



SIEMENS

Ingenuity for life

Siemens Digital Industries Software

Achieving optimal performance during physical verification

Design to silicon

Executive summary

Productivity in physical verification is a multi-faceted objective that requires multiple strategies and technologies. IC design companies must constantly evaluate their process flows, data management, and hardware usage, including working closely with EDA suppliers to ensure accurate, complete, and state-of-the-art tool coverage, and incorporating new technical functionality to extend hardware capabilities and improve runtimes. Siemens Digital Industries Software's commitment to innovation and continuous improvement enables design companies to continue producing novel, reliable, and high-quality electronic products with confidence to an ever-expanding marketplace.

John Ferguson

Contents

The runtime challenge for EDA tools	3
Managing physical verification runtimes.....	4
Reduce the computational workload	5
Rule file optimization.....	5
Engine optimization.....	7
Optimize new functionality for performance.....	8
Data management optimization	9
Database format.....	9
Data hierarchy.....	9
Fill data optimization	9
Parallel processing	11
Summary.....	12
References	12

The runtime challenge for EDA tools

In 1965, Gordon E. Moore famously predicted that the functionality of integrated circuits (ICs) would double every two years. As we all know, this prophecy (dubbed "Moore's Law") has actually been exceeded by the industry, with a doubling of the number of transistors available (enabled by a shrink in transistor size) in an integrated circuit (IC) every 18 months. Each new shrink, referred to as a "process node," has been accompanied by the materials, manufacturing, and design improvements needed to keep the tapeout cycle on schedule. Such a rapid engineering cycle has done wonders for the electronics industry by delivering constantly increasing value at the same or lower cost. However, the technology treadmill has become a huge challenge for electronic design automation (EDA) suppliers, who must deliver design and verification tools that can not only scale with the enormous growth in computing needed to handle exponentially increasing design complexity, but at the same time provide an increasing range of functionality to manage the new and emerging layout effects and dependencies that can affect manufacturability, performance, and reliability.

Consider for example, the impact on physical verification. Figure 1 shows the relative increase in geometric data volume as we go from one node to the next. With

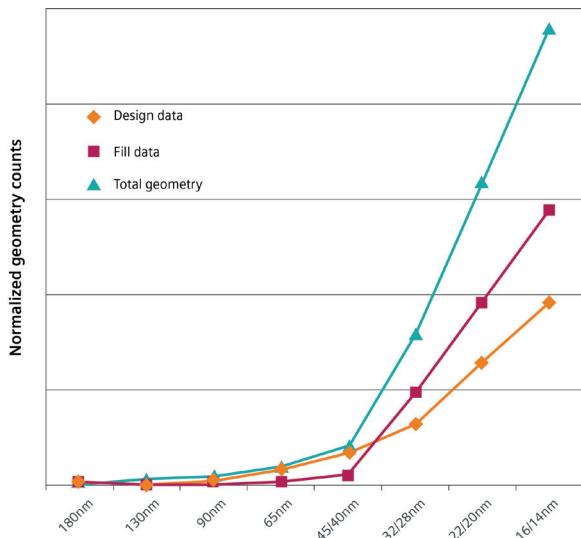


Figure 1: Increase in number of geometries (shapes on lithography masks) per die at historical process nodes.

each new process node, not only are we introducing twice the number of transistors, but the designs also require more metal interconnect wires and interconnect layers to connect these transistors together. In addition, to mitigate inherent limitations in chemical mechanical polishing (CMP), lithography, etch, and rapid thermal annealing (RTA) process steps, additional non-functional metal shapes, often referred to as "dummy fill," are required. While dummy fill has been used for multiple node generations, the 28 nm process node marked a significant change in how dummy fill is handled.

Previously, dummy patterns were placed only in low-density areas on metal interconnect layers to establish a more uniform density across the die. At newer nodes, due to the increase in manufacturing complexities, fill geometries must be placed everywhere there is room to ensure as much uniformity across all layers as possible. This results in a significant increase in the amount of data to be processed during design rule checking (DRC) runs.

