\  
  -location to \  
  -cells HS65_LH_LSDOHLX18\
```

Since voltage islands, multiple voltage supplies, level shifters, isolation logic, and power switches are specified with respect to the power domain, not in RTL, changes are facilitated. Rather than being forced to modify RTL to insert isolation cells, NXP designers were able to use CPF as their golden power intent specification, permitting a generic and scalable methodology from synthesis through routing.

## Secondary Power Pin Connection

The Wasabe key infrastructure IP controls the memory access network. This IP block example is in one power domain with supply voltage WSB\_VDD\_D only; but it interfaces with several other voltage domains. So, the IP has level shifters on the receiving end of other power domains, as shown in Figure 124. In this case, the designer needed to avoid uncontrolled buffering of nets from input pins to the level shifter, and properly connect and route the extra power pins on the level shifter to power distribution.



**Figure 124. Wasabe IP block power domain interface**

Since chip level power domains, TM\_VDD\_D and SOC\_VDD\_D are made visible during the bottom-up block implementation, automation is improved and special handling of these cells removed. CPF provides the notion of virtual power domains (Figure 124) to which pins of the IP block are associated, thus providing the information about their power domains in the instantiation. Thus, level shifters can be implemented seamlessly regardless of the number of domains.

Sample CPF for the virtual power domains on Wasabe follows:



```

# WSB domain
create_power_domain -name WSB_VDD_domain -default \
    -boundary_ports ${e2_cl_wsb__WSB_domain__Output}
update_power_domain -name WSB_VDD_domain -internal_power_net WSB_VDD
# External domains to WSB
#Name: SMC SOC_VDD
create_power_domain -name SOC_VDD_domain \
    -shutoff_condition ip_pmc_1_vsoc_clamp_ena_n \
    -boundary_ports ${e2_cl_wsb__SOC_domain__Input}
# Name: TM TM_VDD
create_power_domain -name TM_VDD_domain \
    -shutoff_condition ip_pmc_1_vtmc_clamp_ena_n \
    -boundary_ports ${e2_cl_wsb__TM_domain__Input}
# Name: ALW_VDD_domain
create_power_domain -name ALW_VDD_domain \
    -boundary_ports ${e2_cl_wsb__ALW_domain__Input}
# Library information
define_isolation_cell -cells {HS65_LH_LSDOHLX18 HS65_LH_LSDOHHX18} -enable E -
valid_location to\
    -power vddout -power_switchable vddin -ground gnd
define_level_shifter_cell -cells {HS65_LH_LSDOHLX18 HS65_LH_LSDOHHX18} \
    -input_voltage_range 0.7:1.2:0.1 \
# Power Nets specified and connected
create_power_nets -nets WSB_VDD
create_global_connection -net VSS -domain WSB_VDD_domain -pins gnd
create_global_connection -net WSB_VDD -domain WSB_VDD_domain -pins vdd
create_global_connection -net WSB_VDD -domain WSB_VDD_domain -pins vddout
# Power Net connections and routing (outside of CPF1.0)
connect2ndPwr SOC_VDD -cells ${LS_Cell_List} -pin vddin -from SOC_VDD_domain -to
WSB_VDD_domain
connect2ndPwr TM_VDD -cells ${LS_Cell_List} -pin vddin -from TM_VDD_domain -to
WSB_VDD_domain
connect2ndPwr ALW_VDD -cells ${LS_Cell_List} -pin vddin -from ALW_VDD_domain -to
WSB_VDD_domain

```

## CPF Mitigates Complexity of Modes and Corners

Power reduction for the SoC depends on combining modes for which the voltage can be static or vary. This greatly increases the number of system level modes, and it is essential to be able to capture these modes and how the transitions between them are governed. More modes and more corners have a significant effect on complexity of verification. Static timing analysis complexity increases with more corners:

- Explosion of STA runs
- Blocks will have operating corners, constraints and libraries that will be combined to create many analysis and optimization views

Four different modes with associated nominal voltages for five domains are shown in Figure 125, below:

	WSB_VDD 0.9v	SOC_VDD 1.2v	TM_VDD 1.2v	DBG 1.2v	ALW 1.2v
all_on	on	on	on	on	on
dbg_off	on	on	on	off	on
tm_off	on	on	off	on	on
soc_off	on	off	on	on	on

**Figure 125. Power modes**

Here is an example of the CPF for several modes of operation for the power domains:

```
create_nominal_condition -name 1.2V -voltage 1.2
...
