

# Accelerating Physical Verification with an In-Design Flow

May 2009



**Author**  
Elango  
Velayutham  
**Implementation**  
Group,  
Synopsys, Inc.

## Introduction

Physical designers are responsible for verifying and ensuring that their designs are clean of Design Rule Check (DRC) violations before delivering the design to the foundry. Since semiconductor foundries use sign-off DRC rules to validate the manufacturability of a design, physical designers must use the same sign-off DRC rules to avoid any missing checks. In the sign-off physical verification flow, designers typically stream out the design and separately run physical verification tool using foundry-supplied sign-off DRC runsets. This method can be extremely time-consuming. It requires time-consuming stream out and stream in process. Also because physical verification is run outside of the implementation tool, it can result in excessive iterations between the implementation tool and physical verification tool to fix DRC errors. This can impact the overall project schedule. In addition, this flow is also very inefficient for ECO changes because the physical verification is unable to limit its DRC checking to only the ECO affected regions of the design.

This paper discusses the need for a concurrent physical design and physical verification flow, also known as an in-design physical verification flow. This flow improves the overall turnaround time and ease of use of the physical verification process. This paper also provides a production-proven example of this flow.

## Physical Designer's Challenge

The biggest challenge for physical designers is how to verify and deliver a design that is free of DRC violations while meeting their tapeout schedule. Tool performance is a key factor for IC designers in meeting their schedules. Consequently, CAD managers and IC designers are looking for solutions that will reduce their overall turnaround time without sacrificing accuracy. Since sign-off physical verification comes at the end of the design cycle, turnaround time is even more crucial. The physical verification solution needs to provide DRC error information as early as possible in the design process, to minimize iterations, and to give designers more control over how much of the design is verified at a given time.

The overall runtime of the physical verification flow is impacted by the size of the design, the number of design rules and the complexity of the rules. For example Figure 1 shows how larger design size impacts the overall physical verification run time.

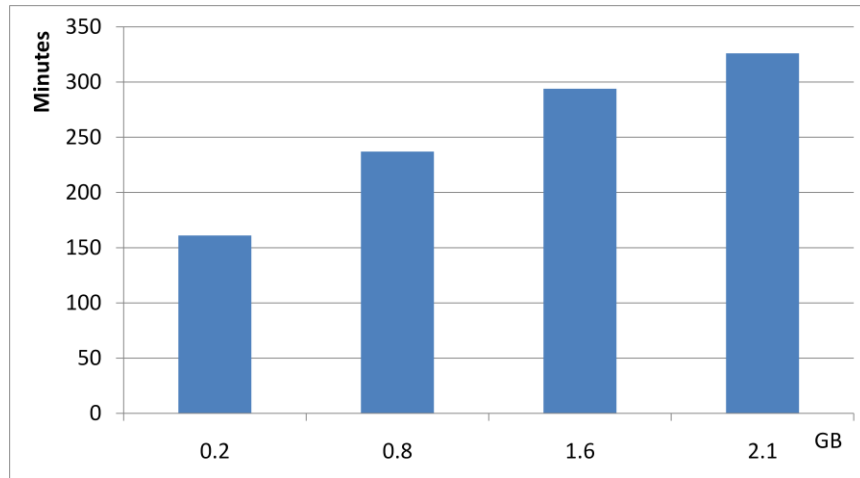


Figure 1: Increasing Single CPU Runtime as Design Size Increases

The number of design rules also impacts the overall runtime. The number of foundry design rules is increasing dramatically from one technology node to the next and this increasing number of rules also increases the overall runtime of the physical verification stage. A comparison of the number of rules at each TSMC technology node is shown below in Figure 2.

