Datasheet

Design Compiler Graphical

Create a Better Starting Point for Faster Physical Implementation

Overview

Continuing the trend of delivering innovative synthesis technology, Design Compiler® Graphical delivers superior quality of results and streamlines the flow for a faster, more predictable design implementation. Design Compiler Graphical uses advanced optimizations combined with accurate net delay modeling to achieve 5% faster timing post-placement. It extends DC Ultra™ topographical technology to provide physical guidance to IC Compiler, tightening timing and area correlation between synthesis and placement to 5% while speeding-up IC Compiler placement by 1.5X.

RTL designers also gain access to IC Compiler's design planning capabilities from inside the familiar Design Compiler synthesis environment. They can perform what-if exploration to identify and fix floorplan issues early and achieve an optimal floorplan. In addition, Design Compiler Graphical predicts circuit congestion "hot spots," providing designers with visualization of congested circuit regions, and performs specialized synthesis optimizations to minimize congestion in these areas. This enables RTL designers to avoid wire-routing congestion problems that occur during detailed routing.

To accelerate synthesis, Design Compiler Graphical includes scalable multicore infrastructure that delivers significant runtime speed-up on multicore compute servers, yielding 2X faster runtimes on four cores. RTL designers can also analyze and optimize designs across multiple modes and corners concurrently to substantially drive down design development time and cost.

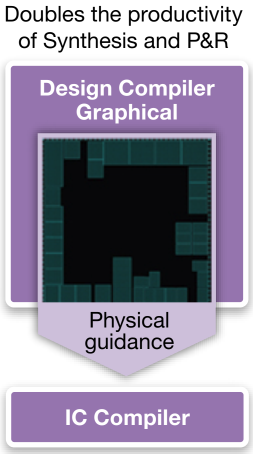

Doubles the productivity of Synthesis and P&R
Design Compiler Graphical
Physical guidance
IC Compiler

Figure 1: Design Compiler Graphical

Key Benefits

- Advanced optimizations deliver 10% faster timing QoR
- Physical guidance to IC Compiler tightens correlation of timing, area and power to within 5% and speeds placement by 1.5X
- Accurate pre- and post-synthesis congestion prediction and congestion-driven optimization eases routing
- Gate-to-gate optimization for smaller area on new or legacy designs while maintaining timing Quality of Results (QoR)

- Cross-probing between RTL, and design views such as schematic, timing reports and physical views for faster debugging
- Early physical visualization and debugging identifies layout issues prior to physical implementation
- Floorplan exploration for faster design convergence to an optimal floorplan
- 2X faster runtime on quad-core compute servers
- Concurrent multi-corner, multi-mode (MCMM) synthesis

Advanced Optimizations Deliver Superior Timing Quality of Results

Physical effects such as coupling capacitance between parallel interconnect wires and resistance values for metal routing layers can have significant impact on design delay and need to be accounted for during synthesis. Design Compiler Graphical uses technology shared with IC Compiler to take these physical effects into account to achieve superior quality of results. The innovative optimizations of Design Compiler Graphical work in conjunction with place and route technology to deliver 5% faster timing on high performance designs. Advanced techniques such as layer aware buffering insertion and timing based incremental placement optimization drives improvements in quality of results.

Physical Guidance to IC Compiler

With designs becoming more complex at smaller geometries, RTL designers require even tighter correlation between synthesis and layout results.

Design Compiler Graphical extends topographical technology in DC Ultra to create physical guidance for IC compiler, streamlining the implementation flow and accelerating IC Compiler placement runtimes by 1.5X. Physical optimizations during synthesis create a better starting for physical implementation and accurately model small-geometry effects, bringing synthesis timing and area results to within 5% of layout.