create_power_mode -name all_on -domain_conditions {WSB_VDD_domain@0.9V
SOC_VDD_domain@1.2V TM_VDD_domain@1.2V ARM_CORE_VDD_domain@1.2V
DBG_VDD_domain@1.2V ALW_VDD_domain@1.2V} -default
create_power_mode -name dbg_off -domain_conditions {WSB_VDD_domain@0.9V
SOC_VDD_domain@1.2V TM_VDD_domain@1.2V ARM_CORE_VDD_domain@1.2V
DBG_VDD_domain@off ALW_VDD_domain@1.2V}
```

With DVFS, any active block may imply a range of operating voltages and therefore a large number of corners. Performing timing optimization and sign-off verification on these corners can be overwhelming and may require many iterations.

Signals between voltage islands can also be challenging. For signoff, on a synchronous path, besides the presence of interface logic, the hold condition should theoretically be explored with the highest voltage on the driving domain and the lowest voltage on the receiving domain. Intermediate operating points will probably need to be verified as well.

In the example SoC, NXP reduced the potential timing issues on path spanning across power domains by making them asynchronous. ([Ref. 26](#))

This set of challenges needs the placeholder and abstraction for proper management provided by CPF. Raising the level of abstraction makes multimode, multi-corner analysis and optimization easier and less error-prone. Associating analysis views to each power mode gave NXP the ability to manage the different constraints and library associated to each operating condition of each power domain for each mode.

Here is the CPF for the analysis views for each power mode:

```
create_operating_corner -name WC_1v1_corner -  
library_set WC_1v1_lib -voltage 1.1 -temperature  
125 -process 1.5  
  
....  
create_analysis_view -name all_on_WC -mode  
all_on -domain_corners  
{WSB_VDD_domain@WC_1v3_corner \  
SOC_VDD_domain@WC_1v1_corner \  
TM_VDD_domain@WC_1v1_corner \  
DBG_VDD_domain@WC_1v1_corner \  
ALW_VDD_domain@WC_1v1_corner}
```

With a CPF centralized approach to single placeholder specification, power modes and operating conditions are concurrently taken into account during synthesis, optimization, STA, and formal verification in a multimode, multi-corner analysis and optimization flow.

The overhead associated with the complexity of designing with multiple supplies is greatly removed by the scalability of the CPF solution.

### ***DFT impact***

Insertion of scan chains across voltage islands can complicate implementation, and commercial tools are struggling to become multi-supply, multi-voltage aware. Some issues that arise from advanced low-power design in DFT include the following:

- Naturally, the test control block needs to be assigned a power domain
- Scan chains may span across power domains, and require level shifters
- CTAG may span voltage domain boundaries. Isolation should be placed within the voltage domain of the IO pin
- Scan chain routing within the boundaries of the voltage islands is preferred
- Over-random stitching of scan flip-flops across the voltage islands creates problems
- Testability of level shifters and switches remains a problem

However, if all power sequencing circuits will be held to a power-on state during test operation, the scan chain may not have to be designed based on voltage islands, a simplifying approach to multi-voltage test.

## CPF-Based Results

The SoC described earlier was designed by NXP as a CPF proof point project and successfully taped out in 2007. The notable results of this design project are as follows:

- Successful fabrication, in a 65nm process, of an aggressive design with 11 voltage islands, three voltage-scalable logic sections, three on-chip switchable domains, five off-chip switchable domains, separate switchable pad ring sections and three modules controlled using DVFS
- A 50% savings in implementing advanced power reduction techniques. In the past, before CPF, we had in some cases experienced 2X productivity drop in the implementation phase for such designs
- CPF power-aware simulation also discovered a critical issue: a time-out mechanism was being powered down in one particular mode, which could have caused deadlock conditions on the communication bus. With CPF we detected that the implementation of this timeout mechanism had been incorrectly placed in a domain that was subject to power-down, thereby disabling itself

This design demonstrated a scalable implementation of voltage islands. Tools understanding the same power design intent, with the highest possible level of abstraction, compensated for the throughput time overhead introduced by designing with multiple supplies.

