White Paper

SYNOPSYS®
Predictable Success

Architecting for Productivity in Custom Design

May 2009

Authors
Les Spruiell and Chris Shaw
Custom Design Product Marketing, Synopsys, Inc.

Introduction

Advances in modern-era user interface and software design techniques have taken root in custom design EDA tools, leading to a new potential for huge productivity savings and faster time-to-results. This white paper explores these techniques as applied to the creative processes of custom and analog design, including industry-recognized use models, advanced techniques in user interface, open language programmable cells and extraction for post-layout simulation.

The Nature of Custom Design

Custom chip design is a highly creative process practiced by a small set of the world's population. Unlike the text-based or "programming" approach used in digital design, the "symbolic" tools used by custom designers are very graphical in nature as designers draw schematics, nudge layouts, and polish their final designs prior to tapeout. A given design may undergo many iterations of trial and error as the design team progresses towards the final design.

Custom design requires that the engineer have complete control over every step of the process. Therefore, productivity can be improved by focusing on the palette of commands regularly used by designers. The palette of commands available in custom design toolsets tends to be small, although each of the tools tends to be highly specialized. For instance, every designer will recognize the staple commands of "Cut," "Copy," and "Paste." However, there are more specialized commands like "Chop" that perform intelligent functions on complex objects (such as guard rings) so that objects are chopped in a way that complies with all design rules.

Improving Productivity at Every Step

Every chip design project moves through stages in which the design team will perform a small but similar set of commands many times. During schematic wiring, the commands "Add Wire" and "Add Pin" may be used hundreds of times for a complex block as the designers iterate through changes. Focusing on these highly used commands to improve and refine their usability is important as it can save designers huge amounts of time downstream.

Additionally, focusing on the usability of commands together can also bring significant productivity to the end user. As an example, allowing a "Nested Undo" within the multipath command helps designers recover from simple errors fast because it eliminates the need for the designer to end the current command, undo the error and reinitiate the original command. This is particularly important in commands that require user input like the "Add Pin" command in which the user has specified multiple pin names (that must be correctly typed in!). Any error in placement should not cause the user to have to reenter the remaining list of names after an undo. Allowing a nested undo within Add Pin prevents this problem.

The process of component placement and wiring during schematic editing is a natural place to look for productivity enhancements because of the repetitive nature of the actions. Typically, these actions are limited to placing instances, wiring nets, adding net names and changing parameters. All of these actions can benefit from new techniques in user interface design.

The process of adding instances can be helped through the time-honored “What You See Is What You Get” paradigm (“WYSIWYG”). This idea can help the designer during the placement of instances by showing the exact view of what the designer will place. This early look at the component, its parameters and the position of all of its pins helps designers avoid placing an incorrect instance or placing it in the wrong position, actions that would otherwise need to be undone.

Similarly, the time to add wires can be dramatically shortened through “high-altitude editing,” reducing the need for absolute mouse control via “Smart Connections” as shown in Figure 1. Since a well-formed schematic cannot have wires ending in space, the pins of the instances become the natural snap points for wires. Giving designers a mechanism that snaps wires to the closest pin allows designers to get close to the target and instantly and accurately make the connection without the tedious zooming and panning required by other systems.

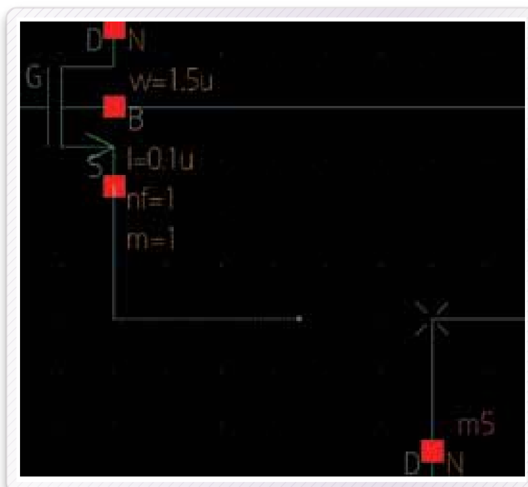

Figure 1: Smart Connect wire snapping speeds high-altitude editing

A key aspect of this Smart Connect system is the need to respect any user-provided net names and not short out nets or mysteriously rename nets without the user’s knowledge. In order to provide this capability the schematic must provide up to date, real-time connectivity that is always kept in synchronization with the rest of the database. This capability allows the system to detect unintentional shorts, or other gross (and embarrassing) wiring mistakes before the designer commits to simulation.

The concept of Smart Connect and high-altitude wiring for schematics also has benefit in the layout process. Given that the connection between two logical components in the schematic also has a corresponding connection in the physical layout, the same techniques can be used to improve productivity and reduce errors.

