Accelerating Block Physical Signoff for Advanced Process Nodes

IC Validator Physical Verification Fusion

Authors

Manoz Palaparthi
Product Marketing Manager,
Senior Staff, Synopsys

Ed Roseboom
Product Engineer, Senior Staff,
Synopsys

Introduction

Physical verification is often on the critical path to tape-out for today's advanced process node designs. Ever-growing design complexity, shorter design cycle times, process advancements and increasing verification requirements are driving the need for faster, more efficient physical verification flows for the leading-edge designs at 7 nm, 5 nm, and below. Identifying and fixing design issues early during block implementation greatly improves the ability to meet project tape-out schedules. Physical verification fusion, also known as in-design verification, is an established flow with well-understood value addition in the design flow for advanced node designs, enabling clean block handoff for full chip signoff.

In this white paper, learn about the latest advances in IC Validator physical verification fusion that further accelerate block physical signoff: Live design rules checking (DRC) for interactive and signoff-quality DRC checking, Explorer DRC for fast identification of key design weaknesses, metal fill-aware timing closure, and best practices for DRC checking and metal fill for ECO flows.

Physical Verification Fusion: Overview

IC Validator physical verification Fusion redefines the boundary between place-and-route and physical signoff, sharing the IC Validator signoff verification engine in the physical implementation tool flow. It enables designers to accelerate block closure faster DRC, fill convergence, and rapid ECO flows. There are several stages in the implementation flow that benefit from physical verification fusion, from design planning to post-route and ECO steps.
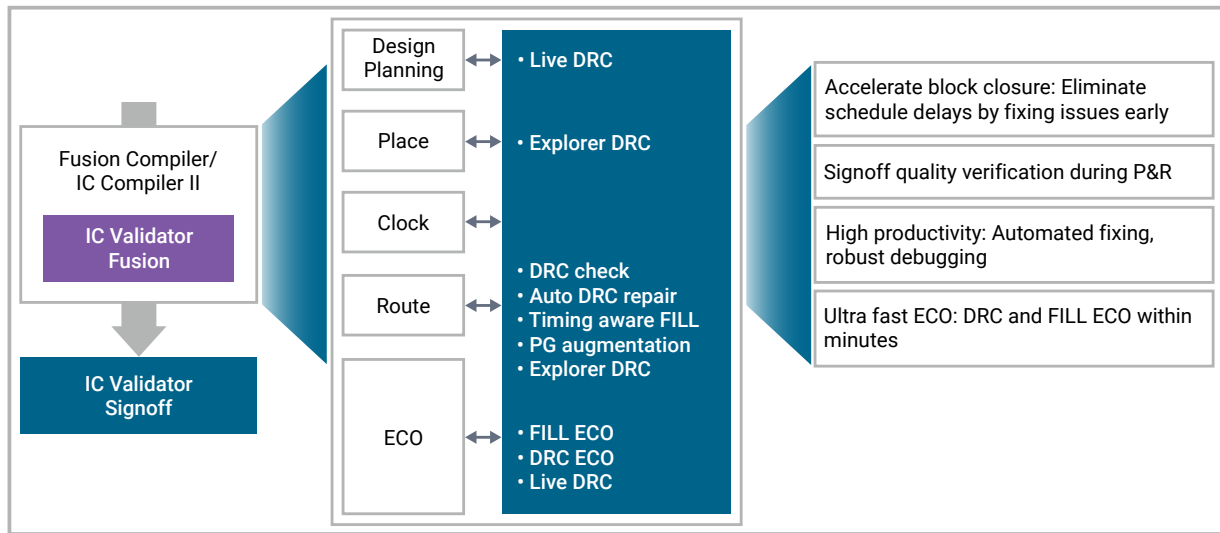
Figure 1: IC Validator physical verification fusion flow

IC Validator Physical verification fusion has been a staple in the designers' physical implementation flows for several years. In the following sections, we highlight the new productivity features in the recent tool releases of IC Validator, available in both Fusion Compiler™ and IC Compiler™ II.