At the same time that the amount of design data that must be checked is ballooning, we are also experiencing a significant increase in the amount and complexity of the rules required to tape out designs. Figure 2 shows the increase in the number of rules required to tape out successive process nodes. Due to the difficulty of

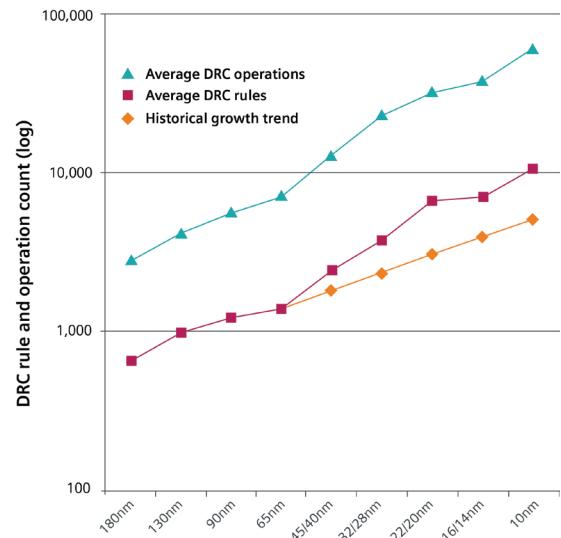


Figure 2: Increase in rules and check operations required for DRC sign-off by process node.

accurately imaging geometries with dimensions much smaller than the wavelength of the light used for exposure in the IC 193i lithography process, the complexity of design rules has grown exponentially from the 40 nm process node forward. Together, these two phenomena significantly impact design rule checking (DRC) runtimes. Not only are there significantly more geometries that must be checked, there are also significantly more rules to check, and the rules are more complex. The result is that a DRC tool must compute many more operations per rule. For example, a chip designed for a 20 nm process requires 3,000 times the total computational effort to tape out than a comparable chip at the

0.25 micron node. Of course, no design team is given a time-to-market window that is 3,000 times longer than in the past, so our challenge as an industry is to enable all this additional computation without significantly increasing the design time.

Typical design cycle planning is based on full-chip DRC runs that complete overnight. The exact definition of "overnight" can vary anywhere from 8 to 12 hours, but the general idea is that if a design team can complete a physical verification job overnight, they then have the day to implement corrections for identified errors, leaving the evening available for the next DRC run.

Managing physical verification runtimes

Managing physical verification runtime is a multi-dimensional challenge. At Siemens, there are three primary strategies we use to keep DRC runtimes within this expected timeframe for each new process node:

- Reduce the computational workload by optimizing the rules and checking operations
- Handle the data more efficiently by adding hierarchy
- Parallelize as much of the computation as possible

Let's look more closely at each of these strategies.

Reduce the computational workload

Two ways to reduce the computational workload are to optimize the design rules, and enhance the DRC tool itself to perform more efficiently.

Rule file optimization

At advanced process nodes, design rules become very complex, often requiring dozens, if not hundreds, of individual check operations to properly describe and implement a rule in DRC. Ideally, the DRC implementation must properly identify real errors without flagging false errors.

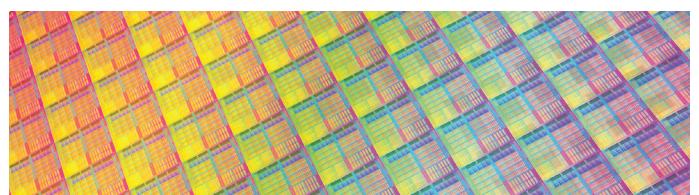
With a robust DRC language, there are often multiple ways to describe the same rule – the challenge is to identify which implementation will be the most computationally efficient. This is no easy task. Since different operations may take different amounts of time to run, the rule implementation with the fewest number of steps does not ensure the fastest possible runtime.

Optimizations to minimize the impact of the explosion of checks and operations can take many forms. Typically, the folks tasked with coding the rules are foundry employees who are experts on the foundry's process requirements, but they may have less expertise on the inner workings of the DRC tools they use to establish the accuracy and coverage of the new rule deck. A rule deck contains tool-specific constructs, normally in an ASCII text format, that describe the algorithms needed to validate geometric adherence to each rule.

In the typical development cycle for a new node, a foundry starts with internal test structures to identify new rule requirements. They work with their internal sign-off DRC tool (which for all major foundries is the Calibre® nmDRC™ tool) to create and test the accuracy of the DRC implementation of the rules. At this stage, the test structures are generally too small and simple to provide accurate expectations of runtime performance for real designs. However, these initial rules become the basis for the design rule manual (DRM), and establish the expected results for the initial rule set applied to test chips. These initial rule files and the DRM are then made available to the very first set of users (typically internal design teams working on intellectual property (IP) designs) as V0.1. Next comes V0.5, which is supplied to the early adopters (external users who have

agreed to design test chips for the target process), as well as internal and external IP and cell library design teams. This version is typically the first time that the sign-off tool supplier is exposed to the rules.