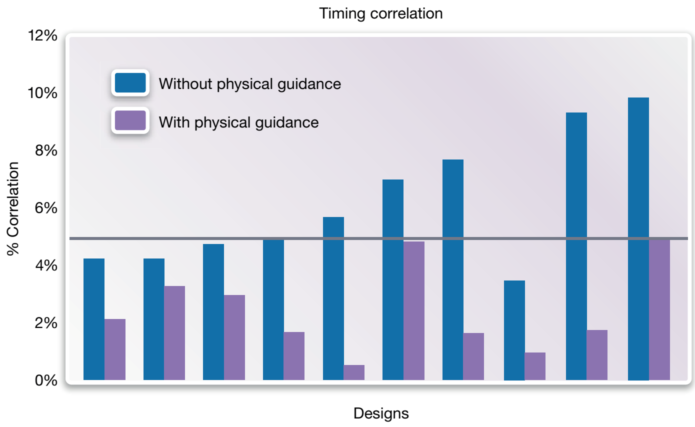


Figure 2: Timing correlation

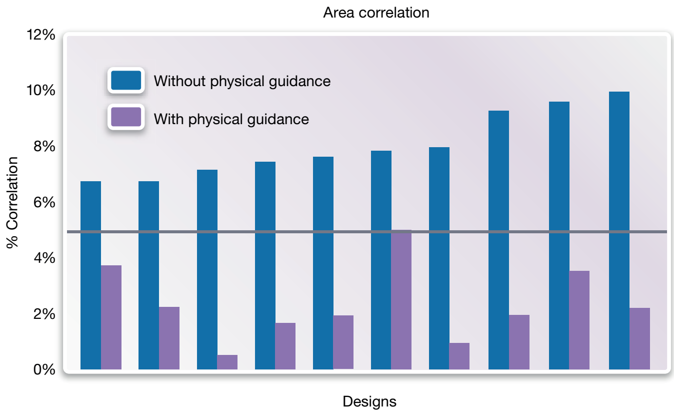


Figure 3: Area correlation

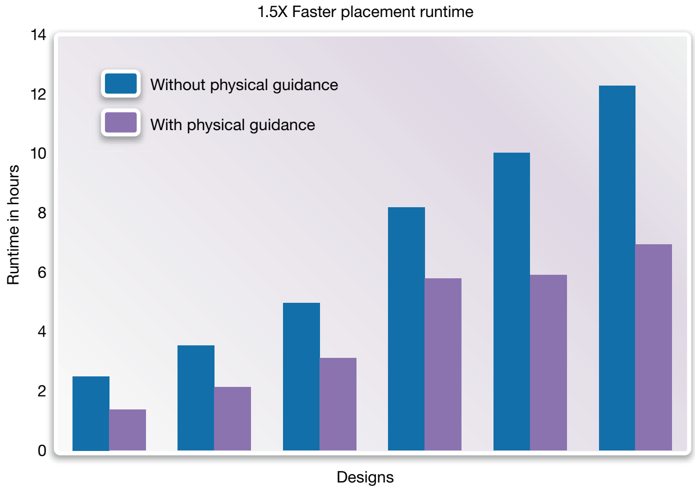


Figure 4: IC Compiler placement runtime

Figures 2 and 3 illustrate improvements in timing and area correlation, respectively, across multiple designs using physical guidance. On the X-axis are the designs and on the Y-axis is the delta between synthesis and layout results. The blue bars (on the left) show the delta between synthesis and layout without passing physical guidance. The purple bars show the delta for the same designs with physical guidance technology. As shown in these figures, results are consistently within 5% when physical guidance is passed from Design Compiler Graphical to IC Compiler. Figure 4 shows placement runtime improvements using physical guidance technology. On the X-axis are the designs and on the Y-axis are the runtimes in hours. The blue bars represent IC Compiler placement runtime without physical guidance and the purple bars represent IC Compiler placement runtime with physical guidance. As illustrated by the figure IC Compiler placement runtime is much faster when physical guidance is passed from Design Compiler Graphical, averaging 1.5X faster.

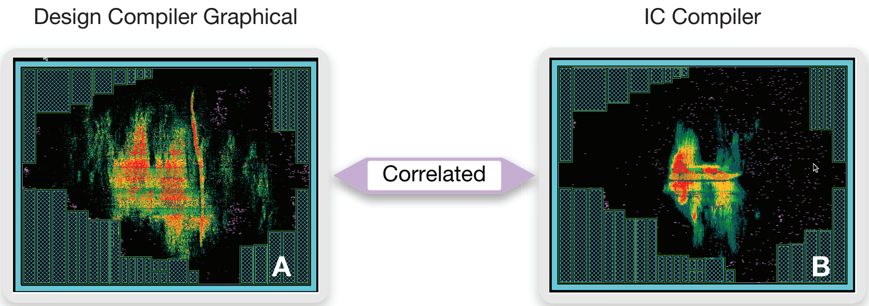


Figure 5: Design Compiler Graphical results

Accurate Pre- and Post-Synthesis Congestion Prediction and Congestion-Driven Optimization