Explorer DRC

Explorer DRC™ rapidly verifies the design and provides useful and actionable DRC feedback to fix the problems quickly. Today's leading-edge designs consist of hundreds of blocks, such as place and route blocks, analog cells, memory, third party IP, and I/O cells. When the design is compiled, there are often high-level problems, such as missing blocks and block placement errors that must be identified and fixed. It is common to encounter millions or billions of DRC errors the first time the chip is compiled and run, due to a handful of high-level issues. IC Validator Explorer DRC automatically runs fundamental rules from the foundry runset and additional methodology to rapidly asses the health of the design. Explorer DRC brings a dramatic change in compute efficiency on dirty designs versus the traditional flow: five times faster runtime with five times few cores used.

Explorer DRC is now available in the fusion flow both in IC Compiler II and Fusion Compiler.

DRC error heatmap provides rapid and intuitive visual assessment of the design. Within minutes, designers can often identify the macro-problems to fix (such as overlaps), instead of getting stuck in the weeds of millions or billions of errors. The heatmap shows hot areas (where there are many violations) in red, progressing to cool areas (with relatively few violation) in blue. Shown below is a sample DRC heatmap that shows the complete DRC profile of the design in a single view.
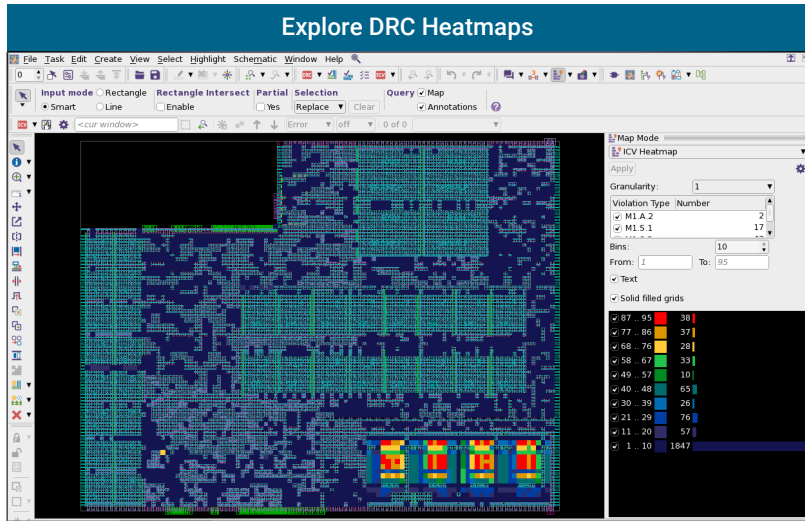
Figure 2: DRC heatmap in Fusion Compiler

Automatic DRC Repair

Automatic DRC repair is an important productivity enabler in place and route flows. The DRC violations re identified based on the rules from foundry provided signoff runsets. The signoff_fix_drc command uses the guidance form verification and automatically performs targeted DRC fixing. The tool can automatically fix as many as 90% or more DRC violations within minutes. This reduces the iterations between place-and-route and verification, thus accelerating post-route and post-eco convergence.
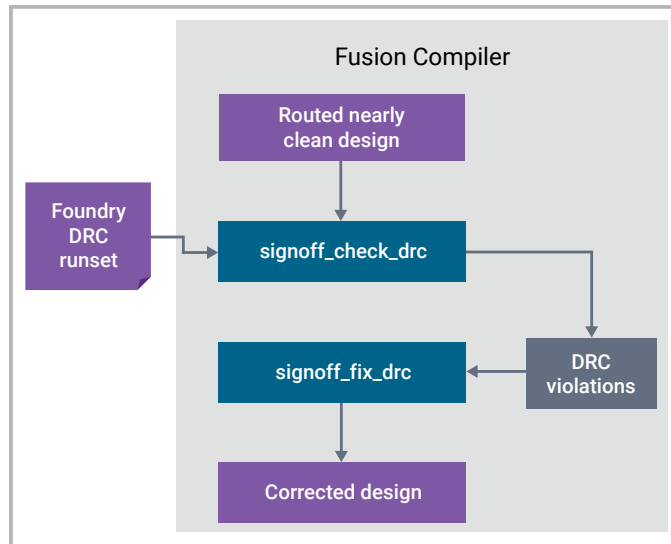