Using best known practices, the DRC tool supplier can provide an initial set of suggestions back to the foundry to help optimize rule performance. At the same time, initial DRC results become available from the early design teams running the rules. While the data in these runs usually represent small designs, this customer design feedback can help to identify some potential performance bottlenecks in rule implementations. At Siemens, we use this data to identify "tall tent poles" in typical DRC runs. This analysis might uncover new ways to improve the performance of existing check operations, or help us devise new operations to meet the checking requirements more efficiently. For example, Calibre engineers may identify certain layout configurations in customer designs that result in slower than expected runtimes. Once these patterns are identified, the Calibre nmDRC tool can be modified to recognize them in a design, and to use specific methods to reduce total processing time. With this information, Siemens then provides additional optimization proposals and suggestions to the foundry.



The suggested optimizations are carefully reviewed by the foundry rule writing teams. They must assess the accuracy of the proposed implementations compared to the initial rule implementation. They must also consider side effects, such as potential differences in how the results are presented to the user. Proposed rule implementations deemed to show considerable improvement without impacting accuracy are included in the next revision to the rule files. Often, as the foundry learns more about yield-limiting design structures, a revised rule file also contains new rules not included in the prior version. During this phase, cooperation between

the foundry rule writers and DRC tool engineers is critical. Figure 3 shows the typical improvements that can be achieved when the tool supplier and foundry rule writers work together to create optimized rule files.

Each new DRC tool feature is tested against the foundry rule decks and regression tests before a tool upgrade can be released to customers. As the development of the DRM and the associated DRC continues, these enhancements are periodically made available to the early adopters with new releases of the DRC tool, providing a constant feedback loop between designers, the DRC tool supplier, and the foundry.

Once all of these initial optimizations and rule changes are in place, other physical verification tool vendors are given access to the DRM. They must code the rule checks on their own, and iterate by comparing their results to the foundry results generated by the foundry's sign-off tool. In the meantime, collaboration continues between the foundry and their signoff tool supplier to further optimize the rule sets. As more exposure is gained through test chips and sample designs of increasing complexity, the process becomes relatively stable and an initial production rule deck is made available. Generally this marks the point at which the introduction of new rules will be sharply reduced. Additional rule file revisions may be issued to correct errors or to provide further optimizations for longer running checks, but there will be relatively few changes that impact sign-off requirements at this point.

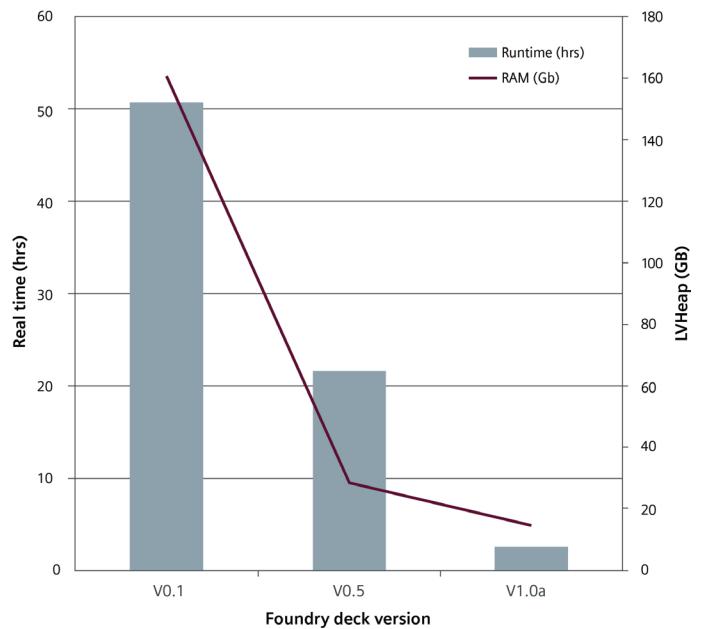


Figure 3: Rule file optimization is a collaborative process between the foundry, who understands the process, and its sign-off DRC tool supplier, who understands the DRC functionality needed to verify design constraints.