Routing congestion occurs when the resources (tracks) needed to route a design exceed the available resources. With more functionality crowded onto a chip, congestion makes it difficult to route designs. During place-and-route designers deploy various techniques to alleviate congestion. These techniques can include changes to the floorplan such as port or macro locations, changing target gate utilization, adding placement blockages, etc. Making such changes during place and route is time consuming and can lead to schedule delays. Furthermore, these techniques may not work and the designer may be required to iterate back to the RTL and recode the RTL source to remove congestion-causing design characteristics. These options are not optimal and can lead to missed schedules, missed design goals and result in added costs.

Design Compiler Graphical identifies and reports RTL structures that have the potential to cause routing congestion problems later in the flow and cross-probe them back to the RTL source where they can be addressed before implementation of the design.

Design Compiler Graphical includes Synopsys' virtual global-routing technology that enables designers to predict wire-routing congestion during RTL synthesis. This technology allows designers to identify and fix design issues to reduce routing congestion, eliminating costly iterations between synthesis and physical implementation to achieve their design goals and speed up place and route.

Design Compiler Graphical provides an automated way to optimize the RTL to reduce routing congestion. It performs specialized optimizations to generate a routing-friendly netlist topology that minimizes highly-congested structures and wire crossings in congested areas. By intelligently choosing netlist structures that are easier to route, Design Compiler Graphical can generate a netlist that is a better starting point for physical implementation, leading to faster place and route.

Figure 5A shows a congestion map as predicted by Design Compiler Graphical. The color distribution of the map indicates the relative routability of the design, with large concentrations of white and red indicating areas of high congestion and blue representing the least-congested areas. Figure 5B shows the congestion map in IC Compiler after optimizing the design in place and route to reduce congestion. It is clear that Design Compiler Graphical is able to identify the design's highly congested areas during RTL synthesis, thus providing designers with valuable information on the design's ability to route during place and route.

Figure 6A shows the congestion characteristics of the same design highlighted in Figure 5A. Figure 6B shows this design after optimizing it to reduce congestion using Design Compiler Graphical. It clearly shows that the congestion optimization technology has significantly reduced wire-routing congestion in synthesis, resulting in a design with minimal to no congestion as shown in Figure 6C after placement in IC Compiler. Design Compiler Graphical has automatically optimized this design to minimize wire-routing congestion by taking into consideration the congestion characteristics of the synthesized cells.