Level shifters, retention logic and on-chip switches were logically inserted, verified, physically implemented and analyzed. Power modes and operating conditions were managed during synthesis, optimization, and STA, with multimode multi-corner analysis and optimization.

CPF provided the placeholder mechanism for power intent specification, avoiding error prone re-entry of the same power intent for each EDA tool in the flow, and supported better IP integration. NXP believes this methodology leads to significant time to market improvement.

Having proven the value of this standard, NXP will continue to drive for the additional functionality required for designing with the most advanced power management techniques, in all forums. NXP, with other LPC members, is currently exploring CPF features such as enhancements to hierarchical IP reuse, memory (and other custom IP) modeling, power network component modeling, associating clocks to power mode transitions, and support for power estimation. Another active LPC working group is currently developing a data model and API interface to support rapid incremental what-if scenarios. A top-level data model view is shown below:

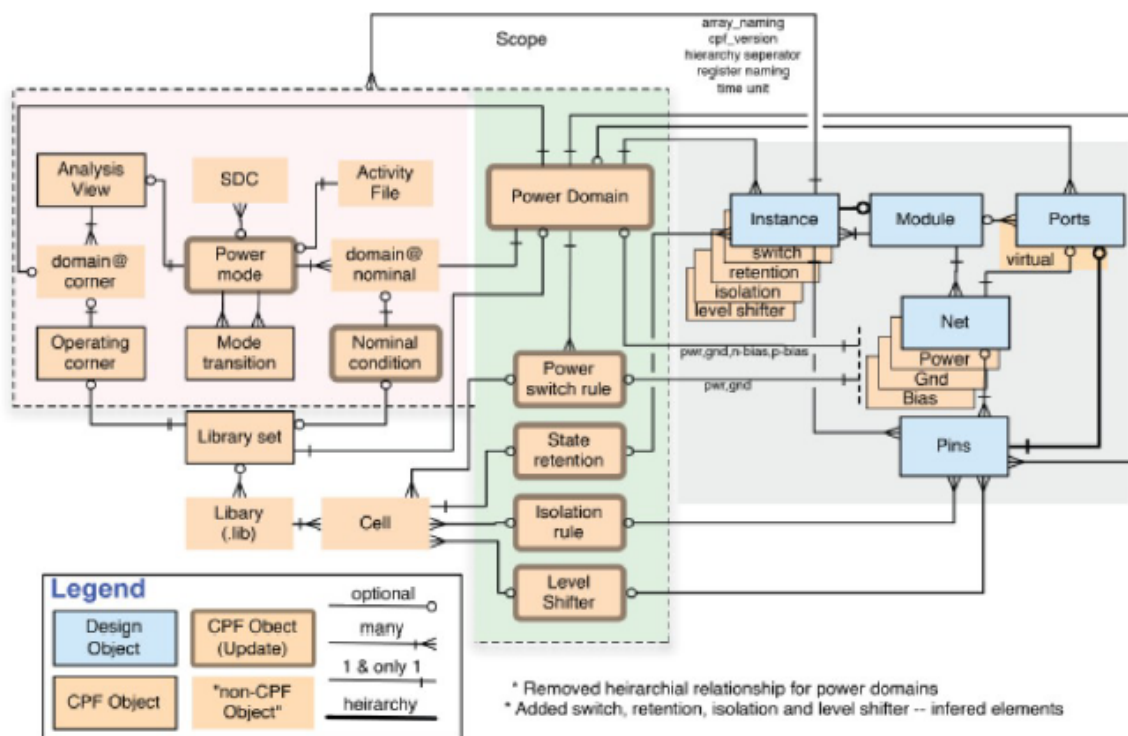


Figure 126. Top-level CPF data model view ([Ref. 22](#))

*Herve Menager is with Corporate Innovation & Technology at NXP Semiconductors. As Design Technology Architect, he is responsible for the SoC design environment and contributes to complex SoCs with a focus on advanced design techniques, including low power, for which he has several publications. He also participates in technical committees for various conferences as well as industry consortium such as Si2. Prior to Philips, he held a variety of positions in physical design ranging from engineering manager responsible for the development of floorplanning and routing technology at Compass Design Automation (VLSI Technology) to methodology engineering at Aristo Technology. Herve holds a MSEE and graduated from ENSEEIHT (Ecole Nationale superieure d'electronique, electrotechnique, informatique, hydraulique et des Telecommunications) National Polytechnic Institute of Toulouse.*

# References and Bibliography

## References

- [1] Fred Pollack of Intel. IEEE – 32<sup>nd</sup> Annual International Symposium on Microarchitecture, Haifa, Israel, 16-18 Nov. 1999. [www.huron.cs.ucdavis.edu/Micro32/homepage.html](http://www.huron.cs.ucdavis.edu/Micro32/homepage.html)
- [2] Steve Schulz. Si2, 2007. [www.si2.org](http://www.si2.org)
- [3] Taiwan Semiconductor Manufacturing Company. *Fine Grain MTCMOS Design Methodology*, TSMC Reference Flow Release 6.0, 2005.