Figure 3: Automatic DRC fixing flow

Incremental DRC for ECO Flows

During ECO, designers need to work on a sub-section of the design that has layout changes. Running a DRC verification on the full design is redundant and time consuming. In IC Compiler II and Fusion Compiler, the ECO changes are continuously tracked. An incremental ECO DRC can be initiated automatically to verify the portions of the layout that have ECO changes. With incremental DRC for ECO flows targeting a specific area with ECO changes, a DRC check can be shortened from hours to minutes. Additional features of incremental DRC include automatic detection of changes areas/layers, and options for layer, rule, or area.
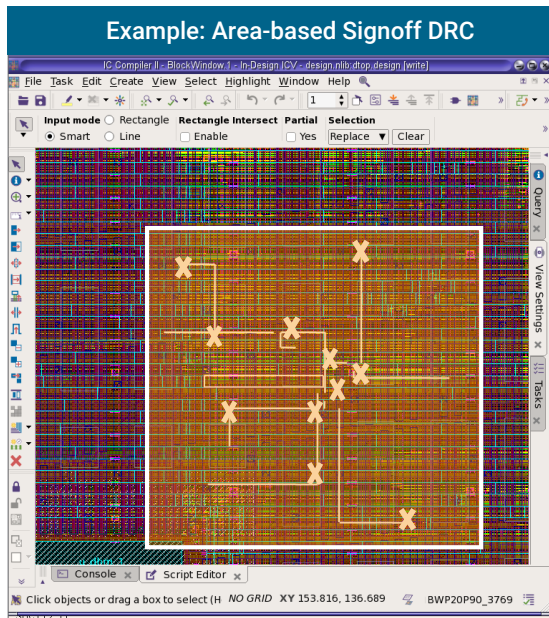
Figure 4: Area-based signoff DRC

Live DRC

When small, manual ECO changes are made, the changes can create a cascade of new DRC violations. Live DRC is a new feature in IC Compiler II and Fusion Compiler that provides DRC feedback within seconds, using foundry-qualified signoff runsets. The place-and-route tool sends the data within the view window to a streamlined IC Validator Live DRC engine. The signoff DRC checks are executed in a few seconds and any resulting violations can be viewed in the error browser and fixed immediately. Designers make the layout changes, run DRC checks, view violations and, fix them - all in one environment, and in just a few seconds. This capability immensely improves the designer productivity during ECO flows.

The key advantages of Live DRC in IC Compiler II and Fusion Compiler are: interactive and signoff-quality DRC checking during place-and-route, view the DRC violations in the error browser, a custom toolbar and recipe-based DRC for additional productivity improvement.
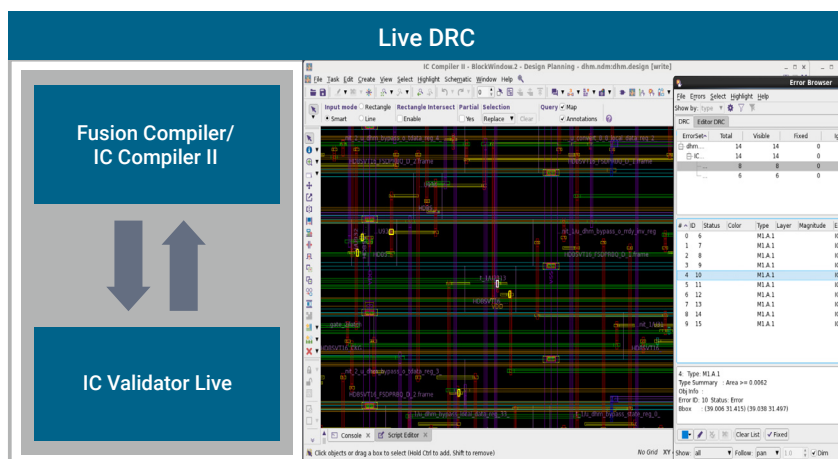

Figure 5: Live DRC checking

Strategy for Metal Fill in Optimization and ECO Flows

As the requirements for advanced process nodes continue to become more stringent, density DRCs for metal fill have also become increasingly more complicated. A specific part of this challenge is integrating metal fill into optimization, timing closure, and ECO strategies. A common issue while doing timing optimization is that after adding metal fill, the timing gets much worse.

