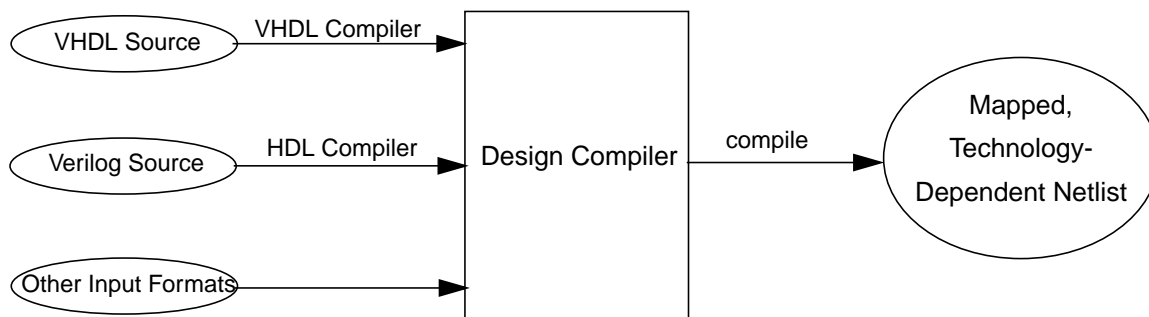


1

Introduction to Design Compiler

Design Compiler is the core of the Synopsys synthesis software products. It provides constraint-driven optimization and supports a wide range of design styles. The Design Compiler tools synthesize your HDL description into a technology-dependent, gate-level design. Design Compiler optimizes combinational or sequential designs for speed, area, and power, and supports both flat and hierarchical designs.

Figure 1-1 Design Compiler Overview



Design Compiler provides links to electronic design automation (EDA) tools, such as place and route tools, and post layout resynthesis techniques, such as in-place optimization. These EDA tool links enable sharing of information (such as forward-directed constraints and delays) between Design Compiler and external tools.

This chapter includes the following sections:

- Design Compiler Products
- User Interfaces
- Supported File Formats
- Supported File Formats
- License Requirements
- Resource Requirements
- High-Level Design Flow

Design Compiler Products

Synopsys provides a spectrum of Design Compiler products, which vary in the complexity of the features offered. Choose the right product for your design environment, based on your synthesis requirements.

Using the Design Compiler products, you can

- Produce fast, area-efficient ASIC designs by using user-specified gate-array, FPGA, or standard-cell libraries
- Translate designs from one technology to another

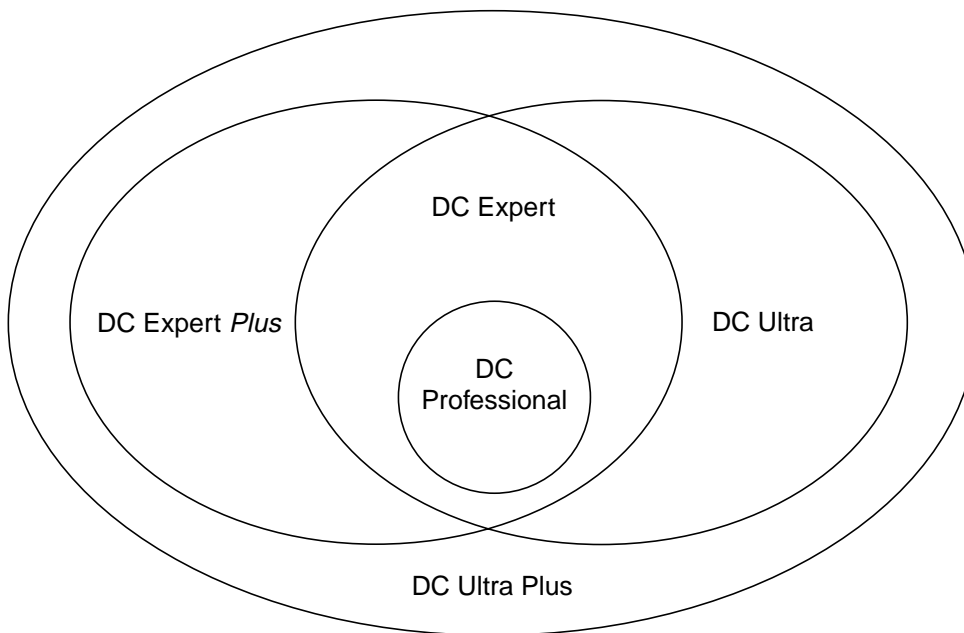
- Explore design tradeoffs involving design constraints such as timing, area, and power under various loading, temperature, and voltage conditions
- Synthesize and optimize a finite state machine, including automatic state assignment and state minimization
- Integrate netlist input and netlist or schematic output into third-party environments while still supporting delay information and place and route constraints
- Create and partition hierarchical schematics automatically

