

## Power-Optimization Solution Serves Ubiquitous RTL Designer

Ask any designer to identify the most common design capture mechanism for today's digital designs and the answer is the register transfer level (RTL). Almost all RTL designs pass through a synthesis tool before place and route for final implementation. Depending on the application, the designer can choose to use a high-level synthesis tool to generate the RTL or bypass synthesis and go directly to custom layout. If the application is a filter for digital signal processing (DSP) that is datapath-intensive, it is quite possible to use a high-level synthesis tool to generate the RTL.



By Kiran Vittal

When it comes to power optimization, power-synthesis tools insert clock gating to reduce power. Layout tools use different voltage-threshold libraries or implement an optimized clock tree layout for low power. Better results can typically be achieved at higher levels of abstraction, such as the architecture planning stage, by doing what-if analysis of the different power and voltage domains or modeling use cases for the design. However, the low-hanging fruit for the ubiquitous RTL designer is to leverage techniques available with RTL power estimation and reduction tools to implement and optimize the enables for clock gating of registers or memory enable and read/write operation.

Today's power-smart RTL designers use an "explicit enable signal" coding style in their VHDL or Verilog designs. This approach allows power-synthesis tools to insert clock-gating logic on these explicit enables. However, such tools do not do the computation of the differential power savings for each enable. And in some cases, the clock gate could cause negative power savings.

In addition, many design teams use legacy IPs or create new RTL with no "explicit enables." These legacy designs or the newly written RTL could benefit from power-reduction tools with sequential formal technology to find

## Take the High Road to Power-Optimized RTL



By Shawn McCloud

The best and surest way to power-optimized register transfer level (RTL) is the "high road," beginning with C synthesis and the unique capabilities that it gives to architectural exploration and low-power optimizations. Starting with the highly abstract ANSI C++ source imparts the digital design traveler with a complete map of the design terrain. Through an interactive synthesis process, one can automatically evaluate different candidate designs for area, performance, and power consumption and make important design decisions at a highly abstract level.

The first generation of commercially available high-level-synthesis (HLS) tools, appearing in the 1990s and early 2000s, focused on signal-processing implementations. Since then, their capabilities have been greatly expanded to target both the dataflow and control domains in the most complex, multimillion-gate application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs). These HLS tools also include the advanced power-optimization capabilities required in today's electronic market.

Reducing both dynamic and static power consumption isn't just paramount to the success of mobile and handheld devices. It also is critical for a growing list of products that seek to reduce cooling, packaging, and power costs. Yet low-power design techniques are largely manual, time-consuming, and error prone at the RTL. Even if the skill set and expertise are available, there rarely is time to fully apply it—let alone find the best balance between power, area, and performance. Fortunately, there's a way out. C synthesis automates proven power-optimization techniques during the high-level synthesis process. It also produces multiple design solutions that are easy to analyze for power consumption while significantly reducing RTL verification.

There's no need to wait for RTL power-assessment tools to begin optimizing designs. After all, today's most advanced C synthesis tools automatically apply common techniques to save power. Such techniques include resource sharing, memory-

new “implicit enable” opportunities with positive power savings. The tool should indicate to the designer where to make changes to the RTL or automatically write out a modified RTL with new enables.

The earliest that accurate results for power reduction can be obtained is when the technology library has been chosen and simulation testbenches are available. Only with both technology and simulation data can the actual power savings be computed. Before either is available, however, it may be possible to get useful results by simply locating possible power-reduction opportunities and providing a “scorecard” on how much clock gating is done in the various modules of the design.

For example, Atrenta’s SpyGlass®-Power solution starts with the RTL description of the design, performs a fast synthesis, and analyzes the result to suggest clocks that could be gated to achieve power efficiencies. Rather than allow a synthesis tool to insert gated clocks based only on the width of the data bus, SpyGlass-Power shows the designer which enables will save the most power with a built-in power-estimation engine. It also creates a constraint file for downstream synthesis. Beyond that, it will report new opportunities for implicit clock enables that may not have occurred to the RTL designer by using sequential formal technology.

For example, ungated downstream registers present an opportunity for power savings if the enable were delayed by one or a few clock cycles. The same could apply to registers upstream. SpyGlass-Power does not just find the new “implicit enables” in the design. It also can give the user the flexibility to selectively auto-fix the RTL and write out the modified design with the new enables, which can be verified with a built-in sequential power equivalence checking (SPEC) tool. Users have claimed up to 40% power reduction at RTL.