- [4] Preeti Gupta. *Be Early with Power*. Chip Design Magazine. [www.chipdesignmag.com/display.php?articleId=613](http://www.chipdesignmag.com/display.php?articleId=613)
- [5] John Blyler. *Chip Design Trends Newsletter*, April 2007.
- [6] Andrew Piziali. *Verification Planning to Functional Closure of Processor-Based SoCs*. DesignCon, Feb. 2006. [www.designcon.com/2006/pdf/3-tp2\\_piziali.pdf](http://www.designcon.com/2006/pdf/3-tp2_piziali.pdf)
- [7] Leena Singh, Leonard Drucker, Neyaz Khan. *Advanced Verification Techniques: A SystemC Based Approach for Successful Tapeout*. Kluwer Academic Publishers, Norwell, MA, 2004.
- [8] Andrew Piziali. *Functional Verification Coverage Measurement and Analysis*. Kluwer Academic Publishers, Norwell, MA, 2004.
- [9] Holly Stump and George Harper. *ESL Synthesis + Power Analysis = Optimal Micro-Architecture*, Chip Design Magazine, Jan. 2007 [www.chipdesignmag.com/display.php?articleId=963&issueId=20](http://www.chipdesignmag.com/display.php?articleId=963&issueId=20)
- [10] Gagan Gupta. *ARC Energy PRO: Technology For Active Power Management*, CDNLive! 2007
- [11] Herve Menager. *Words of Power: Reusable, Holistic, Scalable Multi-voltage Design*. EPD Conference, 12 April 2007. NXP. [www.eda.org/edps/edp07/ApprovedPapers/01%20Herve%20Menager.pdf](http://www.eda.org/edps/edp07/ApprovedPapers/01%20Herve%20Menager.pdf)
- [12] NEC Electronics Environmental Management Report 2007. Eco-product concept UltimateLowPower, page 15. [www.necel.com/eco/en/report07/2007\\_en.pdf](http://www.necel.com/eco/en/report07/2007_en.pdf)
- [13] Integrating Power Awareness into IC Design By Toshiyuki Saito, NEC Electronics Corporation EDA DesignLine 03/01/2007. [www.edadesignline.com/howto/showArticle.jhtml?articleID=197700296](http://www.edadesignline.com/howto/showArticle.jhtml?articleID=197700296)
- [14] Electronic Design, Ron Schneiderman, January 31, 2008 [www.electronicdesign.com/Articles/ArticleID/18111/18111.html#](http://www.electronicdesign.com/Articles/ArticleID/18111/18111.html#)
- [15] NEC TECHNICAL JOURNAL Vol.2 No.4 (December, 2007), pp20-24, [www.nec.co.jp/techrep/en/journal/g07/n04/070406.html](http://www.nec.co.jp/techrep/en/journal/g07/n04/070406.html)
- [16] ASP-DAC 2008 Technical Program, S.Kunie et.al., 8D-4, pp. 748-753.
