White Paper

SYNOPSYS®
Accelerating Innovation

Accelerated Layout for Analog/ Mixed-Signal in Nanometer SoCs

October 2012

Author
Lyndon Lim
Synopsys

Abstract

The complexity of nanometer CMOS design rules has greatly complicated the task of creating and integrating analog/mixed-signal functions in SoC designs. Traditionally, custom layout is tedious and inflexible once started. As a result, design teams don't start layout until the circuit design is nearly complete, and typically work under strong tapeout timeline pressure. However, a much more accelerated approach is possible using automated methods to reduce the total effort needed, enable layout to proceed concurrently with circuit design, and typically produce more optimal layouts—especially for smaller die sizes.

Introduction

Everyone involved in SoC design today understands that timely physical design of the custom sections primarily RF, analog and mixed-signal, but also certain high-performance digital functions—has become much more difficult as designs move from 90- and 65-nm to 28-nm and below.

Part of the challenge is technical: the variability of nanometer silicon is so great that it threatens to make many analog functions non-manufacturable, especially in high-performance designs where parasitics and device characteristics have a critical impact. To cope, foundries have introduced more complex design and DFM rules, and designers have also introduced more elaborate circuit architectures with more devices. Both of these add to the work needed to complete designs.

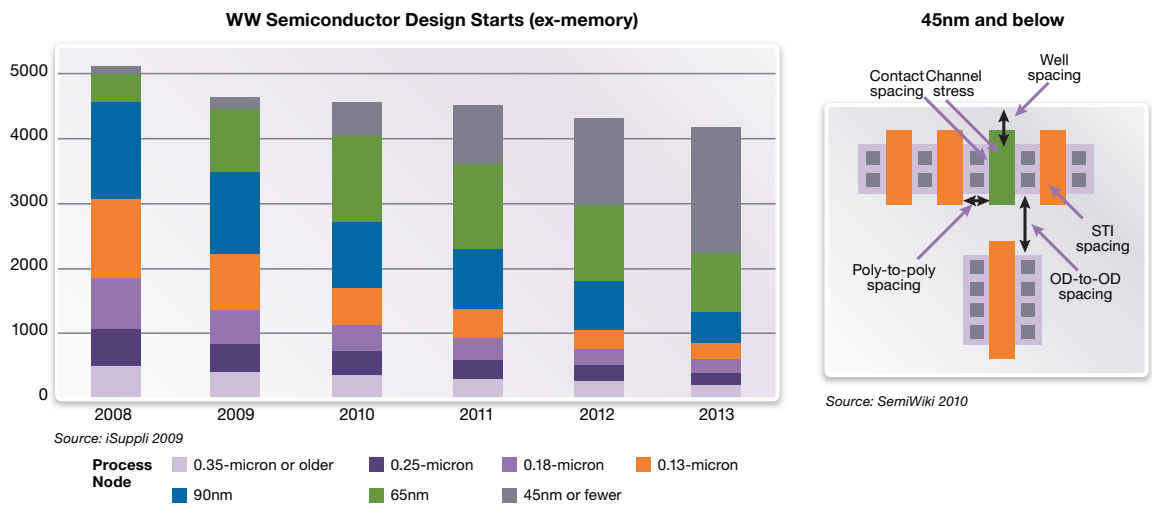

Figure 1: SoC Analog/mixed-signal design must contend with increasingly complex process design rules

Another contributor is the market: the move to advanced geometries has largely been driven by ultra-competitive "winner take all" consumer SoC markets where rapid delivery is essential. In these markets, cycle time is crucial; development teams simply can't afford several extra weeks to finish layout, let alone find and fix layout-related design or architecture problems.

The result is a perfect storm in which SoC analog design teams face a seemingly-unsolvable paradox: more transistors and more design rules for the same (or less) time and effort.

This is a major challenge. There is no way that incremental methodology improvements can keep pace for long; instead, major advances are needed.

