DC Explorer

Early RTL Exploration Accelerates Design Schedules

Overview

DC Explorer enables early RTL exploration leading to a better starting point for RTL synthesis and accelerating design implementation. With tolerance to incomplete design data, faster runtimes and timing and area correlation to DC Ultra™, it provides early visibility into implementation results. DC Explorer enables designers to efficiently perform what-if analyses of various design configurations early in the design cycle to speed the development of high quality RTL and constraints and drive a faster, more convergent design flow. It also generates an early netlist that can be used to begin physical exploration in IC Compiler™ II. With push-button access to IC Compiler design planning from inside the RTL exploration environment, DC Explorer lets designers easily create and modify floorplans very early in the design cycle.

To meet their aggressive schedules for today's "Gigascale" designs, engineers need an RTL exploration solution that enables them to quickly and efficiently perform what-if analyses even before the RTL and constraints are complete so they can improve the design data and create a better starting point for the implementation flow. DC Explorer (Figure 1) provides designers with the RTL exploration capabilities they need.

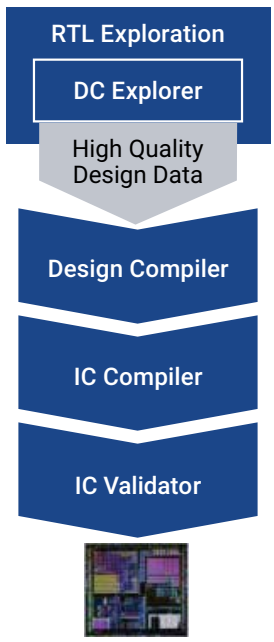


Figure 1: DC Explorer accelerates development of high quality RTL and constraints leading to a faster, more convergent design flow

DC Explorer improves designer productivity and reduces the risk of unexpected project delays downstream in the flow by enabling RTL exploration in the early stages of the design cycle. The ability to efficiently identify potential design issues and make improvements prior to implementation enables faster development of high quality RTL and constraints and the creation of a better starting point for RTL synthesis.

DC Explorer tolerates incomplete design data, rapidly generating a netlist and HTML-based timing reports that engineers can use to refine their design constraints and determine if the current RTL will likely meet their design goals. Designers can evaluate design tradeoffs and perform what-if analyses of various design configurations. This leads to a better RTL and constraints input to synthesis, enabling a highly convergent implementation flow and reducing the risk of unexpected design iterations.

When the design data is incomplete, DC Explorer reports design mismatches to help designers resolve data inconsistencies prior to implementation. To further improve productivity, the netlist generated by DC Explorer can be used to begin early design exploration and block feasibility in IC Compiler II while the design constraints are still being developed. Push-button access to IC Compiler II design planning from within the RTL exploration environment leads to faster development of a floorplan for early physical exploration.

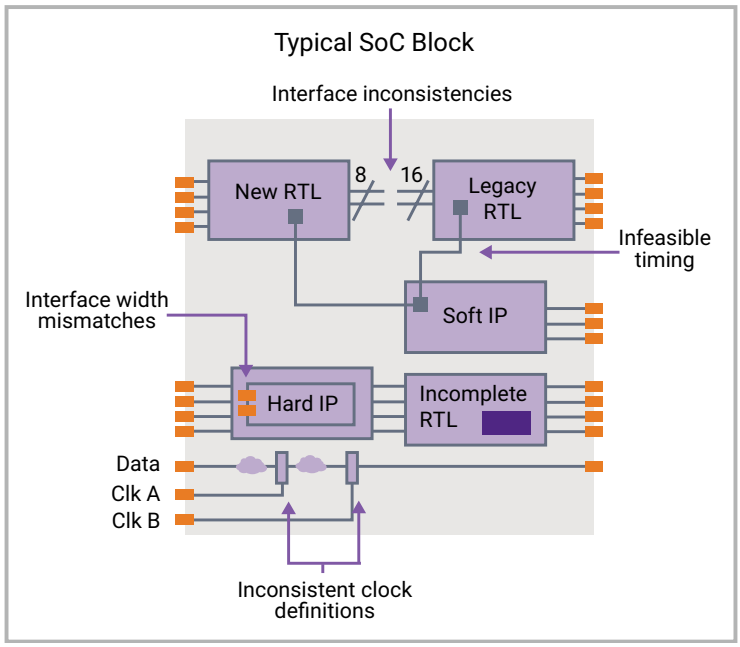


Figure 2: RTL and constraints are incomplete in the early stages of design development