Engine optimization

The next way to reduce computational load is to modify the DRC tool itself. As Joe Sawicki, vicepresident and general manager of the Design-to-Silicon division of Siemens says, "Calibre changes every single node".^[1] At each node, the Calibre toolset is rewritten and enhanced to address the new design and runtime challenges that are inherent in new technologies. Siemens also invests a significant amount of R&D resources to constantly ensure Calibre tools run faster, use less memory, and scale more efficiently at all nodes. At the same time, they work very hard to preserve backwards

compatibility to ensure that, for example, a company using Calibre tools for 0.25 micron does not experience any changes in their use or results, but still gains the benefits from performance improvements. Figure 4 shows the typical rate of runtime improvements that an engine optimization can deliver. In this case, runtimes for the Calibre nmDRC tool were measured across several tool releases for over 20 large SoC designs targeting processes at or below 20 nm. The runtimes were averaged to illustrate the expected runtime improvements for typical designs.

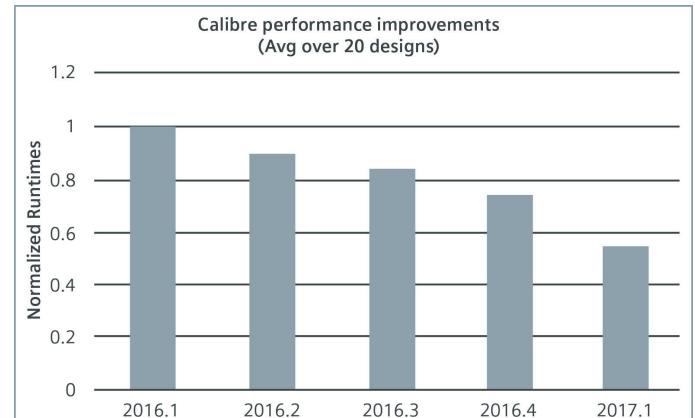
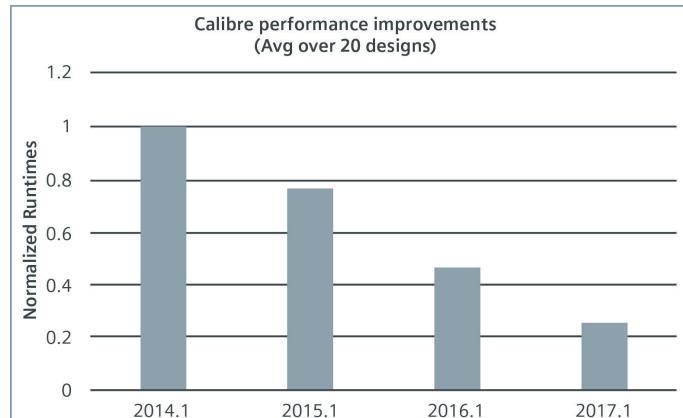


Figure 4: Not only does Calibre performance improve significantly year over year, it continuously improves from release to release.

Optimize new functionality for performance

When new functionality, such as multi-patterning technology, is introduced, it must be optimized to ensure the best possible performance. However, this optimization is not a onetime process. Just as with engine optimization, Siemens continuously evaluates and refines the functionality in the Calibre tool suite to ensure customer success and productivity.

Figure 5 illustrates the runtime improvement achieved over time for multi-patterning processes. When double patterning (DP) was first required, it was implemented as part of Siemens' resolution enhancement technology (RET) toolset used by foundry engineers. Over time, as

more multi-patterning techniques were introduced, the responsibility for multi-patterning decomposition and verification shifted to the design side, so we developed and introduced the Calibre Multi-Patterning tool for designers. We used what we learned from the foundry-side process to ensure that the Calibre Multi-Patterning tool provided reduced runtimes through increased scaling (including native hyperscaling) and increased remote CPU utilization. At the same time, we optimized performance by reducing memory usage and improving the utilization of hierarchy.