The Design Compiler products include

- DC Professional
- DC Expert
- DC Expert *Plus*
- DC Ultra
- DC Ultra Plus

Figure 1-2 shows the relationship between the features in the Design Compiler products.

Figure 1-2 Design Compiler Products



The following sections describe these products.

DC Professional

The DC Professional tools are applied to typical ASIC designs that employ CMOS technology. These designs can utilize multiple clocks; however, the clocks must have the same frequency. DC Professional does not support time borrowing for latch-based designs.

The set of features provided in the DC Professional product comprise the core synthesis features. These features are available in all Design Compiler products.

The core synthesis features include

- Hierarchical compile (top-down or bottom-up)
- Full and incremental compile techniques

- Sequential optimization for complex flip-flops and latches
- I/O pad insertion and optimization
- Finite state machine (FSM) optimization
- Buffer balancing (within a hierarchical block)

DC Expert

The DC Expert tools are applied to high-performance ASIC and IC designs.

In addition to the core synthesis features, DC Expert provides the following features:

- Multifrequency clocks
- Time borrowing for latch-based designs
- Critical path resynthesis
- Retiming for minimum clock period
- In-place optimization
- Specification of both minimum and maximum constraints for optimization and timing analysis
- Synthesis of control logic and certain data-path and structured logic
- Modeling of multiple paths that share a startpoint and endpoint but have different timing requirements

DC Expert *Plus*

The DC Expert *Plus* tools are applied to high-performance ASIC and IC designs that utilize scan test techniques.

In addition to the DC Expert features, DC Expert *Plus* provides integrated design-for-test capabilities, including constraint-driven scan insertion during compile.

DC Ultra

The DC Ultra tools are applied to high-performance deep submicron ASIC and IC designs, where maximum control over the optimization process is required.

In addition to the DC Expert features, DC Ultra provides the following features:

- Additional high-effort delay minimization algorithms
- Support for the cell-degradation design rule
- Additional options for the `set_cost_priority` command
- Finer optimization control with the `set_compile_directives` command
- Additional options for `report_timing`
- Support for behavioral optimization of arithmetic (BOA)
- Support for behavioral retiming (BRT)

DC Ultra Plus

The DC Ultra Plus tools are applied to high-performance deep submicron ASIC and IC designs that utilize scan test techniques.

In addition to the DC Ultra features, DC Ultra Plus provides integrated design-for-test capabilities.

User Interfaces

Design Compiler provides the following user interfaces:

The Design Compiler shell (dc_shell)

dc_shell is the command-line interface for the synthesis products (Design Compiler products and options).

To run dc_shell, you must have a Design-Compiler license.

This book describes the dc_shell interface.

Design Analyzer

The Design Analyzer interface is the graphical user interface (GUI) for the synthesis products. The GUI includes most synthesis commands in its menus. You can access any synthesis command not implemented in the menus by using the GUI command window.

To run Design Analyzer, you must have a Design-Analyzer license in addition to the Design-Compiler license.

The *Design Analyzer Reference Manual* describes the Design Analyzer interface.

These user interfaces support both Design Compiler shell language (dcsh) and tool command language (Tcl). For information about these languages, see the *Design Compiler Command-Line Interface Guide*.

Design Compiler Options

Synopsys provides several products that are integrated into the Design Compiler environment as options of Design Compiler. Unless otherwise stated, you must have the Design Compiler product to use these options.

HDL Compiler Products

The HDL Compiler products include

- HDL Compiler for Verilog
- VHDL Compiler

These products read and write Verilog or VHDL design files. The HDL Compiler product reads these design files and performs translation and architectural optimization of the designs. For more information about the HDL Compiler products, see the *HDL Compiler for Verilog Reference Manual* or the *VHDL Compiler Reference Manual*.