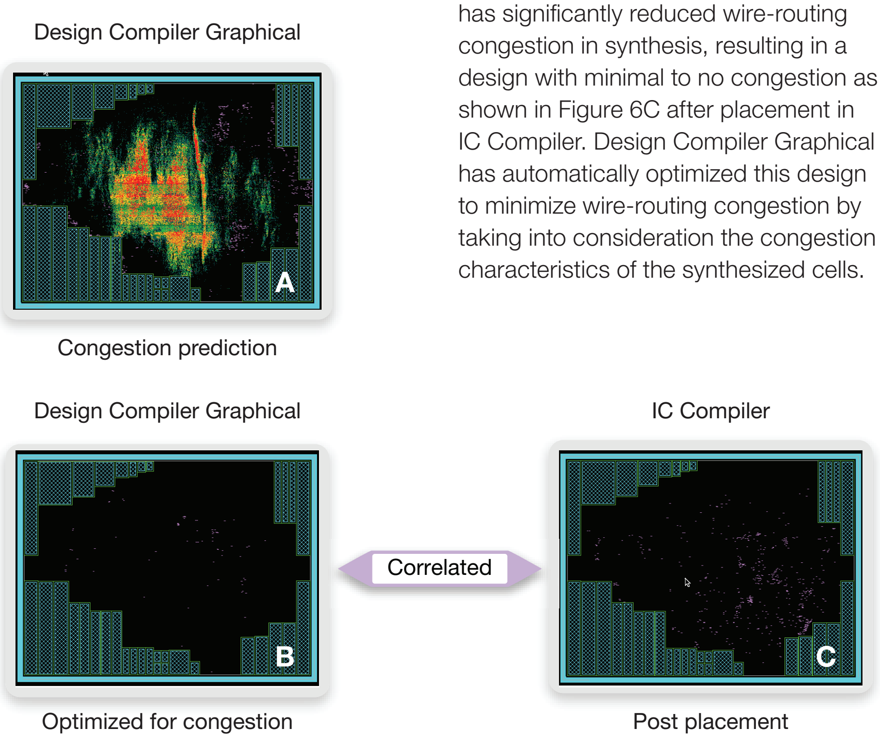
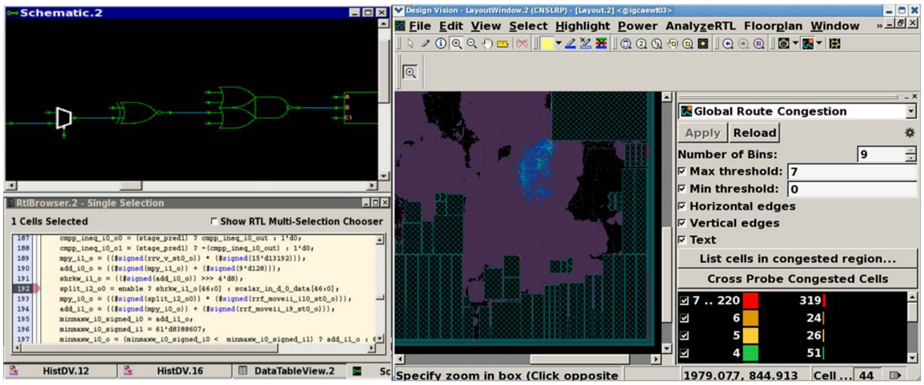


Figure 6: Design Compiler Graphical and IC Compiler congestion maps

Area Reduction Technologies

Design Compiler Graphical provides optimization technologies that monotonically reduce gate-to-gate area by an average of 10% while maintaining timing Quality of Results (QoR). These advanced optimizations operate on both new and legacy design netlists, with or without physical information and at all process nodes. Area reduction is achieved without re-synthesis and without affecting timing results for maximum productivity.


Schematic.2
A
B
CI
RtlBrowser.2 - Single Selection
HistDV.12
HistDV.16
DataTableView.2
Sc
Design Vision - LayoutWindow.2 (CNSLRP) - [Layout.2] <@igcaewt03>
File Edit View Select Highlight Power AnalyzeRTL Floorplan Window
Apply
Reload
List cells in congested region...
Cross Probe Congested Cells
Specify zoom in box (Click opposite
Cell ...

1 Cells Selected

□ Show RTL Multi-Selection Chooser

```
cmpp_ineq_i0_o0 = (stage_pred1) ? cmpp_ineq_i0_out : 1'd0;
cmpp_ineq_i0_o1 = (stage_pred1) ? ~(cmpp_ineq_i0_out) : 1'd0;
mpy_i1_o = (($signed(rrv_v_st0_o)) * ($signed(15'd13192)));
add_i0_o = (($signed(mpy_i1_o)) + ($signed(9'd128)));
shrkw_i1_o = (($signed(add_i0_o)) >>> 4'd8);
split_i2_o0 = enable ? shrkw_i1_o[46:0] : scalar_in_d_0_data[46:0];
mpy_i0_o = (($signed(split_i2_o0)) * ($signed(rrf_moveii_i10_st0_o)));
add_i1_o = (($signed(mpy_i0_o)) + ($signed(rrf_moveii_i9_st0_o)));
minmaxw_i0_signed_i0 = add_i1_o;
minmaxw_i0_signed_i1 = 61'd8388607;
minmaxw_i0_o = (minmaxw_i0_signed_i0 <  minmaxw_i0_signed_i1) ? add_i1_o : 6
```