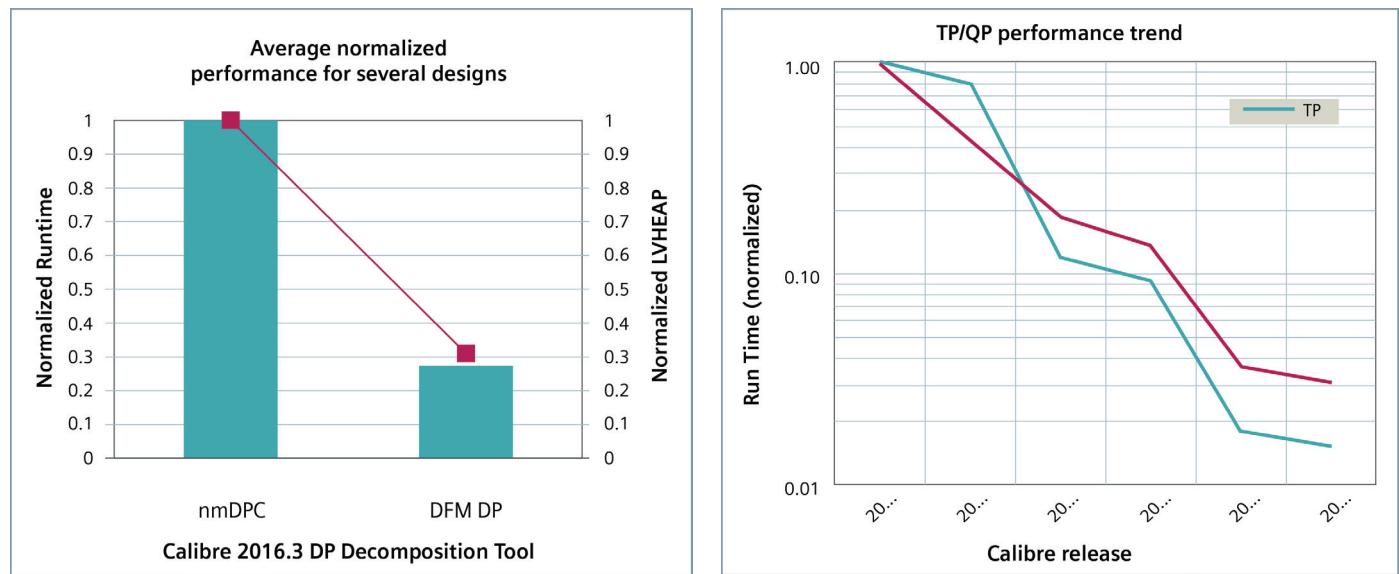


Figure 5: On the left is the improvement gained by re-implementing our DP functionality in a new tool, and on the right is the runtime performance trend for the Calibre multipatterning technology.

Data management optimization

Database format

Data transfer is important in the design phase, in the handoff to manufacturing, and in the manufacturing cycle. Not surprisingly, the choice of database format can have a significant impact on data transfer and file I/O.

In the design space, data is passed from the layout design tool into physical verification. At this step, it is critical that the format fed into the verification tool is consistent with the data format used by the foundry. By eliminating the need for a data format translation at tapeout, a design company eliminates both the chance of introducing errors created by the translation, and the extra time that it would require. In addition, the physical verification tool often generates new layout data, such as fill, so data consistency is also important during the design iteration phase.

Once a design is clean and ready for manufacturing, it is transferred electronically to the foundry. As a result, it is important that the format used to ultimately communicate the layout for sign-off be the same as that used by the foundry. Historically, the GDSII format^[2] was the format of choice. However, GDSII has a number of syntactic limitations that adversely impact the file size. File size dictates how quickly such data can be shared, and the continued growth in design database size results in slower transfer times. Recently, the foundries have begun adopting the OASIS format^[3] as an alternative. Unlike GDSII, OASIS is configured to minimize redundancy when describing geometric patterns, leading to anywhere from a 10X to >100X reduction in file size, which ultimately can significantly reduce internal iteration read and write times, providing a noticeable improvement over the use of GDSII (figure 6).

Data hierarchy

Another way to improve computing efficiency is by optimizing with respect to the layout data to be checked. A great example of optimizing based on data patterns is the use of hierarchy. If a design contains a cell or block that is repeated several times in the design, significant computation time can be saved by checking each unique layout pattern only once, then checking the relationship of each pattern instantiation to its surrounding objects. In the Calibre nmDRC tool, this capability is called hierarchical checking.

