By Tom Daspit

The Ever-Elusive Design Closure

Over the last decade, a growing impatience with older technologies has driven electronics companies to bring new or derivative products to market every four to six months. Now more than ever, time-to-market has a very serious meaning.

But companies like Sun report that sometimes design closure takes anywhere from 130 to 190 percent longer than planned—in market terms, this means product, or even company, death.

Is there any relief in sight? The answer will not come from one great innovation. Instead, a combination of things must occur simultaneously to relieve the hardships encountered on the quest for design closure. Much emphasis has been placed on timing-closure solutions at the back end of the design cycle. If you look at the design flow itself, you can see that RTL engineers also go through multiple loops during the design process. The first loop is during the development of the RTL, as designers write, simulate and then synthesize the RTL code. Then they check the performance and most often go back to modify the RTL and synthesize it again. If the performance goals are not met, they change the constraints and synthesize once more.

Design closure, in reality, consists of getting the RTL code correct before synthesis and getting the performance after layout to meet the post-synthesis goals.

Once the performance goals are met with synthesis, the design moves into the layout stages. After layout, post-layout performance is compared with post-synthesis performance and, typically, the numbers don't match. Designers then usually perform physical optimization, but often this buys them only about a 10 percent improvement in their overall timing goals. If the goals are still not met, then one of two things can happen. Either the designers sacrifice performance to get the product out to market or they go back to restructure the RTL, which can kill the design schedule because of reverification and implementation time.

What engineers need today is a mechanism to improve the RTL-coding process. Most of today's designers don't understand how their RTL code ends up being converted to gates through synthesis. Up front they don't understand the performance they can obtain from a given technology based on their design objectives. Yet, both of these are necessary for an RTL designer to get the code right the first time and to minimize the number of synthesis-to-RTL loops they get stuck in.

Today, design closure is equated with timing closure, which focuses on reducing synthesis-to-layout iterations, not the RTL-to-synthesis iterations. If the RTL is written incorrectly, no amount of effort during the physical-optimization stage will result in design closure.

Many have tried to pin the blame on synthesis. Synthesis is just the implementation tool; it cannot fix a bad design that arises from an architectural problem in the RTL code. What counts is the stuff that happens before and after the synthesis process.

Design closure, in reality, consists of two parts:

1. Getting the RTL code correct before synthesis, and
2. Getting the performance after layout to meet the post-synthesis goals.

The first step, getting the code correct before synthesis, is what we call RTL closure. RTL closure allows designers to exit the RTL-coding stage with architecturally correct RTL code. This greatly reduces the requirement for the designers to loop between synthesis and RTL.

RTL closure is achieved based on the following:

- Knowledge of how the code is converted to vendor-specific gates,
- Accurate timing prediction before synthesis,
- Virtual prototyping to constrain and drive synthesis, and
- Driving implementation to meet the design goals.

Ultimately, then, design closure is really RTL closure combined with timing closure. It really cannot occur without the other two. ■

Tom Daspit is vice president of product marketing for InTime Software, Cupertino, Calif.