Global Route Congestion

Number of Bins: 9

☑ Max threshold: 7

☑ Min threshold: 0

☑ Horizontal edges

☑ Vertical edges

☑ Text

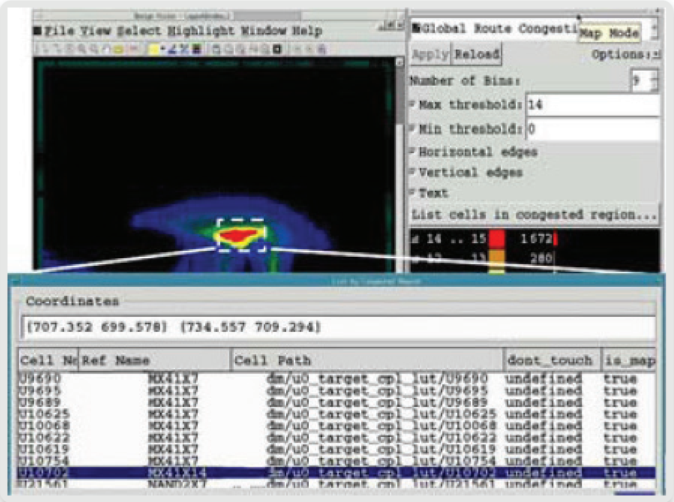
☑	7 .. 220	🟥	319
☑	6	🟧	24
☑	5	🟨	26
☑	4	🟩	51

1979.077, 844.913 | Cell ... 44

Figure 7: Cross-probing between RTL, schematic and congestion views

Cross-Probing

Cross-probing between the RTL source code and other design views such as schematic, timing reports and physical implementation provide designers with the ability to quickly detect potential design issues and fix them at the source. Early visibility into potential design issues using multiple views accelerates debug and achieving design goals.


File View Select Highlight Window Help
Map Mode
Apply
Reload
Options:
List cells in congested region...

Global Route Congesti

Number of Bins: 9

Max threshold: 14

Min threshold: 0

Horizontal edges

Vertical edges

Text

☑	14 .. 15	🟥	1672
	13	🟧	280

Coordinates

[707.352 699.578] [734.557 709.294]

Cell Na	Ref Name	Cell Path	dont_touch	is_map
U9690	MX41X7	dm/u0_target_cpl_lut/U9690	undefined	true
U9695	MX41X7	dm/u0_target_cpl_lut/U9695	undefined	true
U9689	MX41X7	dm/u0_target_cpl_lut/U9689	undefined	true
U10625	MX41X7	dm/u0_target_cpl_lut/U10625	undefined	true
U10068	MX41X7	dm/u0_target_cpl_lut/U10068	undefined	true
U10622	MX41X7	dm/u0_target_cpl_lut/U10622	undefined	true
U10619	MX41X7	dm/u0_target_cpl_lut/U10619	undefined	true
U10754	MX41X7	dm/u0_target_cpl_lut/U10754	undefined	true
U10702	MX41X14	dm/u0_target_cpl_lut/U10702	undefined	true
U21561	NAND2X7	dm/u0_target_cpl_lut/U21561	undefined	true

Figure 8: Interactive analysis of congestion map in Design Compiler Graphical

Early Physical Visualization

Design Compiler Graphical includes a physical viewer that allows RTL designers to view layout congestion in their design during synthesis, as shown in Figure 8. Routing congestion that is related to floorplan, such as macro placement or port location, cannot be automatically optimized in synthesis. These congestion issues can only be resolved by changing the floorplan. Using Design Compiler Graphical's physical viewer, designers can identify floorplan issues such as sub-optimal macro or port locations that are causing timing violations or congestion "hot-spots" and use the floorplan exploration capability described later to take corrective measures to alleviate congestion problems before place and route.