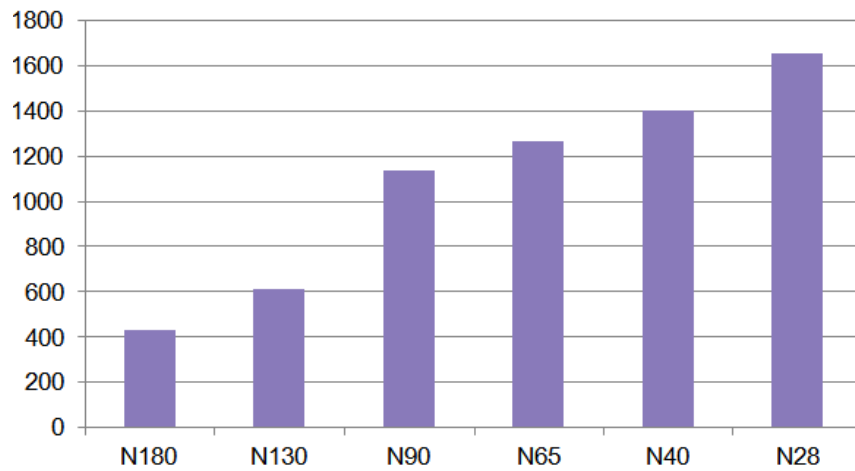


Figure 2: Number of Design Rules at Each Technology Node (Source: TSMC)

Another factor causing longer runtime is the implementation of complex rules in the DRC runsets. Complex rules must typically be written using many separate commands which increases the overall run time. The example in Figure 3 shows a complex rule.

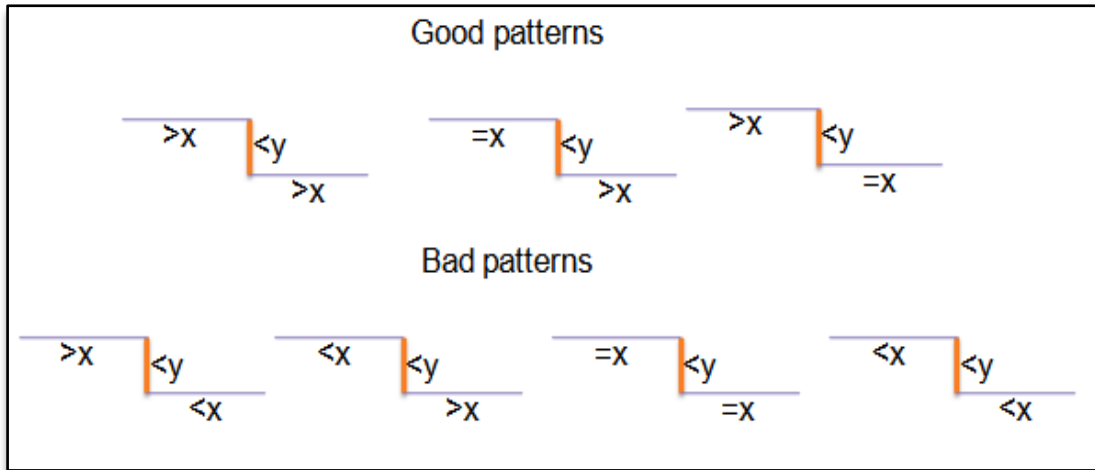


Figure 3: Complex Short Edge Rule

### Traditional Flow: “Implement - Then - Verify”

Designers have traditionally run the DRC check after initial place and route, after each ECO, and for final sign-off. During the design implementation phase (place and route and ECO steps), designers typically either use a DRC utility provided by the implementation tool or stream out the design to verify by the foundry-qualified DRC tool. For final sign-off physical verification designers always stream out the design to run physical verification using foundry runsets for complete DRC checking. Figure 4 shows this traditional flow.