Custom Layout Methodology (1): Traditional Flow 1990 – Present

Custom layout has traditionally been done mostly by hand, with a skilled designer placing each device and routing each wire, generally working bottom-up one subcircuit or sub-block at a time until the entire design is finished. Innovations over the years for specific tasks, such as parameterized cells ("PCells"), and special-array generators such as modgens, MCells™, or pattern cells, helped improve productivity.

This traditional methodology gives design teams very fine-grained control over the exact geometry of the layout to optimize it for the electrical requirements of the circuit; but it has the disadvantage that the designer's time is also needed to make the geometry conform to the process design rules—a repetitive and detailed task arguably better suited to a computer than to a creative human design engineer.

Nevertheless, for moderate-size circuits on relatively uncomplicated process rules, this approach has been effective. The challenge is that the nanometer SoC world increasingly presents the opposite circumstance—large complex designs with very complicated design rules. This exposes the fundamental limitations of the traditional methodology: that creating a layout takes a long time; and that once a layout is created, changing it is almost as much work as starting over. Any analog design engineer who has a taken a major circuit update to his or her mask design team in mid-layout has experienced this first-hand.

Because of this, layout performed while the circuit design is still changing is essentially throw-away work, and therefore nearly all design teams wait until the circuit design phase is nearly complete before starting layout. So the time for circuit design and the time for layout are essentially additive, with both on the critical path to tapeout.

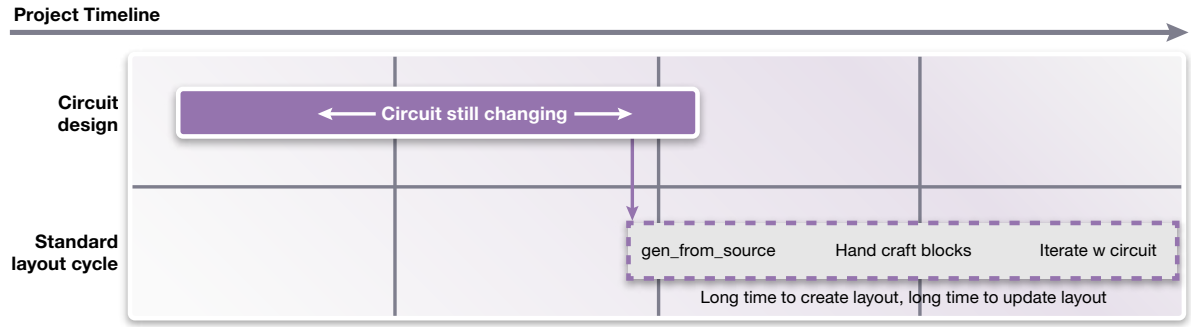

Figure 2: Traditional custom layout does not start until the circuit design is nearly complete

Furthermore, in this sequential flow the layout phase is usually done under substantial time pressure, since it is basically the last task before tapeout. This has an indirect effect on quality. Without time for layout exploration or further optimization, normally what tapes out is a minimally-acceptable layout, one which meets the primary performance specifications but may well waste important secondary goals such as die area, and occasionally robustness. Furthermore, the chances for precise circuit layout optimization—for parasitics, for example—are compromised, since the circuit design team doesn't see an accurate layout until very late in the design cycle. In addition, the rigidity of layout can even discourage designers from improving or reconsidering the original

circuit topology itself. This, more than anything, can lead to sub-optimal performance, if not outright failure to meet specifications or schedules.

Other growing limitations of the traditional flow in the nanometer SoC world are that its block-by-block nature provides little support for floorplanning (though larger analog/mixed-signal designs often need it); and that handcrafting layout to a specific set of design rules means the IP is not easily portable to other technologies, or even to updated foundry rulesets for the same technology.

Custom Layout Methodology (2): Accelerated Layout Flow 2010 – Present

The emergence of new custom layout automation methods is an important development.