Key Benefits

- Early RTL exploration creates a better starting point for RTL synthesis
- Tolerance of incomplete design data for faster development of high quality RTL and constraints
- Pre and post-synthesis routing congestion analysis and reporting for early RTL feedback
- Histogram showing levels of logic for RTL analysis and potential timing improvement
- Cross-probing between RTL, schematic, timing reports, congestion and physical views for fast debug
- Faster runtime than RTL synthesis for efficient what-if analyses
- Timing and area correlation with DC Ultra for early visibility into implementation results
- Optionally reads in physical constraints for tighter correlation with Design Compiler® Graphical
- Push-button access to IC Compiler II design planning for faster floorplan development and exploration
- Script-compatible with DC Ultra for easy deployment into existing flows
- Support for UPF for early power intent development
- Multicore compute platform support delivers additional 2X runtime speedup on 4 cores

Early RTL Exploration for Faster Design Convergence

Design data for today's large and complex ICs often comes from multiple sources, and at the early development stage the design blocks and IP have varying levels of consistency and completeness, as shown in Figure 2. These issues can take time to fix, delaying the start of the RTL synthesis process and exposing designers to higher risk of design iterations downstream in the flow.

Design Mismatch Scenarios
Missing cells in logical library
Missing pins in logical library cells
Missing pins in RTL modules
Missing RTL modules
Case mismatch between pin names
Bus pin naming style
Bus vs. bit-blasted
Missing pins in physical library cell
Missing physical library cell
Pin direction mismatch
Port width mismatch
Unconnected interface pin
Pin name mismatches

Figure 3: Examples of design mismatches DC Explorer tolerates

Tolerance of Incomplete Design Data

DC Explorer sidesteps this bottleneck by providing early RTL exploration that is tolerant of incomplete and inconsistent design data. As shown in Figure 3, DC Explorer is tolerant of a variety of design mismatches. One of these, "Missing pins in RTL modules," is illustrated in Figure 4, which shows an instantiation of an RTL block with I/O not defined in the block itself.

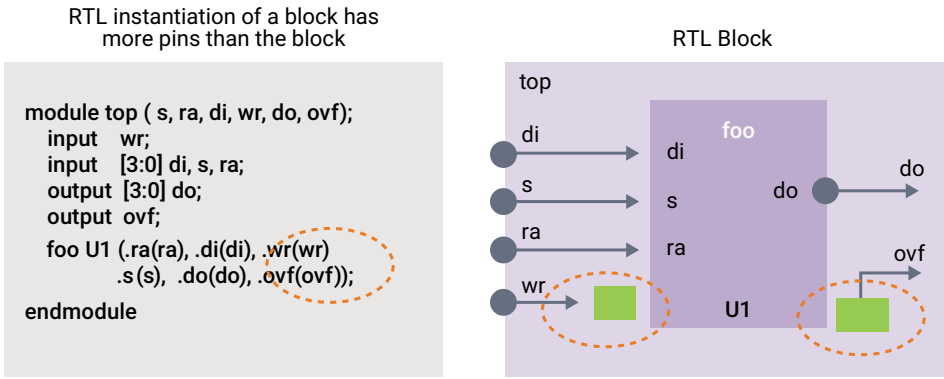


Figure 4: Example of an RTL instantiation that has more pins than the block it references

DC Explorer identifies issues like this, generating a summary list and detailed reports of the design mismatches. These comprehensive reports provide designers the opportunity to make corrections prior to RTL synthesis, speeding the creation of a "clean" RTL. Despite incomplete or missing data, DC Explorer runs through the entire design and script without stopping, preserving the design intent by resolving mismatches and anchoring dangling logic, and generating a netlist that can be used for physical exploration even before the RTL is complete.