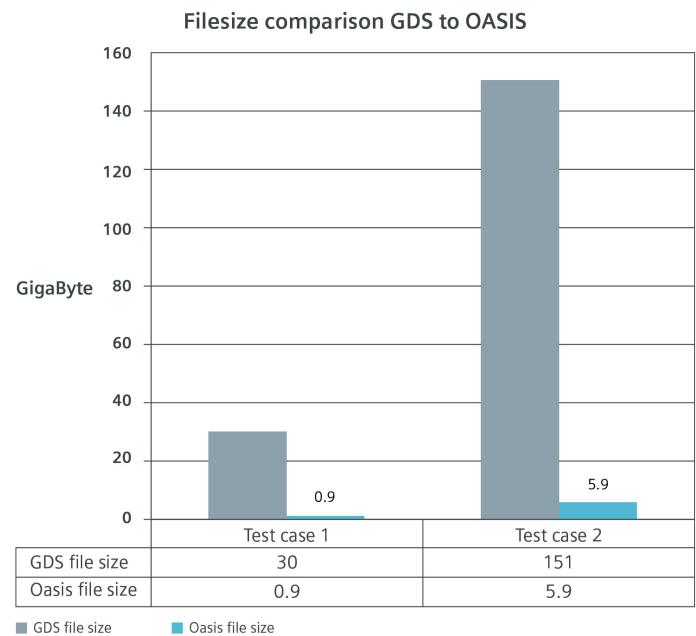


Figure 6: OASIS database size compared to a GDS database for the same design.

In the Calibre nmDRC tool, hierarchy is used strictly for performance optimization. The goal is to produce the exact same results when checking hierarchically as if the checks were done flat (i.e., with no recognition of repetition), but to do so much faster. It's important to note that Calibre DRC hierarchy may not match the inherent design hierarchy for good reasons. For example, there may be inherent hierarchy in a design layout that does not have enough repetition or size to make a significant runtime difference, so Calibre software would not use this hierarchy. On the other hand, there may be pattern repetition in the layout that is useful for runtime optimization, but isn't directly related to a specific cell structure in the design hierarchy. For these reasons, the Calibre Platform generates its own internal view of the hierarchy based on the input layers and knowledge of the operations required for the specific run.

Fill data optimization

A good example of the benefits of hierarchy can be observed in the handling of fill data. At advanced processes, the presence of fill geometries has become

ubiquitous. Fill must be added to essentially any open space in the layout where it can fit without breaking design rules. At the same time, the rules for fill are increasingly complex, the use of fill itself and the specific fill methodology both affect DRC runtime, and fill runtime and file size are concerning to design teams. Figure 7 shows the changes in fill strategy from 65 nm to 20 nm.

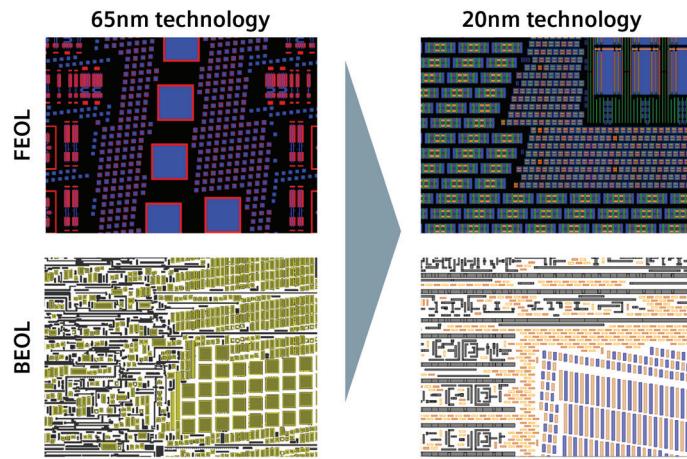


Figure 7: Fill utilization has changed significantly for today's advanced nodes.

Traditionally fill was left until the end of the design cycle, just before tapeout. With this approach, filling is performed at the chip level, which can (and usually does) have a significant detrimental effect on DRC runtimes. A DRC tool must check design rules (like spacing between shapes) for fill, as well as the native design shapes. Unfortunately, the fill shapes are not in the same hierarchy as the design, and typically cannot be captured as repeated patterns because the fill is inserted without regard to the original hierarchy. This limitation means the only way to check millions or billions of fill spacings is in flat mode at the full-chip level, which can be painfully slow.

File size is a critical factor in both runtime and the ability to move fill databases between design and place and route (P&R) environments, and ultimately in delivery of the tapeout database to the foundry for mask generation. The analysis engine must be able to implement multi-layer fill shapes for both the front end of line (FEOL) base layers, and back end of line (BEOL) metal layers. The FEOL rules include fill shapes that must maintain uniform density relationships across multiple layers to reduce the variability created by rapid thermal annealing (RTA) and stress. To further reduce the manufacturing stress and provide structural stability, BEOL fill

incorporates dummy vias, which must also satisfy multi-layer rules. Beyond the complexity of the new fill rules, the number of individual fill elements continues to increase dramatically with each new technology. All of these factors create huge GDS files that take longer to transfer and process. Many design teams have given up on the idea of bringing billions of fill shapes back into their P&R environment because of the impact on their P&R systems.