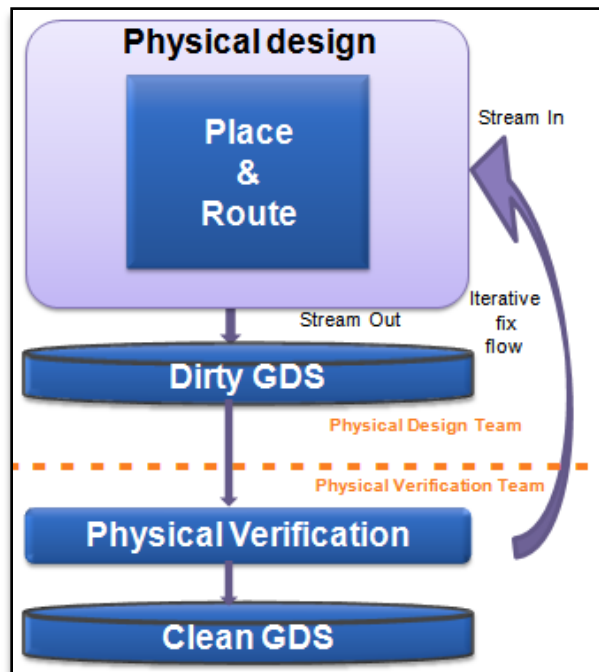


Figure 4: Traditional Implement-Then Verify Flow

As can be seen, closing timing and generating fill to meet all density requirements takes many iterations. For this case study, each iteration required approximately 6.5 hours (1 hour for fill generation, two 2.5- hour runs for timing analysis and 20 minutes for the density check). Adding this time for all the iterations totals approximately twenty-six hours of processing.

This implement-then-verify flow has many disadvantages:

- ▶ Time-consuming stream out needed to run physical verification
- ▶ Feedback loop between tools increases overall turnaround time
- ▶ Iterative fixing flow adds time to physical verification stage
- ▶ Physical verification tool knowledge required for physical designers

Figures 5 and 6 show how much time is spent on the stream out step for different design sizes and for the number of instances in the design.

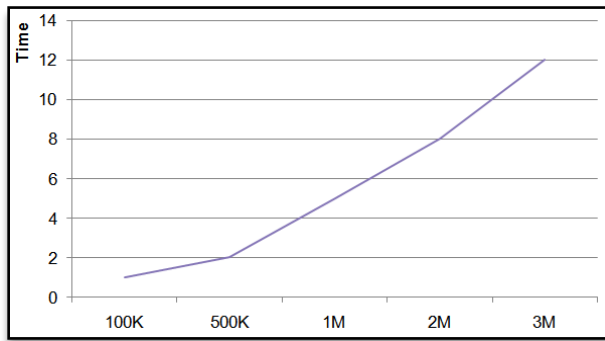


Figure 5: Stream Out Time (minutes) Versus Number of Instances

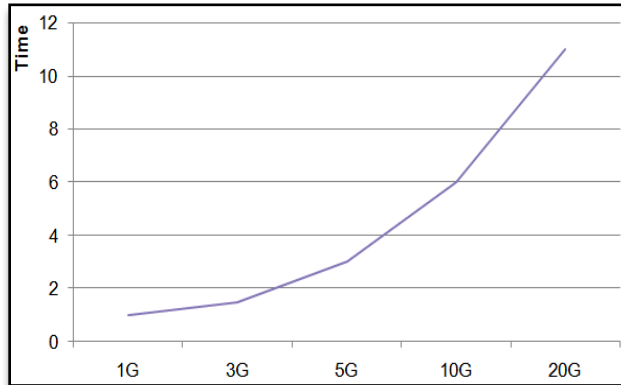


Figure 6: Stream Out Time (minutes) Versus GDSII Size

While the time for a single stream out step does not seem like much, the physical verification process for a typical design will require multiple iterations, thus incurring multiple stream outs.

## The Need for Physical Verification Within the Implementation Flow

The ability to run physical verification within the implementation flow is important to physical designers because it will simplify physical verification and more importantly, reduce the overall turnaround time. The in-design DRC flow should have these features:

- ▶ Simple for designers to setup and run
- ▶ No stream out step to avoid the feedback loop
- ▶ Ability to find and fix errors directly within the implementation tool
- ▶ Ability to check all foundry rules within the Implementation tool to obtain sign-off quality results
- ▶ Ability to check only routing layers for faster design closure
- ▶ ECO changes verified in minutes
- ▶ Ability to run with multiple CPUs to reduce overall turnaround time
- ▶ Physical designers not required to have detailed physical verification tool knowledge

## Synopsys In-Design Physical Verification Within IC Compiler

The Synopsys In-Design physical verification flow delivers all of the required features needed for physical verification within the implementation flow. IC Compiler is linked with IC Validator signoff DRC link to provide sign-off quality DRC checking using foundry qualified runsets. DRC checking is run on the same design database that IC Compiler uses, which eliminates the need to stream out the design. This saves a significant amount of time and eliminates any possibility of data transfer error. Figure 7 shows the In-Design physical verification flow with IC Validator DRC checking run from within IC Compiler.