- [17] Fujitsu Annual Report 2007. [www.fujitsu.com/global/about/ir/library/annualrep/2007/](http://www.fujitsu.com/global/about/ir/library/annualrep/2007/)
- [18] STARC Releases 'PRIDE' Reference Design Flow Using Cadence Low-Power and DFM Solutions [www.cadence.com/company/newsroom/press\\_releases/pr.aspx?xml=012108\\_starc](http://www.cadence.com/company/newsroom/press_releases/pr.aspx?xml=012108_starc)
- [19] CPF Description Guidelines for ASIC / ASSP customers. Available from Fujitsu.
- [20] CPF handoff guidelines to help accelerate the handoff between Fujitsu and its ASIC / ASSP design customers. Available from Fujitsu.
- [21] H. Menager, Words of Power: Reusable, Holistic, Scalable Multi-voltage Design. EPD2007 Workshop, April 12th 2007.
- [22] Steve Schulz, A Practical Case Study in Low Power Design Methodology, EPN Online. [www.epn-online.com/page/new56459/a-practical-case-study-in-low-power-design-methodology.html](http://www.epn-online.com/page/new56459/a-practical-case-study-in-low-power-design-methodology.html)
- [23] H. Menager, Experience implementing a complex SoC, leveraging a reusable low power specification, CDN Live 2007.
- [24] H. Menager, Low Power specification and scalable approach to designing a complex SoC, Si2 LPC DAC 2007 workshop, June 3rd 2007.



- [25] H. Menager, M. Korenhof, M. Huiskes, NXP Semiconductors. Improving Design Turn Around Time On A Complex SoC By Leveraging A Reusable Low Power Specification, Presented at IP07 on October 16th, 2007. [www.us.design-reuse.com/articles/17703/soc-reusable-low-power.htm](http://www.us.design-reuse.com/articles/17703/soc-reusable-low-power.htm)
- [26] N. Wingen “What if you could design tomorrow's system today?” Design, Automation, and Test in Europe, pp. 835 – 840.
- [27] A P Niranjana and P Wiscombe, “Island of Synchronicity, a design methodology for SoC Design” Design Automation and Test in Europe, Feb.2004 pp.488-491.

## Additional Low-Power References of Interest

- [1] David Lammers. *Leakage current needs multipronged attack*. In *CMP – Power Management Design Line* (5 Dec. 2005) [www.eetimes.com/news/latest/showArticle.jhtml?articleID=174900266](http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=174900266)
- [2] Jack Horgan. *Low Power SoC Design*. EDA Cafe (17-21 May 2004) [www10.edacafe.com/nbc/articles/view\\_weekly.php?articleid=209217](http://www10.edacafe.com/nbc/articles/view_weekly.php?articleid=209217)
- [3] Robert Aitken, George Kuo, and Ed Wan. *Low-power flow enables multi-supply voltage ICs*. EE Times (21 March 2005). [www.eetimes.com/showArticle.jhtml?articleID=159902216](http://www.eetimes.com/showArticle.jhtml?articleID=159902216)
- [4] Mohit Bhatnagar, Jack Erickson, Anand Iyer, Pete McCrorie. *Save Those Watts with A Power-Aware Design Flow for SoCs*. Electronic Design (6 July 2006). [www.elecdesign.com/Articles/Index.cfm?ArticleID=12946](http://www.elecdesign.com/Articles/Index.cfm?ArticleID=12946)
- [5] PFI Core Comp Team. *CPF AE LP Workshop Training Material*. Cadence Internal Training 2006-2007
- [6] Milind Padhye, Noah Bamford, Saji George. *Wireless Low Power Design and Verification Challenges*. EPD Conference, 12 April 2007. Freescale Semiconductor Inc. [www.eda.org/edps/edp07/ApprovedPapers/04%20Milind%20Padhye.pdf](http://www.eda.org/edps/edp07/ApprovedPapers/04%20Milind%20Padhye.pdf)
- [7] Krisztián Flautner. *Blurring the Layers of Abstractions: Time to take a step back?* EPD Conference, 12 April 2007. ARM Limited. [www.eda.org/edps/edp07/ApprovedPapers/03%20Kris%20Flauter.pdf](http://www.eda.org/edps/edp07/ApprovedPapers/03%20Kris%20Flauter.pdf)
- [8] David Chinnery, Kurt Keutzer. *Closing the Power Gap between ASIC & Custom: Tools and Techniques for Low Power Design*, Springer, 2007.