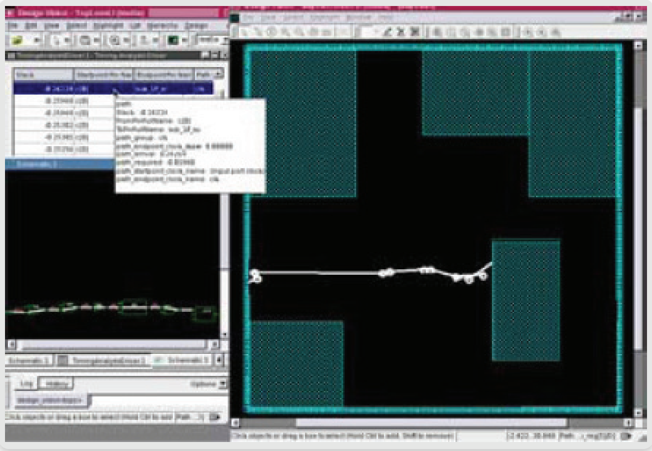


Figure 9: Design Compiler Graphical physical view

These interactive visualization capabilities also include the ability to cross-highlight suspect physical cells in the congestion map to the netlist, as shown in Figure 9. This allows the designer to easily isolate problem timing paths and make necessary changes during RTL synthesis.

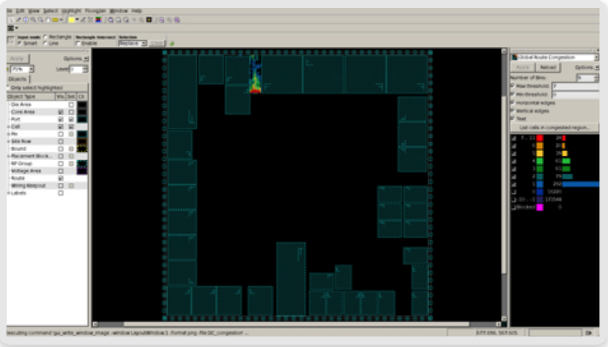


Figure 10: Routing congestion identified in Design Compiler

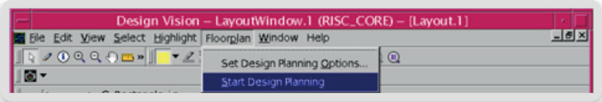

Design Vision – LayoutWindow.1 (RISC_CORE) – [Layout.1]
File Edit View Select Highlight Floorplan Window Help
Set Design Planning Options...
Start Design Planning

Figure 11: Accessing IC Compiler design planning within synthesis

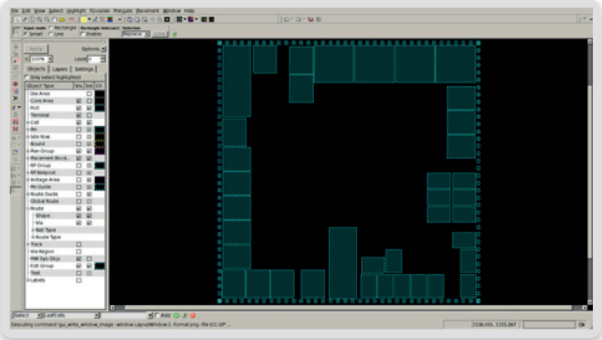


Figure 12: Floorplan editing to address routing congestion

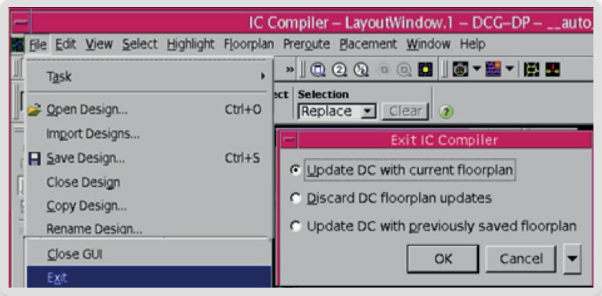

IC Compiler – LayoutWindow.1 – DCG-DP – __auto
File Edit View Select Highlight Floorplan Preroute Placement Window Help
Task
Open Design... Ctrl+O
Import Designs...
Save Design... Ctrl+S
Close Design
Copy Design...
Rename Design...
Close GUI
Exit
Selection
Replace
Clear
Exit IC Compiler
● Update DC with current floorplan
○ Discard DC floorplan updates
○ Update DC with previously saved floorplan
OK
Cancel