A solution to this challenge is metal fill-aware timing. With this approach, the metal fill is done before optimization so that the optimization addresses timing. Though this process can enhance predictability, it is also worthwhile to protect a small number of timing-sensitive nets from metal fill, as they may cause timing problems. With Fusion Compiler, these critical nets can be precisely fenced off to prevent timing issues occurring during route optimization while avoiding density DRC violations. This feature works on metal fill in the same layer and vertical layers and can even fix density errors by re-inserting small amounts of metal fill into selected areas.

Completely removing the metal fill from a critical net for timing purposes is not always viable. There are new rules that dictate a maximum open area in addition to having a necessary amount of metal fill or routing in an area. This makes managing both density and timing a challenge. In these cases, there is a need to balance the timing requirements and metal fill. Introducing metal fill-aware timing early in the timing closure cycle makes solving timing challenges much more predictable. Particularly, TNS and violating setup paths become more predictable further along in the flow as opposed to a huge uptick in violations (often around 50%) using the traditional flow.
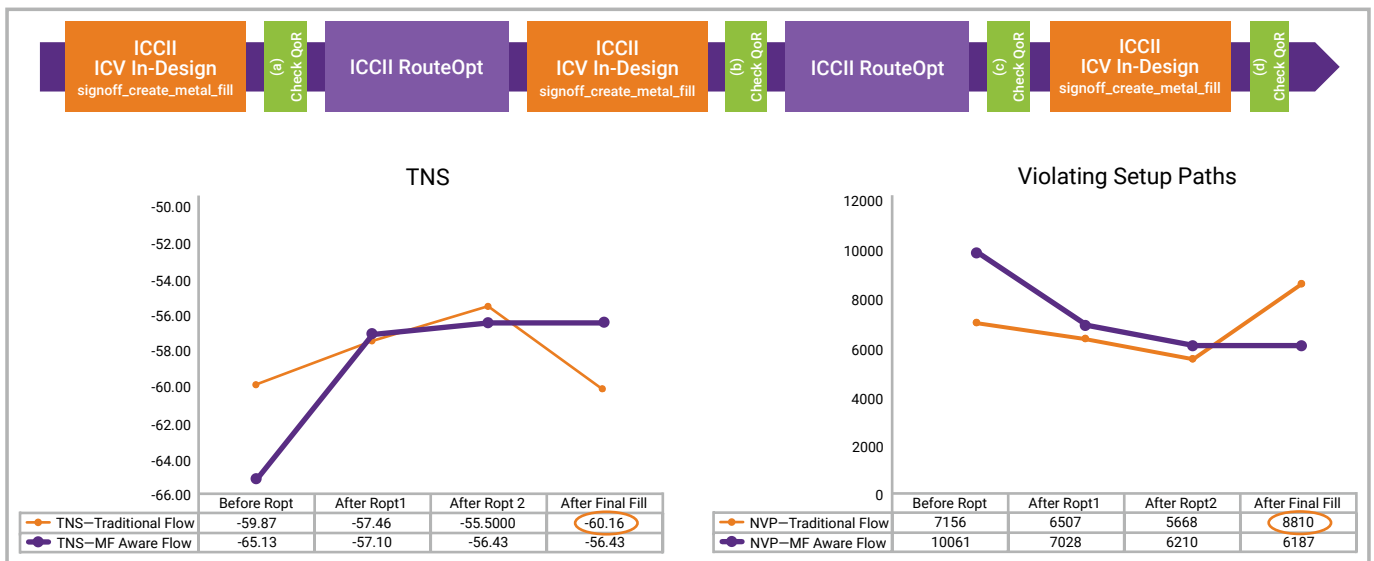

Figure 6: Metal fill aware timing improves predictability

Conclusion

Growing design and manufacturing complexity at advanced process nodes is driving the need for a highly productive physical verification solution. The new innovations in IC Validator physical verification fusion, Explorer DRC, Live DRC, automatic DRC repair, incremental DRC, and metal FILL features, deliver enhanced productivity to accelerate physical verification closure in block-level design phase.