- [28] Milind Padhye, *Wireless Low Power Design and Verification Challenges*, IEEE, EDP Conference, April 12, 2007. [www.eda.org/edps/edp07/ApprovedPapers/04%20Milind%20Padhye.pdf](http://www.eda.org/edps/edp07/ApprovedPapers/04%20Milind%20Padhye.pdf)
- [29] Hailin Jiang, Marek-Sadowska, M., Nassif, S.R., ECE Dept., UCSB, Santa Barbara, CA, USA Benefits and costs of power-gating technique. Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference.
- [30] Kang, S.M.S. California Univ., Santa Cruz, CA, USA Elements of low power design for integrated systems, In Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium.
- [31] Milind Padhye, Low Power Design Challenge in Wireless with Deep Submicron Geometries, IEEE.
- [32] Milind Padhye, CPFIed Low Power Design and Verification [www.cdnusers.org/interviewcpfi\\_milindpadhye\\_freescale/tabid/421/Default.aspx](http://www.cdnusers.org/interviewcpfi_milindpadhye_freescale/tabid/421/Default.aspx)
- [33] LC Lu, TSMC and George Kuo, Cadence Design Systems, Ensuring Power Designing Works at 65nm [EDA DesignLine](http://www.edadesignline.com/howto/202600354;jsessionid=XEKEBPHVOAHJ0QSNL0SKH0CJUNN2JVN?pgno=2), 10/22/2007 [www.edadesignline.com/howto/202600354;jsessionid=XEKEBPHVOAHJ0QSNL0SKH0CJUNN2JVN?pgno=2](http://www.edadesignline.com/howto/202600354;jsessionid=XEKEBPHVOAHJ0QSNL0SKH0CJUNN2JVN?pgno=2)

## Low-Power Links

[Silicon Integration Initiative \(Si2\) and Low Power Coalition \(LPC\)](#)



[Si2 Low Power Coalition CPF Specification](#)

[Si2 CPF 1.0 Quick Reference Programmer's Guide](#)

## Power Forward Initiative

Participants



## Cadence Low-Power Links

[www.cadence.com/lowpower](http://www.cadence.com/lowpower): Technologies, news, white papers, success stories, webinar

[www.Cdnusers.org](http://www.Cdnusers.org): Low-Power Community



## CPF Terminology Glossary

The following glossary terms are directly from the [Si2 CPF 1.0 specification](#).

### Design Objects

Design objects are objects that are being named in the description of the design, which can be in the form of RTL files or a netlist. Design objects can be referenced by the CPF commands.

Design Object	Definition
<i>Design</i>	<b>The top-level module.</b>
<i>Instance</i>	An instantiation of a module or library cell. <ul style="list-style-type: none"><li>▪ Hierarchical instances are instantiations of modules.</li><li>▪ Leaf instances are instantiations of library cells.</li></ul>
<i>Module</i>	<b>A logic block in the design.</b>
<i>Net</i>	A connection between instance pins and ports.
<i>Pad</i>	An instance of an I/O cell.
<i>Pin</i>	An entry point to or exit point from an instance or library cell.
<i>Port</i>	An entry point to or exit point from the design or a module.

### CPF Objects

CPF objects are objects that are being defined (named) in the CPF constraint file. CPF objects can be referenced by the CPF commands.

CPF Object	Definition
<i>Analysis View</i>	A view that associates an operating corner with a power mode for which SDC constraints were specified. The set of active views represent the different design variations (MMMC, that is, multi-mode multi-corner) that will be timed and optimized.
<i>Isolation Rule</i>	Defines the location and type of isolation logic to be added and the condition for when to enable the logic.
<i>Level Shifter Rule</i>	Defines the location and type of level shifter logic to be added.
<i>Library Set</i>	A set (collection) of libraries that was characterized for the same set of operating conditions. By giving the set a name it is easy to reference the set when defining operating corners.
<i>Nominal Operating Condition</i>	A typical operating condition under which the design or blocks perform.