Figure 13: Saving Floorplan updates

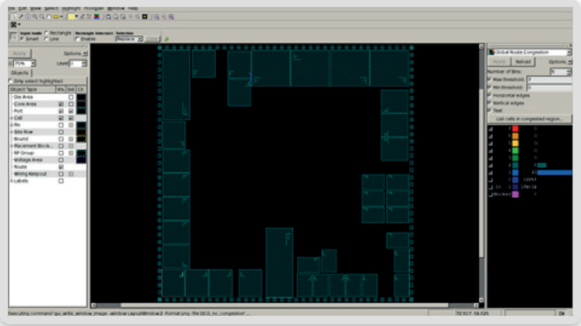


Figure 14: Congestion eliminated

Floorplan Exploration for Faster Design Convergence

Traditionally, if changes to design floorplans were needed, RTL designers had to ask their counterparts on physical design teams to adjust the floorplan, resulting in iterations between the teams. With immense time-to -market pressures, designers need a solution to reduce these iterations. Design Compiler Graphical provides RTL designers access to IC Compiler design planning capabilities from within the familiar synthesis environment.

After detecting design issues, such as routing congestion or timing violations due to floorplan characteristics, RTL designers can now amend the floorplan and re-synthesize the design with an updated floorplan without ever leaving the synthesis environment. The IC Compiler design planning menus have been simplified to give RTL designers ease of use for simple floorplan modifications. An option is available for expert users to utilize the full, advanced floorplanning capabilities. The Design Compiler Graphical and IC Compiler design planning link is transparent to users hence no setup or data transfer is required. Once the designer has created an optimal floorplan, he can re-synthesize the design with the new floorplan and also save it to be used for physical implementation downstream.

Figure 10 shows an example of design layout where congestion hot spots occurred due to a very narrow channel between macros as shown in Design Compiler Graphical's layout viewer. A click on the "Start Design Planning" menu option in Design Compiler (see Figure 11) opens a new IC Compiler design planning window with the design floorplan loaded for editing. With very few maneuvers RTL designers can move the macro to eliminate this narrow channel as shown in Figure 12. Once the floorplan edits are made, designers can save the floorplan as shown in Figure 13 and re-synthesize the design with the updated floorplan. As shown by the congestion map in Figure 14, with the updated floorplan, routing congestion has been eliminated and the design is ready for physical implementation.

Design Compiler Graphical helps RTL designers perform a what-if analysis of floorplan quickly and efficiently so that they can be ensured that the design will meet its targets during physical implementation without requiring iterations.

Multicore Infrastructure Delivering 2X Faster Runtime on 4 Cores

The advent of multicore processors in computer platforms has boosted the processing power available to designers. Design Compiler Graphical has a scalable infrastructure to take advantage of multicore compute servers. Using an optimized scheme of distributed and multithreaded parallelization, Design Compiler Graphical delivers a 2X improvement in runtime on quad-core platforms. The new infrastructure delivers runtime benefits without varying the quality of results.

Multi-Mode, Multi-Corner (MCMM) Synthesis

MCMM optimization is useful for designs that can operate in many modes such as test mode, low-power active mode, stand-by mode and so on. Used along with specification of power intent in the Unified Power Format (UPF), it serves as the key enabling technology for performing dynamic voltage and frequency scaling (DVFS) design realization.

Implementation of a design must take into account all of the different operational modes that a design can have. For example, a single block can operate in fully functional, low-power active, stand-by and/or completely shut-down modes. Without multi-mode optimization, the typical flow is to perform timing optimization sequentially for the different modes targeting various corners that represent the different operating conditions and constraints. Design Compiler Graphical's multi-mode concurrent optimization reduces the number of iterations providing faster time to results for multi-mode designs.

One of the primary benefits of performing multi-corner optimization is to get optimal leakage results. Prior to the availability of Design Compiler Graphical MCMM optimization, designers typically optimized their designs either by performing leakage power optimization on the same corner as timing optimization, or by performing leakage and timing optimization sequentially using different corners for worst case timing and worst case leakage.

The graph below (Figure 16) shows the effects of the worst case timing and leakage corner of a 65nm low power process where the higher voltage corner (1.32V) reflects the best performance but leakage is at its worst. Conversely, the low voltage corner (0.9V) reflects the best leakage; however, performance (or speed) of the design will be at its slowest at this environmental condition. Both of these corners must be accounted for simultaneously in order to achieve the optimal leakage and performance objectives for the design.