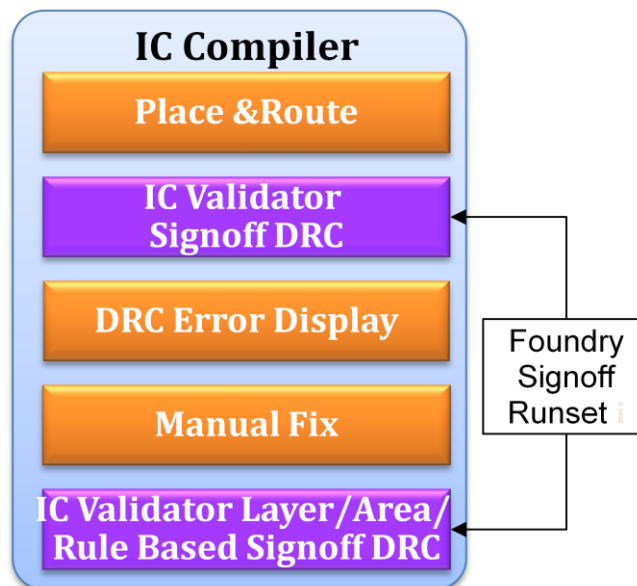


Figure 7: IC Compiler-IC Validator In-Design DRC Flow

By using the In-Design DRC flow within IC Compiler, physical designers can now perform DRC checks more efficiently and more easily. They can quickly check that all the routing layers are clean of DRC errors before running DRC checks on all the layers. There is only a one time setup required to specify the foundry runsets. ECO changes can be verified in minutes by running DRC checks on only the regions of the design affected by

the ECO changes. The IC Compiler error browser can be used for debugging DRC errors. The DRC checks can be run on multiple CPUs to reduce the overall runtime by using the distributed processing option.

### Key Features of the In-Design DRC Flow:

- ▶ No stream out and stream in of the design:  
The in-design DRC flow within IC Compiler eliminates the need for stream out to run physical verification. Instead, the in-design DRC flow uses the same Milkyway database that IC Compiler uses, and IC Validator outputs the Milkyway error cell for debugging. Since many iterations are involved in physical verification, multiple stream out and stream in steps are eliminated, thereby saving a significant amount of time.
- ▶ Layer- based DRC checks:  
With the in-design DRC flow, DRC checks can be run on specific layers instead of all layers. For example, during ECO stages of design, designers typically restrict the layers used for re-routing the ECO affected nets. Correspondingly, designers would like to run DRC checks on only those specific layers used for the ECO re-routing. The in-design DRC flow allows DRC checks to be run on specific layers. Figure 8 shows the runtime savings of running single-layer DRC checks versus all-layer DRC checks.

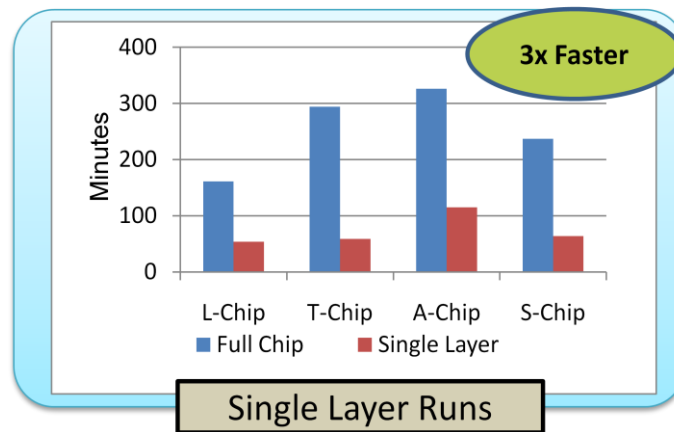


Figure 8: Runtime for Single Layer DRC Checks Versus All-Layer DRC Checks

- ▶ Rule-based DRC checks:  
With the in-design DRC flow, specific DRC rules can be run in the DRC runs instead of running all rules. For example, after metal fill insertion is performed, designer would like to run a DRC check using only the density rule to ensure that the metal fill insertions do not create density violations in the design. The in-design DRC flow allows specific rules to be run in the DRC runs. Figure 9 shows the runtime benefit of running the DRC check with one rule versus running all rules and all layers checks.