The Calibre solution to this increase is to raise the level of abstraction by moving from individual polygons to a cell-based fill solution, defining a multi-layer pattern of fill shapes that is repeated in many places across the chip. Fill cells are a natural extension of the multi-level fill constructs that must be maintained, and can be used for both FEOL and BEOL.

Calibre developed its SmartFill hierarchical fill flow that provides huge improvements in post-fill DRC runtimes (figure 8). The "fill as you go" approach enables designers to add fill at the block level first for any large block that is placed more than once. When the chip is assembled, the fill is consistent for each placement of that block. This technique has the added benefit of providing a more accurate assessment of the block's performance, since the effects of fill parasitics are included. Fill is also applied at the top level to cover any areas not addressed by pre-filled blocks. This "fill as you go" technique allows Calibre hierarchical tools to check the fill within the block only once, not for each placement, reducing post-fill runtimes by over a factor of four.

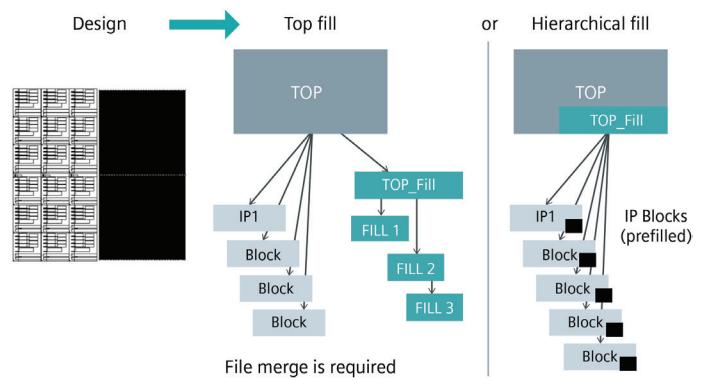


Figure 8: Compared to traditional top-down fill techniques, hierarchical fill provides a number of productivity benefits in the design flow.

The Calibre cell-based approach to fill provides a benefit in both runtime and file size, which helps maintain the project schedule. New cell-based fill techniques with compression introduce hierarchy into the fill. Maintaining that hierarchy in the design flow can have a significant

effect on file size. In one test case, the original fill database in OASIS was 61,442,407 (59MB). After the ECO fill flow, the size of the fill database was 61,497,018 (59MB), an increase of less than 0.1 percent. An “on disk” flow that supports OASIS (and to a lesser degree, GDS) can also help control file size. In the same test case, that hierarchical 59MB OASIS file equated to a 12.9GB flat GDS file. The other benefit of the OASIS format, of course, is that it works with all P&R tools.

Parallel processing

Even with well-optimized rule files and data handling strategies, and the latest DRC tool version, it is still impossible to complete full-chip DRC runs overnight without taking advantage of parallelization. There are two compounding challenges for physical verification: the volume of layout data, and the computational complexity of the rules. Fortunately, each of these aspects can be effectively addressed with parallelization to help ensure fastest total runtimes.

Parallelizing the layout data to be checked in the Calibre toolset takes advantage of the Calibre hierarchical optimization. As one CPU is checking the geometries of a layer associated with one cell, another CPU can be checking the same rule on the same layer, but for a different cell. This maximizes the use of shared memory, largely eliminating the memory contention associated with single-CPU runs.

Parallelizing the DRC checking operations in a rule file allows any two rules to be run on separate CPUs in parallel. Since all input layers for each of the parallelized operations must be available, any operation requiring a derived layer on input must first wait until that layer has been derived. Once the layer has been derived, all operations dependent upon it can proceed.

Both of these approaches can take advantage of multiple CPUs on the same machine, or CPUs distributed over a symmetric multi-processing (SMP) backplane or network. In addition, the Calibre platform can take advantage of virtual cores, or “hyperthreads,” which provide about 20 percent of the processing power of a real CPU (figure 9).

The Calibre platform also takes advantage of memory on distributed machines, which helps reduce the total physical memory required on the master system. This approach is known as hyper-remote scaling, and is the current recommended best practice for full-chip DRC sign-off runs at advanced nodes (figure 10). Using the

hyper-remote option, Calibre DRC jobs with reasonably optimized rule files and hierarchical input data can effectively scale to as many as 2000 CPUs.