Another powerful user interface technique for maximizing designer productivity is known as “On-Canvas Editing.” This technique eliminates the need for additional popups or dialogs and simply lets the user click on what needs to change, and change it as shown in Figure 2.

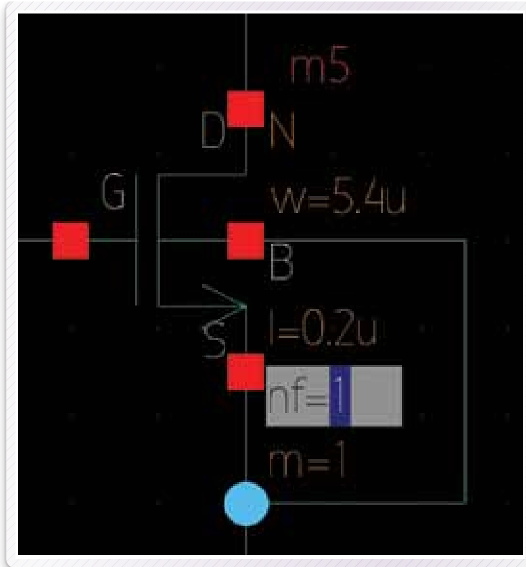

Figure 2: On-canvas editing allows instant parameter and net name changes

Applying this concept to layout requires that the connections have an explicit physical location so these Smart Connect techniques evolve to the concepts of “Auto Tap,” “Auto Width” and “Align Assist” as shown in Figure 3. These capabilities allow a designer to start wiring the layout by simply picking up the starting point’s layer and wire width, and help to keep the wires aligned with other objects in the layout. The ability to quickly and accurately start, position and end a wire from a high altitude provides dramatic increases in layout productivity.

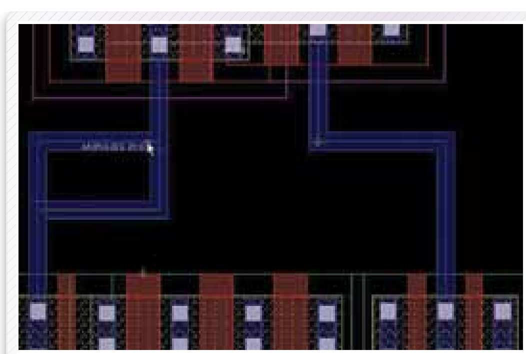

Figure 3: Smart Connect and Align Assist speed wiring tasks by showing center alignment between design objects

Other examples that emphasize the “Automation” in “Electronic Design Automation” can be found in guard ring and via array generation. Historically these two actions relied on the designer correctly guiding the software to add a welled guard ring (with appropriate contacts) to isolate sensitive components in the layout from noise. Since the guard rings depend on the actual layout of the transistors and wiring, any change to that layout triggers a change in the guard ring as shown in Figure 4.

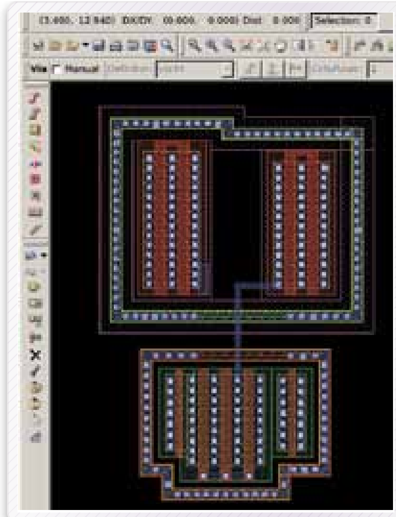

Figure 4: Generate Conformal guard rings automatically with smart chopping of metal layers

This capability gives the designer the tools needed to quickly wrap any arbitrary set of components with a minimum number of actions. If the tool provides both rectangular and conformal (polygon) guard rings, along with net names, spacing and well specification, the user sees significant improvements in productivity. If the layout requires a change, the user can simply delete the guard ring, make the changes and add the ring back with a minimum of effort.

Similarly, the creation of one or more vias between layers can benefit from similar technologies. Vias must follow the process design rules and since that information is known to the system, along with information such as net names and wiring layers, vias can automatically be added through a simple click or windowing command, as shown by the selection in Figure 5. The system can use the dimensions of the area along with the design rules of the process, to place the vias correctly the first time. Should an ECO trigger a change, the actions to reconnect the new wiring is as simple and as productive as it was the first time around.