Any automation helps of course. However, incremental automation which merely improves productivity in the traditional flow by small percentages will be quickly made irrelevant by increases in circuit and design rule complexity. The emergence of larger-scale automation which facilitates a significantly more powerful design flow not only has greater appeal, but also a substantially larger practical impact on the problems at hand.

The key advance is the ability to quickly lay out not just a single structure (e.g., a differential pair or current mirror) in the presence of only a few localized rules, but rather an entire design hierarchy in the presence of all rules and requirements (complex design rules, area, signal flow, matching, etc). Such a capability enables not only the rapid creation of layout, but also rapid updates and assimilation of changes to circuits, floorplans, pinouts, PDKs, design rules, etc.

Automating layout for an entire design hierarchy makes the “throw-away” nature of layout performed on a changing design acceptable. This in turn makes possible an “accelerated” design flow, in which layout starts much earlier in the project and runs concurrently with circuit design even though the circuit hierarchies themselves are still evolving, some parts of which may not yet exist. Acelerated layout starts sooner, finishes faster, and accommodates changes and updates easily—both in the layout and the circuit topology itself. By the time the traditional layout cycle would be ready to start, the accelerated cycle is nearly finished.

In this way the circuit design team can accurately see early on how their design will look. Layout-related electrical effects can be simulated at any point in time, and this information can be used to drive refinement of the circuits and layouts, as appropriate. Gross errors or serious performance problems are detected early instead of late in the design cycle.

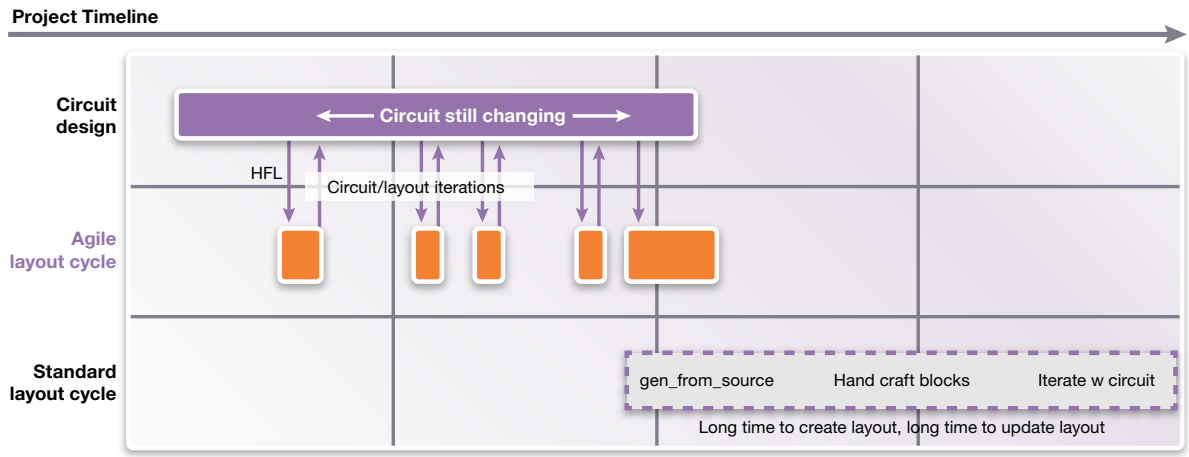

Figure 3: Accelerated layout starts sooner, finishes faster, and produces a better layout

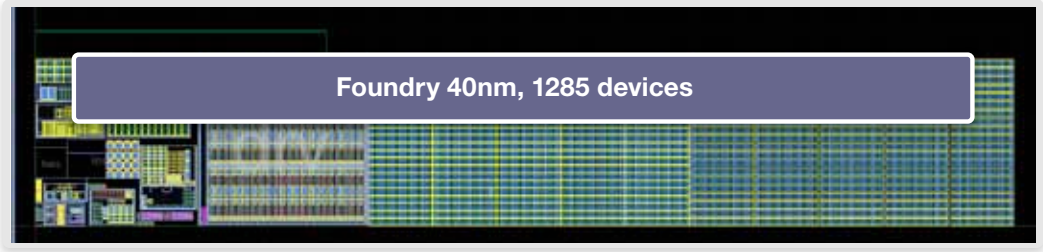

Figure 4: Example of accelerated layout benefits. This hierarchical layout was created 4–6 weeks before layout would normally have begun. It yielded important information to the circuit designer early in the design process, supporting desirable and necessary changes to the circuit topology.