RTL Analyzer

The RTL Analyzer product measures design performance and relates it directly to HDL source code. It allows you to locate the source code underlying timing-, area-, and power-related issues in a design and to refine the source code before and after synthesis. For more information, see the *RTL Analyzer User Guide*.

FPGA Compiler

The FPGA Compiler product is available either as a stand-alone product or as an option of Design Compiler. As an option to Design Compiler, FPGA Compiler enables input of FPGA technology libraries and data formats as well as providing FPGA-specific optimization algorithms with features for high-performance FPGA implementations. For more information, see the *FPGA Compiler User Guide*.

Behavioral Compiler

The Behavioral Compiler product is a high-level synthesis product that synthesizes data-path, memory, and control logic from a behavioral HDL specification. It schedules operations such as additions, multiplications, memory reads, and constraints to control latency, throughput, resource types, and I/O activity.

Behavioral Compiler is used with a behavioral specification and generates optimally scheduled designs (that can be simulated and synthesized) as input to Design Compiler.

You can run Behavioral Compiler as a stand-alone product or as an option to Design Compiler. For more information, see the Behavioral Compiler documentation.

Test Compiler Products

The Synopsys test products combine test synthesis, automatic test-pattern generation (ATPG), fault simulation, and test management to automate design-for-test (DFT), helping design teams get highly testable designs to market in a shorter time. For more information, see the Test Compiler documentation.

Power Products

The Synopsys power products offer a complete methodology for power, including

- Power analysis (DesignPower)

You can run DesignPower as a stand-alone product or as an option of Design Compiler.

- Optimization and links-to-layout (Power Compiler)

Power Compiler requires at least the DC Expert license.

For more information about the Power products, see the *Power Products Reference Manual*.

Floorplan Manager

The Floorplan Manager product enables the use of post-layout and post-floorplan physical information to reoptimize designs. It also provides methods for exchanging placement and constraint information between Design Compiler and physical design tools. For more information, see the *Floorplan Manager User Guide*.

ECO Compiler

The ECO Compiler product provides a formal methodology for handling late-stage engineering change orders (ECOs). ECO Compiler automates implementing an ECO into an already synthesized design and provides an alternative to resynthesizing a stable block netlist that has undergone layout and timing analysis. It reuses as much of the layout and timing as possible by preserving and reusing as many cells and nets from the netlist as possible. For more information, see the *ECO Compiler User Guide*.

Supported File Formats

Table 1-1 shows the supported file formats for all data input and output by Design Compiler.

Table 1-1 Supported File Formats

| Data | Formats |
|-------------|--|
| | EDIF |
| | LSI Logic Corporation netlist format (LSI) |
| | Mentor Intermediate Format (MIF) |

Table 1-1 Supported File Formats (continued)

| Data | Formats |
|-----------------|--|
| | Programmable logic array (PLA) |
| | Synopsys equation |
| Netlist | Synopsys state table |
| | Synopsys database format (.db) |
| | Tegas Design Language (TDL) |
| | Verilog |
| | VHDL |
| Timing | Standard Delay Format (SDF) |
| Command Script | dcsh, Tcl |
| Cell Clustering | Physical Design Exchange Format (PDEF) |
| Library | Synopsys library source (.lib) |
| | Synopsys database format (.db) |
| Parasitics | dc_shell Command scripts |

All formats except .db, EDIF, equation, PLA, and state table require special license keys.

License Requirements

You need licenses to use Design Compiler and its related products.

Table lists the Design Compiler tools and interfaces and their required licenses. When multiple licenses are listed, all are required.

Table 1-2 License Requirements

| Design Compiler Tool | License Name |
|-----------------------------|---|
| DC Professional | Design-Compiler |
| DC Expert | Design-Compiler, DC-Expert |
| DC Expert <i>Plus</i> | Design-Compiler, Test-Compiler, TestManager |
| DC Ultra | Design-Compiler, DC-Ultra-Features, DC-Ultra-Opt |
| DC Ultra Plus | Design-Compiler, DC-Ultra-Features, DC-Ultra-Opt, Test-Compiler, Test-Manager |

| Design Compiler Option | License Name |
|---------------------------------------|--|
| Design Analyzer | Design-Analyzer |
| Behavioral Compiler | Behavioral-Compiler |
| DesignPower | Power-Analysis |
| ECO Compiler | ECO-Compiler |
| Floorplan Manager | Floorplan-Management |
| FPGA Compiler | FPGA-Option (or FPGA-Compiler) |
| HDL Compiler for Verilog | HDL-Compiler |
| Power Compiler (includes DesignPower) | DC-Expert, Power-Optimization or DC-Expert, Power-Analysis, Power-Optimization-Upgrade |
| RTL Analyzer | RTL-Analyzer |
| Test Compiler | Test-Compiler |
| Test Compiler Plus | Test-Compiler, Test-Compiler-Plus |
| VHDL Compiler | VHDL-Compiler |