Pre and Post-Synthesis Congestion Analysis

DC Explorer provides analysis and reporting on RTL structures that are likely to cause routing congestion issues later on in the design flow. By having access to this type of information sooner in the design cycle, designers can modify their RTL before implementation. DC Explorer also detects and reports congestion post-synthesis. Many of these RTL related congestion issues can be alleviated using Design Compiler Graphical that includes innovative optimization for congestion reduction.

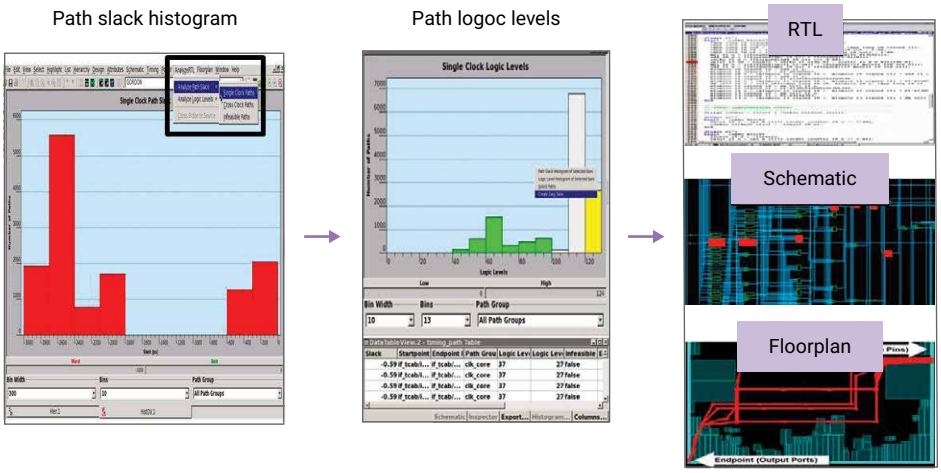


Figure 5: Levels of logic histogram for high quality design data

RTL Analysis and Cross-Probing

RTL analysis capabilities include a histogram showing user-configurable, color-coded levels of logic based on library analysis enabling early feedback to determine the likelihood of the design meeting timing. (See figure 5.) Early analysis of potential timing issues by viewing levels of logic provides insight into the type of changes that will be needed to meet timing such as RTL changes, floorplan modifications or additional constraints.

Cross-probing between the RTL source code and other design views such as schematic, timing reports and histograms, congestion, and physical implementation views for early visibility into potential timing and congestion issues and accelerate the creation of high quality RTL and constraints. (See figure 6.)

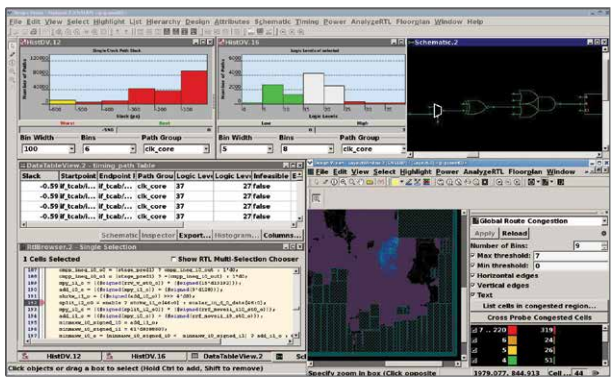


Figure 6: Cross-probing across all aspects of timing paths

Early Netlist and Push-Button Access to Floorplanning for Physical Design Exploration

Since DC Explorer generates a usable netlist, either a DDC or ASCII (Verilog), prior to RTL completion, designers can begin to explore floorplan options, create initial floorplan partitions and assess the physical feasibility of the design much earlier in the flow. With instant access to IC Compiler II design planning from within the familiar Design Vision layout viewer, RTL designers can create an early floorplan, modify it, and read in the physical constraints without ever leaving the RTL exploration environment (Figures 7-8).