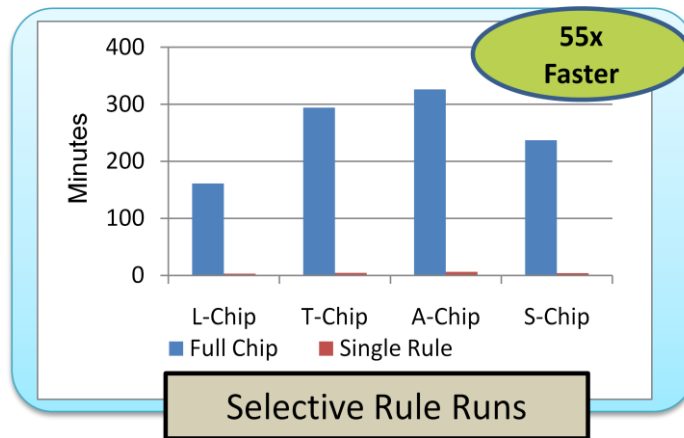


Figure 9: Runtime for a Single Rule DRC Check Versus All layers DRC Checks

▶ Area-based DRC checks

Often DRC fixes are localized to a specific area, and designers would like to quickly check only the fixes for that area. The in-design DRC flow allows DRC checks to be performance on a single region or multiple regions. Figure 10 shows the runtime savings of running the DRC check on a single region versus running the DRC check on the full chip.

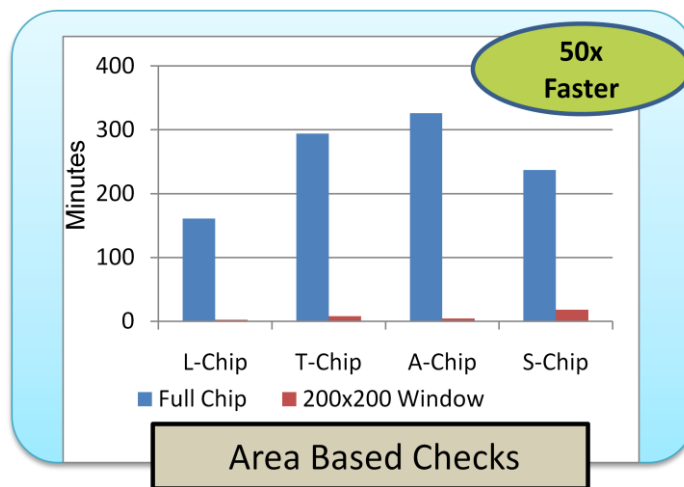


Figure 10: Runtime for a Single Region Check Versus All-Layer DRC Checks

## Summary

By using the Synopsys In-Design Physical Verification flow that links IC Validator within IC Compiler, physical designers can quickly and easily ensure that their designs are free of DRC violations without leaving the IC Compiler environment. By using the same database for implementation, DRC checking and debugging the in-design flow enables designers to fix DRC problems easier within IC Compiler. With rule-based, layer-based, and area-based DRC checking capabilities designers can now perform DRC checks faster and are able to more predictably meet their project schedule. Bu using the distributed processing capability, designers are assured that foundry-certified DRC verification on all layers will be completed in the shortest possible amount of time.

The IC Compiler–IC Validator In-Design DRC link delivers:

- ▶ Sign-off quality DRC checking
- ▶ Faster design closure within the implementation flow
- ▶ Push-button ease of use
- ▶ Distributed processing capability to reduce total turnaround time

To find out more about Synopsys and IC Validator, visit <http://www.synopsys.com/icvalidator>