In addition, with the complexities of sub-micron processes, the worst case leakage corner is now changing from the typical hot temperature (125C) at VDD+10% case to a temperature inversion corner usually occurring at a colder temperature environment. Design Compiler Graphical's MCMM optimization can take all of these different process corners into account to provide the best leakage results while trying to minimize impact to performance. The ability to concurrently optimize for multiple modes and corners reduces iterations between the front and back-end implementation, allowing designers to achieve rapid design convergence.

Easy to Adopt

Design Compiler Graphical is designed for seamless integration into the current RTL synthesis use model. It uses the same setup as DC Ultra. Similarly, it is designed for RTL designers, does not require deep physical design expertise, and improves productivity by enabling

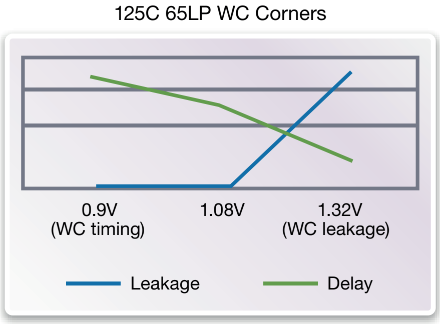


Figure 15: Worst case leakage and timing corners

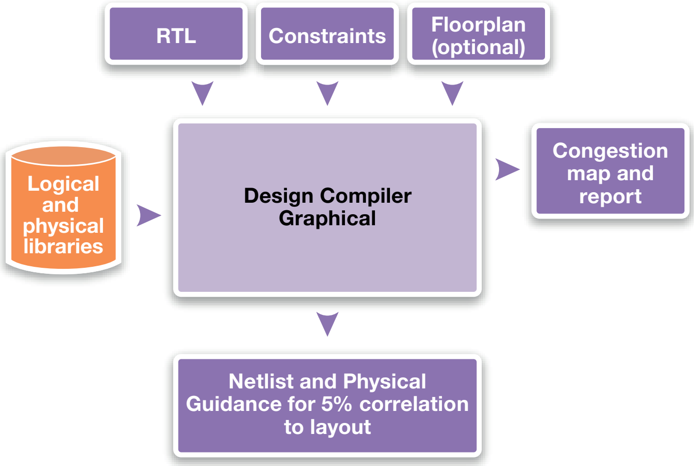

RTL
Constraints
Floorplan (optional)
Logical and physical libraries
Design Compiler Graphical
Congestion map and report
Netlist and Physical Guidance for 5% correlation to layout

Figure 16: Design Compiler Graphical inputs and outputs

more informed decisions during RTL synthesis with early visibility into what the design characteristics will be during place and route. The inputs to Design Compiler Graphical, which are the same as DC Ultra with topographical technology, are listed below and also shown in Figure 16:

- Design RTL
- Logical Library (db)
- Physical Library (Milkyway™)
- Design constraints (SDC)
- Optional physical constraints (Floorplan)

The output is a netlist optimized for timing, area, test, power and congestion with accurately-predicted layout results, ready for physical implementation hand-off. In addition, physical guidance for Synopsys place-and-route solution IC Compiler can be created, tightening timing and area correlation to 5 percent while speeding up IC Compiler's placement phase by 1.5X.

Conclusion

Design Compiler Graphical boosts RTL designers' productivity and offers superior quality of results. It provides the ability to accurately predict, visualize and alleviate routing congestion, creating an easy-to-route netlist substantially reducing iterations between synthesis and physical implementation. Optimization technologies that reduce gate-to-gate area by an average of 10% while maintaining timing Quality of Results (QoR) operate on both new and legacy design netlists. Design Compiler Graphical enables RTL designers to perform floorplan exploration from within the synthesis environment and converge on the optimal floorplan faster. In addition, it accurately models the effects of smaller geometries, deploys advanced physical optimizations and generates physical guidance for IC Compiler place-and route solution to further tighten the correlation and accelerate physical implementation. Design Compiler Graphical is 2X faster on multicore compute servers while ensuring zero deviation of synthesis results. It also optimizes designs across multiple modes and corners concurrently to achieve optimal results faster.

Availability

Design Compiler Graphical is available as an add-on to DC Ultra.

For more information about Synopsys products, support services or training, visit us on the web at: www.synopsys.com, contact your local sales representative or call 650.584.5000.

01/14.AP.CS3872.