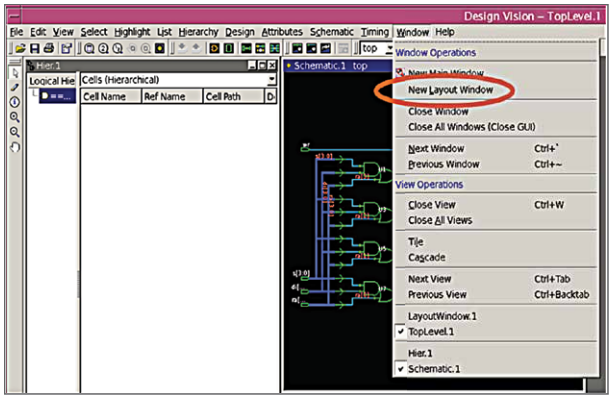


Figure 7: Accessing IC Compiler II design planning from within DC Explorer

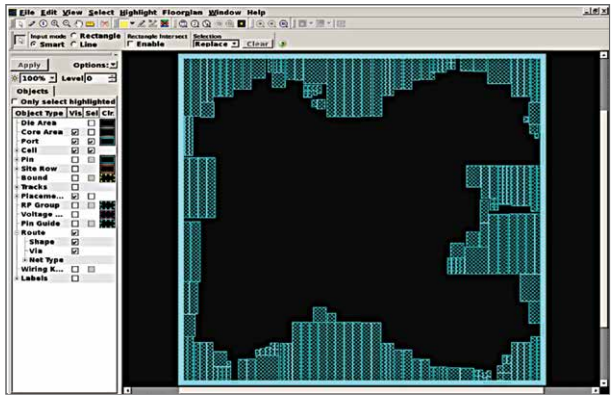


Figure 8: Modifications to the floorplan are read in to DC Explorer to tighten timing correlation with DC Ultra (Topographical)

Using this same floorplan for synthesis ensures tight timing correlation between DC Explorer and DC Ultra (Topographical).

Efficient what-if analyses

DC Explorer performs optimizations for timing, area, leakage power (%LVT) and dynamic power (clock gating) faster than RTL synthesis. The very fast runtimes for large designs (Figure 9) enable efficient what-if analyses to assess the RTL and constraints and determine if the design will likely meet its timing, area and power goals. Based on the outcomes, the designer can then make changes as needed, such as adding pipeline stages to adjust the latency and improve timing, prior to starting implementation of the design.

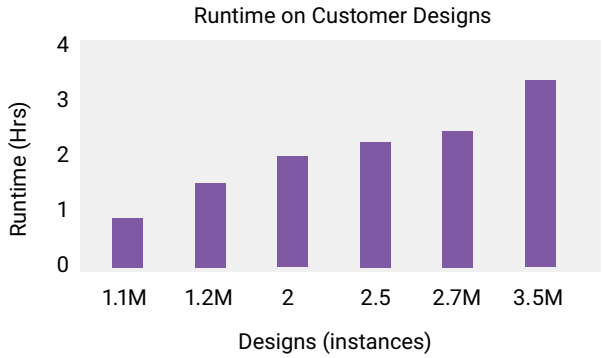
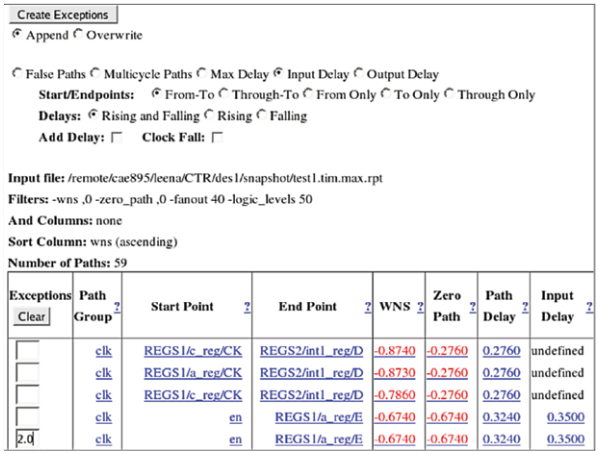