The product suite also provides other early analysis tools—mainly at the RTL—with the goal of uncovering clock domain crossing (CDC) bugs or testability issues. The SpyGlass-Power solution is CDC-aware so as to not implement new clock gating on domain crossings that can cause functional bugs.

If the application calls for high-level synthesis, such tools can also add power-saving techniques like clock

access optimizations, multiple clock-domain management, dynamic voltage and frequency scaling, and clock gating.

Automating the prevailing low-power techniques at a higher abstraction is essential, as it impacts all design activities. It also is the surest way to circumvent the lack of time and expertise found in manual RTL flows. By automating these proven techniques at a high level of abstraction, C synthesis presents vast potential to optimize for power in a manner that cannot be done otherwise. This is because of the insight and control C synthesis gives over so many aspects of the design. In other words, as algorithmic synthesis builds the RTL description from C code, it automatically inserts all of the technological intent, such as timing, clocks, registers, datapaths, IO access, and FSM. In doing so, it offers a unique ability to optimize across a wide range of hardware corners.

Currently, for example, clock gating is typically done with a bulldozer. The vast majority of designers simply use a global enable to fully suspend or enable a design because of the work it takes to achieve clock gating at a register level of granularity. Designers with enough time and skills might be able to insert clock gating on important registers within the design. However, it’s very difficult if not impossible to make an inclusive set. As a result, a lot of potential power savings is left on the table. To recover this potential, high-level synthesis tools like Catapult® C identify all clock-gating candidates in the design. This aspect ensures that every register that can be gated will be gated. The resulting design is far more optimized than what can be done by hand. On average, users have seen a 40% reduction in power usage from multi-level clock gating alone. One customer even saw savings of 95%.

The high-level-synthesis road is clearly the best way to a low-power RTL destination. C synthesis delivers power-optimized RTL that is free of the errors that are inevitably introduced by manual HDL coding. In addition, it automatically creates multiple RTL netlists, which are unthinkable when designing by hand. As a result, the designer can analyze a variety of scenarios and then choose the best balance of area, performance, and power for his or her application. Working at a higher level of abstraction increases the designer’s influence over these three design characteristics, leading to greater power savings and a better mix than is practical through manual RTL coding.

When the designer arrives with this set of low-power design solutions in hand, he or she will want to use a power-analysis tool that makes it easier to accurately measure the power consumption of each in order to pick the best fit. When

gating. The true effectiveness of these techniques requires more detail—detail that is typically only available in the later stages of RTL. By using formal techniques to look backwards and forwards across multiple cycles, it is possible to find new enables that were not found by high-level synthesis tools. By using a tool like SpyGlass-Power in conjunction with high-level synthesis tools like Catapult® C, optimal results can be achieved. High-level synthesis enables can be validated by SpyGlass and additional power-saving opportunities are typically found as well. This hybrid strategy is the best approach to early power optimization.

*Kiran Vittal is a product marketing director at Atrenta with 19 years of experience in EDA and semiconductor design. Prior to joining Atrenta, he held engineering, field-application, and product-marketing positions at Synopsys Inc., ViewLogic Inc., and Mentor Graphics Inc. Vittal holds an MBA from Santa Clara University and a bachelor's degree in electronics engineering from India. He can be reached at kvittal@atrenta.com.*

combined with high-level synthesis tools, power-analysis tools become even more useful. They can help the designer analyze and compare several netlists instead of only the one that is possible with hand-coded RTL. Also, only a fraction of time is spent verifying the chosen RTL design, as the C++ code has already been exhaustively verified.

C synthesis now commands a very wide scope and applicability. It can be used to design multimillion-gate subsystems comprised of algorithm and control logic including interfaces, finite state machines, and datapaths found in the most complicated designs. It's important to do power management as these complex systems are synthesized. So it makes sense to continue the journey at the confluence of these two powerful power-optimization technologies.

*Shawn McCloud is the product line director for Mentor Graphics high-level synthesis technology. He joined Mentor Graphics in 1994 after several years as a senior system architect responsible for RISC- and CISC-based microprocessor design. McCloud received his BS in electrical and computer engineering from Case Western Reserve University.*

## Far more than just Standards....



- **90+ Si2 Corporate Members**
- **All coalitions led by companies who design ICs**
- **Board of Directors: AMD, ARM, Cadence, IBM, Intel, LSI, National Semiconductor, NXP, Sequence Design, Synopsys**

**Join the leaders in the semi and EDA industry!**



**.....It's about Adoption and ROI**

**Featured Downloads**

At: [www.si2.org/?page=1038](http://www.si2.org/?page=1038)

[WWW.si2.org](http://WWW.si2.org)

