



The Next Roadblock to Custom Design Productivity: Design Constraints **Mark Waller, VP of Research and Development, Pulsic**

Abstract

Design constraints, which express design intent, are one of the pieces of ancillary data that are critical to the success or failure of a custom design. Design constraints aren't usually contained within layout files or library information, but without these critical data, designs may not meet specifications. Today, most custom design teams manage constraints in ad-hoc, manual fashions. This ad-hoc approach has become a significant limitation when it comes to automating the custom design process, which in turn can limit both design productivity and accuracy. Moving forward, the custom design community is starting to look to standards efforts to ease the burden of correlating and communicating design constraints throughout the design process.

Design Constraints Are Critical to Custom-Design Success

Design constraints are data that express design intent, both on the macro and micro levels. For example, a design constraint may specify the target frequency for the entire circuit, or just the current load required for a particular net. Design constraints for custom designs originate with the circuit designer(s) during the layout or schematic-capture phase of the design process. The designer notes necessary design and operating conditions for the circuit or portions of the circuit, placing "constraints" on how the design is handled during subsequent design phases so that these necessary conditions are maintained through every phase of the design process. Without design constraints, the design's function or performance might be negatively impacted by alterations during the design process.

Design constraints ensure that design intent is considered at every phase of the design process. While most design constraints originate in the layout schematic-capture design phase, they may be added or modified throughout the design process in response to engineering change orders (ECOs). Design constraints communicate these changes up and down the design flow, creating a consistent understanding of necessary specifications.

Design Constraints vs. Technology Constraints

Design constraints are not to be confused with *technology constraints*, which are specified by the foundry and define the limits of the process technology, such as minimum feature size, minimum routing width, etc. There may be some overlap between technology and design constraints. For example, both may have a specification for minimum routing width. In this case, the more conservative of the two specifications would take precedence.

One important difference between technology and design constraints is that most technology constraints (from whatever foundry) can be represented in a standard fashion in most design flows today, thanks to the efforts of such groups as OpenAccess, Si2, and the IPL Alliance, et al. Today, there is no standard method of representing design constraints. Most design teams use ad-hoc, manual methods for conveying design constraints through the design process.

Design Constraints Today: Spreadsheets and Post-Its®

As critical as design constraints are to successful custom design, the most common methods of documenting and tracking design constraints today are entirely manual, and employ spreadsheets or even hand-written paper notations. These manual notations are created by the circuit designer and then must be passed along to the members of the design team managing each phase of design. Changes in constraints due to ECOs likewise are communicated manually, usually via an email message, with an updated spreadsheet attached in the best-case scenario (or instructions to update hand-written notes in the worst-case).

Some larger design teams may automate the documentation and implementation of design constraints using custom scripts, but even this process is very laborious and error-prone as these scripts still must be manually updated every time the constraints change.

Some custom design automation tools offer formats for entering design constraints. However, these formats are proprietary and apply only to the design phase the particular tool handles. Today, no common format exists for communicating design constraints through the design process or across design-tool platforms.

Ad-Hoc Methods Limit Accuracy and Productivity

The ad-hoc methods used to manage design constraints for custom designs result in losses of both accuracy and productivity. Manual methods for documenting, tracking, and implementing constraints are vulnerable to errors not only in transcription and communication, but also in execution and verification. Obviously, productivity is limited when each design team member must receive a communication every time a constraint is specified or updated and then must manually note or update the constraint information as it applies to their phase of the design process.

Even more harmful is the fact that the ad-hoc nature of the handling of design constraints in today's design process can be a barrier to the adoption of automation tools for custom design. The effort required to translate and manage design constraints among varied design automation platforms have kept some design teams using manual design methodologies, despite the productivity gains offered by design automation.

At today's advanced process geometries, however, custom designs have become so large and complex that automation is a necessity in terms of both productivity and accuracy.

Example: Parasitic Constraints and Routing

Parasitic constraints and their impact on routing are a prime illustration of the need for automation in the custom design process. Prior to the adoption of deep-submicron process technologies, custom designers could focus their efforts on placement alone to account for the vast majority of the unintended, or parasitic, electrical effects generated by their circuit layouts. Most parasitic constraints and signal integrity problems could be addressed through placement, since the majority of these parasitic effects (most especially capacitance and resistance) were generated by the components themselves. Routing the wires connecting the placed components was simply a matter of following design-rule checks (DRCs) and minimizing wire length.

However, for very deep-submicron processes (45nm and below), most of the parasitics are generated by the taller, thinner, more closely packed wires themselves, not by the components. The number and complexity of the interactive variables that must be considered during routing not only have rendered hand-routing impossible, but created a need to use parasitic constraints to drive the router required to automate the process.

These parasitic constraints do not apply to the routing process only, however. Circuit performance is inextricably connected to parasitic constraints, both in terms of circuit speed and function. Therefore performance constraints and parasitic constraints must be defined and used throughout the design process, and then used by all the automation tools employed by the design team.

The Future of Constraints

The custom design community has recognized that the lack of a standard way to communicate design constraints is a significant barrier to the adoption of automation technologies that can both improve productivity and decrease error. Multiple efforts are underway to address this void, such as the work by several leading companies with the IPL Alliance to develop a standard format for design constraints.

One company contributing to the IPL Alliance effort is Pulsic, a leader in custom-design automation. Pulsic has been working with other leading companies to address both the particular need to define parasitic constraints discussed in the previous section, and the general need for standardized constraints throughout the custom design process. Over the course of a decade working with custom design teams, Pulsic has helped its customers tape out over 200 designs, reducing their time-to-market by up to 50% through automation of their custom routing. Standardized constraints would not only ease the automation path for current users, but also enable more custom design teams to reap the benefits that automation offers. As custom design automation extends to the hierarchical floorplanning and placement phases, the need for standardized constraints only increases.

Pulsic has been able to lend its many years of experience working with customers in the memory, custom digital and analog design communities to the constraints standardization

effort. Pulsic is reaching out to these communities to include them in the process by creating a feedback channel on its website at www.pulsic.com.

While many design teams will continue to have some few “customized” design constraints particular to their designs, applications, or design flow, there are many constraints that are universal (or nearly so) that will be critical to include in a useful standard. Among these are bus constraints, group constraints, net constraints, signal integrity constraints, spacing constraints, track control constraints and via control constraints.

A standard format for design constraints that is applicable across design automation platforms and that translates to all phases of the design process will help more custom designers adopt the automation technologies required to deal effectively and efficiently with the complexities of deep-submicron design. Such a standard would both speed the custom design process and ensure greater quality in the resulting designs.