Figure 9: Fast DC Explorer runtimes allow for multiple iterations per day on large designs

Early Visibility Into Implementation Results

DC Explorer provides valuable early visibility into implementation results. It generates easy-to-navigate, HTMLbased timing reports, shown in Figure 10, which designers can use to quickly identify infeasible timing paths and correct the RTL and timing constraints before RTL synthesis.

Timing paths can be sorted, for example, by worst negative slack to uncover the most severe timing issues right away. False paths, multicycle paths, inputto-output maximum delays and input/ output delays can all be easily specified in detail. Then, with the click of a button, the constraints are generated for use in subsequent analyses. By identifying and correcting critical timing issues before they are encountered in the implementation phase, designers save significant time downstream in the flow.



Create Exceptions

Append Overwrite

False Paths Multicycle Paths Max Delay Input Delay Output Delay

Start/Endpoints: From-To Through-To From Only To Only Through Only

Delays: Rising and Falling Rising Falling

Add Delay: **Clock Fall:**

Input file: /remote/cae895/leena/CTR/des1/snapshot/test1.tim.max.rpt

Filters: -wns ,0 -zero_path ,0 -fanout 40 -logic_levels 50

And Columns: none

Sort Column: wns (ascending)

Number of Paths: 59

Exceptions Clear	Path Group ?	Start Point ?	End Point ?	WNS ?	Zero Path ?	Path Delay ?	Input Delay ?
	clk	REGS1/c_reg/CK	REGS2/int1_reg/D	-0.8740	-0.2760	0.2760	undefined
	clk	REGS1/a_reg/CK	REGS2/int1_reg/D	-0.8730	-0.2760	0.2760	undefined
	clk	REGS1/c_reg/CK	REGS2/int1_reg/D	-0.7860	-0.2760	0.2760	undefined
	clk	en	REGS1/a_reg/E	-0.6740	-0.6740	0.3240	0.3500
2.0	clk	en	REGS1/a_reg/E	-0.6740	-0.6740	0.3240	0.3500

Figure 10: DC Explorer generates HTML timing reports to accelerate early feedback on design feasibility

Easy to Adopt

DC Explorer is script-compatible with Design Compiler for fast, easy deployment into existing synthesis flows. It can run off of a DC synthesis script without requiring any modifications or adjustments. In addition, the hierarchical flow support in DC Explorer is also consistent with that of DC Ultra. The inputs to DC Explorer, shown in Figure 11, are the design RTL, the logical library (db) and the design constraints (SDC). Floorplan constraints and the physical library are optional. If a Milkyway™ physical library is available as an input, DC Explorer reports mismatches between the physical and logical libraries.

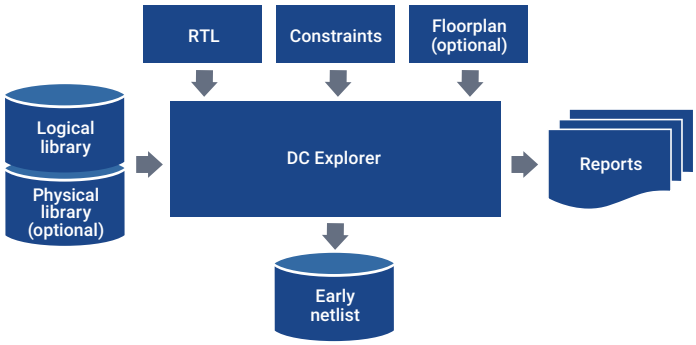


Figure 11: DC Explorer inputs and outputs

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. DC Explorer takes advantage of multicore compute servers to deliver a 2X improvement in runtime on quad-core platforms.