Beyond general predictability and avoiding surprises in layout, the accelerated flow offers several advantages for the nanometer SoC world:

- **Significantly lower layout effort** via automation of tasks previously done by hand, especially in the presence of very complex design rules. Iterative manual cycles to rework and fix DR violations are avoided.
- **Significant time-to-market improvement.** In an accelerated design flow, layout not only takes less total time, but also starts earlier and finishes soon after the circuit design is complete.
- **Typically 10–20% die size reduction** at 28nm vs. handcrafted layout.

This last point is slightly counterintuitive. When we began working with 40nm designs, we noticed that Helix layouts generally came in smaller than expected. The answer turned out to be “white space reduction.”

In the traditional flow, layout teams produce blocks, assemble them together, resimulate with parasitics, and tape out. Layout iterations are slow, and little time is available to hand-optimize the exact shapes and positions of the individual blocks for area efficiency. Furthermore, the human designer’s ability to account and optimize for all design rules at once, while also attempting to satisfy all electrical, yield and layout structure requirements, has become extremely difficult; therefore, layout sub-structure typically becomes “guardbanded” with white space as a means of managing layout complexity. As a result, nanometer custom IC layouts often contain noticeable unused space between many of the blocks, or what one customer dubbed “doily designs.”

However, in the accelerated flow the layout iterations are very fast. Furthermore, the layout schedule is gated not by the layout effort itself, but by the circuit design schedule, allowing time for optimization. In a sense, unacceptably slow iterations late in development are replaced with acceptably fast ones early—resulting in a better design. The ability of an automated tool to optimize and globally account for large numbers of complex design rules is a key factor. We expect the advantage to increase with larger designs and sub-28nm geometries.

Two other advantages to the accelerated flow are:

- A much-improved floorplanning flow. The ability to rapidly lay out an entire design hierarchy, even with tens of thousands of devices, makes floorplanning a large design a much more accurate and predictable process. Instead of guessing at the sizes, aspect ratios, and best pinouts of the blocks, the team simply places the entire design at the device level in order to guide the floorplanning process.
- IP migration. In the agile flow the layout can be regenerated quickly to new design rules. Blocks whose circuit architectures have not changed can be ported immediately. Those with circuit architecture changes may require updates; but even this is fast in this flow.

Note the accelerated flow does not preclude incorporation of manually laid out blocks into the hierarchy, nor is it limited by such.

Automation Requirements for Accelerated Layout Flows

The central concept of an accelerated layout flow is the ability to create and modify large layouts quickly with little human effort. To do this, automation is essential; without it, layout iterations are slow, manual-intensive, and rigid. The key automation criteria are:

- **The layout engine must be capable of handling the entire design hierarchy,** not just individual blocks If only part of the design can be handled by the automated tools, then the rest of the design must be laid out manually at each iteration. Without automated iterations, the project team cannot start layout until the circuit design has stopped moving—essentially a return to the traditional flow. While adding incremental automation to the traditional flow is not a bad thing in itself, it does not yield the productivity, time-to-market, die size and other advantages of the accelerated flow. Implied corollaries to this are hierarchical support, high capacity, and fast runtimes.
- **The engine must produce high-quality layouts.** This includes respect for design requirements such as matching and signal flow structure, inclusion of advanced/sophisticated primitive device structures, DRC/LVS-clean, proper generation of complex geometrical features such as wells, triple-wells, guard rings, etc., and adherence to deep nanometer design practices such as grids and density control. If the layouts generated by the engine must be corrected by hand, then rapid and low-effort iterations are not possible—again forcing a return to the traditional flow.
- **The constraint architecture and use model must be extremely good.** Constraints are how the designer tells the tool what he or she wants the layout to look like. If the constraint architecture and use model are cumbersome, then the designer will spend as much time grappling with the tool as it would take to simply do the layout by hand. The constraints themselves need to be simple, transparent, intuitive and modular; connection with layout outcomes must be easily predictable, and the engine must be consistently able to find high-quality solutions that meet the constraints. Weak fundamental constraint architectures were a persistent problem with early layout automation tools.
- **Must support customer and foundry production PDKs.** Many previous approaches to layout automation either relied on sets of built-in generic device structures or required users to essentially flatten their primitive devices to static polygons. These restrictions are impractical in real-world advanced design processes. Full-function support for customer and foundry PDK devices allows customers to utilize their own advanced devices and enables out-of-the-box layout automation through leveraging of foundry-standard PDKs, such as TSMC iPDKs.
- **Open and interoperable system.** Use of the OpenAccess database, standard data formats where standards exist, and non-proprietary formats and/or ASCII files where standards don't exist, etc. Nanometer SoC design is so complex that no one vendor can supply everything needed.
- **Nice to have: fully-automatic generation of a good starting point.** The traditional starting point for layout is "gen_from_source" or an equivalent; not really a layout as much as a "bill of materials" for the layout. But if the automated tool can produce an actual starting layout, it further accelerates an accelerated layout flow and potentially enables capabilities such as early accurate estimate area and electrical effects due to layout (various parasitics, changes in device characteristics, etc).

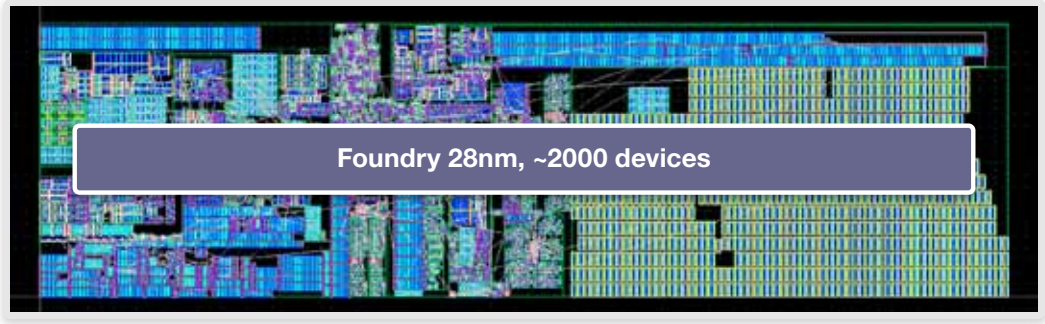

Figure 5: Design example. This design was produced using Helix First Look mode; the inputs were a schematic and PDK, with no manually-entered constraints; outputs created by this mode were the layout and the constraints associated with it. Area was within 1% of final tapeout area. This design used approximately 2,000 devices with 136 unique cells and 12 levels of hierarchy.

Summary

The drive to integrate analog/mixed-signal IP in sub-28nm silicon faces a major bottleneck in the effort and rigidity of the traditional manual approach to custom IC layout. However, a much more accelerated methodology, enabled by automation that supports very fast and low-effort layout iterations, can produce higher-quality designs in dramatically less time and with a fraction of the effort needed by the traditional flow. As design rule complexity continues to increase, and as device counts grow, the accelerated approach vs. the traditional approach will have a greater and greater return on investment.

Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com

©2012 Synopsys, Inc. All rights reserved. Synopsys is a trademark of Synopsys, Inc. in the United States and other countries. A list of Synopsys trademarks is available at http://www.synopsys.com/copyright.html. All other names mentioned herein are trademarks or registered trademarks of their respective owners.

10/12.RP.CS2288.