<i>Mode Transition</i>	Defines when the design transitions between the specified power modes.

<i>Operating Corner</i>	A specific set of process, voltage, and temperature values under which the design must be able to perform.
<i>Power Domain</i>	<p>A collection of instances that use the same power supply during normal operation and that can be switched on or off at the same time. You can also associate boundary ports with a power domain to indicate that the drivers for these ports belong to the same power domain. The only leaf instances allowed are IP blocks and I/O pads. A power domain can be nested within another power domain.</p> <p>At the physical level a power domain contains:</p> <ul style="list-style-type: none"> <li>▪ _ A set of (regular) physical gates with a single power and a single ground rail connecting to the same pair of power and ground nets</li> <li>▪ _ The nets driven by these physical gates</li> <li>▪ _ A set of special gates such as level shifter cells, state retention cells, isolation cells, power switches, always-on cells, or multi-rail hard macros (such as, I/Os, memories, and so on) with multiple power and ground rails. At least one pair of the power or ground rails in these special gates or macros must be connecting to the same pair of power and ground nets as the (regular) physical gates connect to.</li> </ul> <p>At the logic level a power domain contains:</p> <ul style="list-style-type: none"> <li>▪ _ A set of logic gates that correspond to the (regular) physical gates of this power domain</li> <li>▪ _ The nets driven by these logic gates</li> <li>▪ _ A set of special gates such as level shifter cells, state retention cells, isolation cells, power switches, always-on cells, or multi-rail hard macros (such as, I/Os, memories, and so on) that correspond to the physical implementation of these gates in this power domain.</li> </ul> <p>At RTL a power domain contains:</p> <ul style="list-style-type: none"> <li>▪ _ The computational elements (operators, process, function and conditional statements) that correspond to the logic gates in this power domain</li> <li>▪ _ The signals that correspond to the nets driven by the corresponding logic gates.</li> </ul>
<i>Power Mode</i>	A static state of a design in which each power domain operates on a specific nominal condition.
<i>Power Switch Rule</i>	Defines the location and type of power switches to be added and the condition for when to enable the power switch.
<i>State Retention Rule</i>	Defines the instances to be replaced with state retention flip-flops and the conditions for when to save and restore their states.

## Special Library Cells for Power Management

Library Cell	Definition
<i>Always-on Cell</i>	A special cell located in a switched-off domain, and whose power supply is continuous on even when the power supply for the rest of the logic in the power domain is off.
<i>Isolation Cell</i>	Logic used to isolate signals between two power domains where one is switched on and one is switched off. The most common usage of such cell is to isolate signals originating in a power domain that is being switched off, from the power domain that receives these signals and that remains switched on.
<i>Level Shifter Cell</i>	Logic to pass data signals between power domains operating at different voltages.
<i>Power Clamp Cell</i>	A special diode cell to clamp a signal to a particular voltage.
<i>Power Switch Cell</i>	Logic used to connect and disconnect the power supply from the gates in a power domain.
<i>State Retention Cell</i>	Special flop or latch used to retain the state of the cell when its main power supply is shut off.