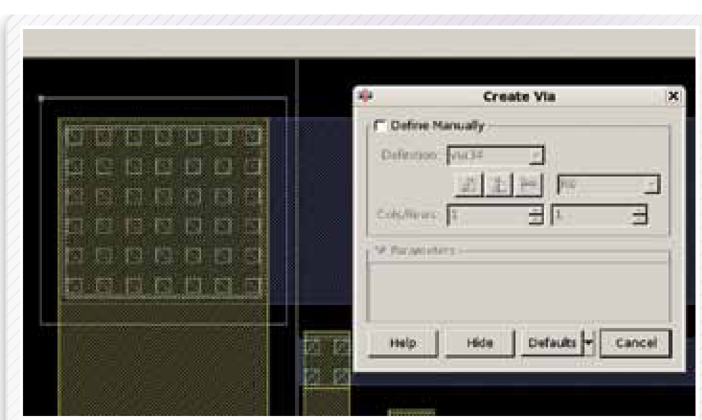

Figure 5: Automatically create design rule-correct via arrays based on net name

Custom Designer's interactive bus router utility provides a highly productive way to manually route large numbers of bus wires while making via insertions quick and easy. The utility lets designers route busses, complete with flight lines, and using the command's context sensitive menu choose between a number of different via insertion patterns. Using the command's "diagonal via down" pattern automatically inserts the proper via, and begins routing on the layer below, as shown in Figure 6.

Figure 6: Interactive bus router showing flight lines and automatic via insertions

As we have seen, designers often iterate through a small set of commands many times. It is therefore possible to derive extra productivity by helping the designer remember past actions that can be repeated. Proper history management, that is, the action of keeping a list of all commands used in a session, allows rapid access to replayed commands without hunting through the menu hierarchy. As time progresses, the latest commands move to the top of the list while the infrequent commands move to the bottom. The last command is always on top of the stack, ready for quick replay. For instance, if the user performs a "Copy, Paste, Paste, Move, Copy, Align, Move, Save" command series, then the history toolbar would present to the user an iconified toolbar of "Copy, Paste, Align, Move, Save," simplifying the user's interaction with the system.

Keeping the history of changes in the session, regardless of activity, allows designers to explore avenues, roll back changes, modify their approach and benefit from the system's long and accurate memory.

Enabling Easy Adoption

Advances in user interface design concepts like high-altitude editing bring with them a host of benefits ranging from fast and better analysis algorithms to better interaction, visualization and a better understanding of the problem to a more complete verification of a design's efficacy.

One compelling reason to update to a new toolset is to derive extra productivity from advances in the design of user-driven software. One key technique involves the tried and true GUI design tenet of "Never Surprise the User". This tenet has been historically proven time and again and is best evidenced by the example of a proffered competitive solution, which at the onset looked like the solution of the modern era. In reality it turned out to have so many surprises that the cost of adoption was so high that most companies were simply left out in the cold.

The key technique to achieve this goal is to provide a common industry "look and feel," including common "bindkeys" (keyboard shortcuts), common library devices and structures and a common set of industry-accepted commands and associated menus. Since the number of semiconductor designers is limited to a small subset of the overall population, use models have been developed and become familiar in the industry over the years, appearing across many tools from many companies. Any alternative system must provide for rapid adoption by users with an existing knowledge of historic designs tools. The primary goal is to give the user the ability to be up and running in minutes with little or no learning curve.

Another aspect of productivity in new design software is to minimize or eliminate "Information Overload." In the past, CAD tool developers, while trying to provide the maximum flexibility in their tools, presented far too much information to the user through a profusion of windows and dialogs.

Layout engineers went from little or no information to multiple (flashing, beeping and hidden) windows and found that the majority of their interactions with the system (e.g., a mouse click or a keyboard press) were focused on *manipulating the user interface and not the design*.

Quite quickly, it became apparent that the window you needed had become buried underneath all of the other windows or that some modal dialog[1] had locked the system but was also buried in the mass of overlapping windows. The only choice for the user is to stop what they are doing on the design and move windows around until the right window is found.

Other productivity-sapping annoyances include unnecessary questions from the application to the user. Although the original author thought that giving maximum flexibility to the user was a good idea, that user will rapidly grow frustrated with answering the same question when 99.9% of the time the answer is always the same. It may only be one mouse click but when repeated many times a day, for days, months and years on end it takes a toll on the user's productivity[2].

Summary

The goal of this paper has been to point out some of the new techniques that are improving custom design productivity. That process itself is a highly creative, interactive process that benefits from new thinking about user interfaces, more productive commands and an intuitive and easy-to-learn system.

As with everything in the electronics industry, the pace of change in design processes always drives the need for newer and better tools. These techniques can bring new levels of productivity and reduce the barriers to adoption. The fun is back in IC design again.

---

[1] A Modal Dialog is a dialog that requires an answer from the user, and has locked out all other interactions with the system until the user responds.

[2] A powerful example is found in a popular computer operating system that insists on notifying the user that they just printed a document and requiring that the user click to make the notification go away. This is particularly frustrating considering that the *last action the user performed was to send the document to the printer!*