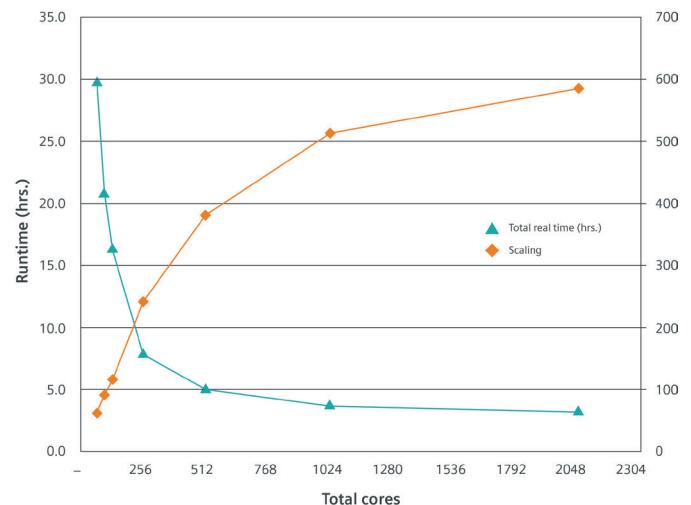


Figure 9: Hyperthreading can significantly decrease runtimes without adding physical hardware.

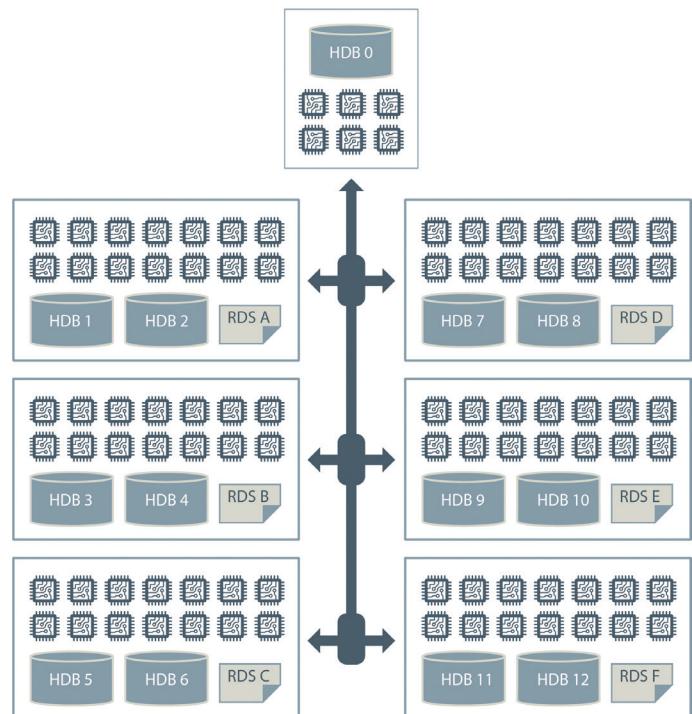


Figure 10: A Calibre distributed hyper-remote configuration enables the most efficient scaling of both data and rule operations for full-chip signoff runs.

Summary

Productivity in physical verification is a multi-faceted objective that requires multiple strategies and technologies. Companies who want to ensure they are maximizing the value of their time and resources must carefully evaluate both their process flows and the tools they use. EDA suppliers use a wide variety of optimizations and enhancements to ensure their tools operate as efficiently as possible, while retaining the accuracy that is crucial to IC success. By constantly evaluating tool performance, working closely with the foundries to ensure accurate and adequate coverage, and incorporating new technical capabilities to extend functionality and improve runtimes, Siemens enables design companies to continue producing innovative, reliable, and high quality electronic products to an ever-expanding marketplace.

References

1. Joe Sawicki, DAC '16 Troublemakers Panel. June 2016. <http://www.deepchip.com/items/0563-01.html>
2. <https://en.wikipedia.org/wiki/GDSII>
3. https://en.wikipedia.org/wiki/Open_Artwork_System_Interchange_Standard

Siemens Digital Industries Software

Headquarters

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe

Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific

Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on [LinkedIn](#), [Twitter](#), [Facebook](#) and [Instagram](#). Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

© 2017 Siemens. A list of relevant Siemens trademarks can be found [here](#). Other trademarks belong to their respective owners.

81401-C2 10/17 N