## Index

- Adaptive voltage scaling, 129, 133, 141
- Body bias, 12, 17
- Clock distribution, 11, 12, 54, 76
- Clock gating, 13, 54, 56, 57, 58, 77, 81, 129, 133, 141
- Clock tree, 11, 12, 54, 76
- Common Power Format, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 41, 42, 43, 52, 53, 64, 66, 67, 71, 76, 164, 165
- Common Power Format (CPF), 2, 3, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 36, 38, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 52, 53, 54, 55, 56, 62, 63, 64, 66, 67, 68, 69, 71, 75, 76, 78, 82, 83, 85, 86, 87, 89, 93, 98, 99, 101, 102, 105, 106, 107, 108, 109, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 128, 129, 130, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165
- Conformal, 27, 28, 32, 121, 132, 136, 138, 168
- CPF Object, 165
  - Analysis View, 165
  - Isolation Rule, 165
  - Level Shifter Rule, 165
  - Library Set, 165
  - Mode Transition, 165
  - Nominal Operating Condition, 165
  - Operating Corner, 166
  - Power Domain, 20, 82, 86, 166
  - Power Mode, 151, 166
  - Power Switch Rule, 166
  - State Retention Rule, 166
- Design for test (DFT), 29, 55, 58, 59, 71, 147, 148, 152, 159
- Design Object, 32, 33, 165
  - Instance, 165
  - Module, 165
  - Net, 143, 165
  - Pad, 165
  - Pin, 12, 54, 61, 155, 165
  - Port, 165
- DVFS, 55
- Dynamic power, 9, 10, 54, 74, 129
- Dynamic voltage and frequency scaling, 12
- Dynamic voltage and frequency scaling (DVFS), 12, 13, 14, 27, 45, 54, 55, 56, 62, 65, 66, 68, 69, 77, 93, 101, 103, 104, 105, 113, 129, 133, 141, 146, 148, 149, 158, 160
- Dynamic voltage scaling, 12, 14, 103
- Energy, 2, 8, 31, 54, 99, 100, 101, 106, 131, 162
- Front End Design, 52, 135
- Incisive Unified Simulator (IUS), 23, 24, 25, 121, 132, 152, 153
- IR drop, 14, 76, 79, 87, 89, 138
- Leakage, 10, 11, 54, 110, 111, 145, 163
- Level shifter, 26, 44, 63, 65, 82, 83, 84, 134, 148, 154, 160
- Library Cells, 166
  - Always On Cell, 166
  - Isolation Cell, 89, 166
  - Level Shifter Cell, 166
  - Power Clamp Cell, 166
  - Power Switch Cell, 86, 166
  - State Retention Cell, 166
- Logic Restructuring, 60
- Low Power Coalition, 7, 164
- Low Power Coalition (LPC), 3, 7, 129, 146, 160, 162, 164
- MTCMOS, 162
- Multi-Supply Voltage (MSV), 12, 13, 14, 27, 50, 55, 56, 62, 63, 66, 76, 77, 79, 82, 83, 84, 93, 116, 118, 129, 133, 141, 146, 147, 148, 149, 151
- Multi-voltage, 162
- Multi-Vt, 12, 13, 54, 62, 76, 77, 81, 129, 133, 141
- Operand Isolation, 11, 59, 60
- Physical Implementation, 30, 31, 76, 86
- Power density, 6
- Power domain, 21, 30, 31, 37, 41, 86, 119, 150
- Power gating, 12, 14, 49
- Power Logic, 154

Power shutoff (PSO), 12, 14, 24, 55  
 Power shut-off (PSO), 12, 14, 21, 24, 29,  
     37, 41, 49, 50, 54, 55, 56, 62, 69, 83,  
     85, 86, 87, 88, 89, 90, 91, 92, 97, 102,  
     103, 104, 105, 113, 118, 129, 133,  
     134, 141, 148  
 Resizing, 61  
 Retention, 15, 71, 90, 118, 151, 166  
 RTL Compiler, 26, 57, 73, 132  
 Si2, 3, 7, 18, 98, 129, 146, 148, 161,  
     162, 164, 165  
 SoC Encounter, 30, 116, 132, 136, 138,  
     139, 168  
 State retention, 15, 44, 45, 55, 90, 91,  
     148  
 State Retentive Power Gating (SRPG),  
     14, 15, 49, 71, 76, 90, 91  
 Synthesis, 26, 27, 54, 56, 57, 62, 63, 65,  
     66, 69, 73, 74, 75, 80, 81, 82, 85, 137,  
     162  
 Test, 29, 30, 58, 59, 118, 119, 130, 133,  
     163  
     Scan, 159  
 Transition rate buffering, 12, 54, 61  
 Verification, 23, 25, 32, 34, 35, 38, 39,  
     40, 50, 54, 102, 133, 138, 148, 151,  
     162, 163  
 Voltage island, 12, 148, 154  
 Voltage scaling, 65



Cadence Design Systems, Inc.

[www.cadence.com](http://www.cadence.com)

© 2008 Cadence Design Systems, Inc. All rights reserved.  
 Cadence, the Cadence logo, Conformal, Encounter, Incisive, and Verilog are either  
 trademarks or registered trademarks of Cadence Design Systems, Inc.