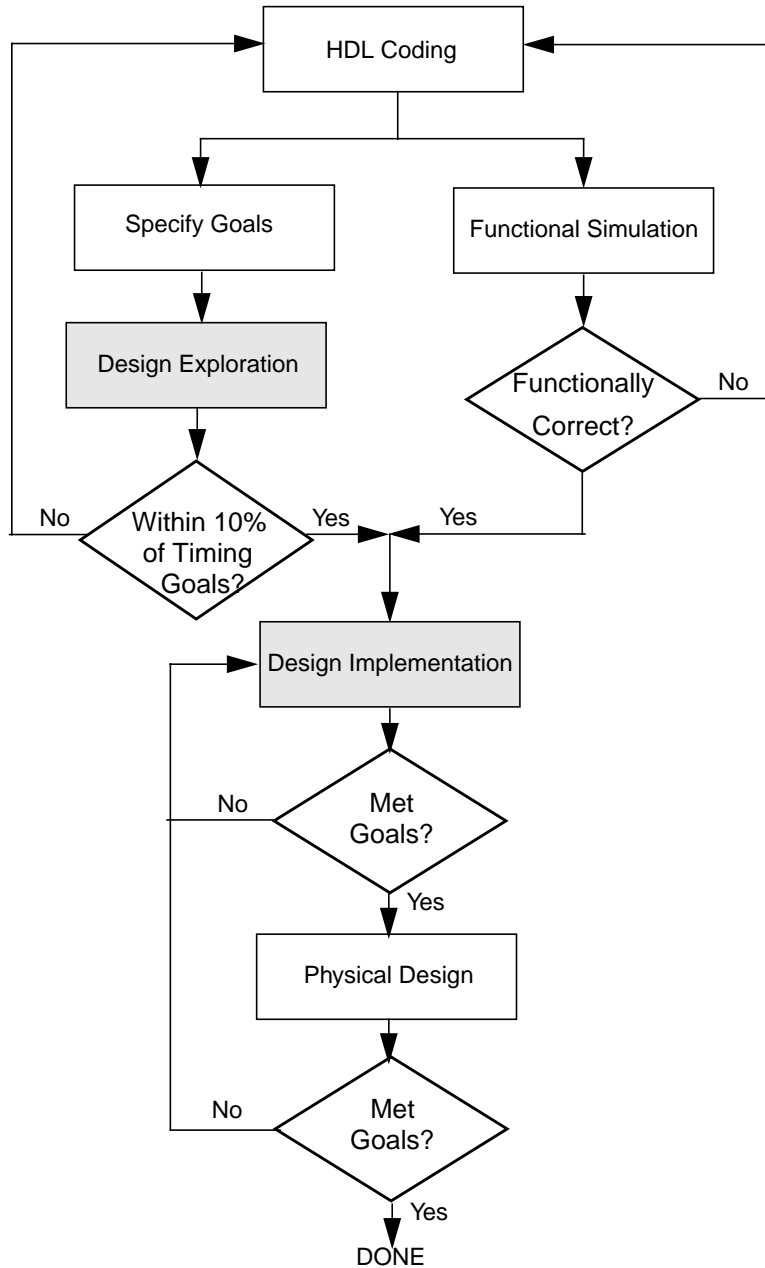
Resource Requirements

To compile 100K gates, you need at least 512 MB of RAM and 1 GB of swap space. In addition, you need approximately 1 GB of disk space for each 100K gates, plus approximately 1 GB for the Synopsys synthesis software.

High-Level Design Flow

Figure 1-3 shows the complete high-level design flow. The shaded areas show the synthesis tasks. This manual describes how to use Design Compiler to perform these tasks. For an overview of the synthesis design flow, see Chapter 2, “Getting a Quick Start.”

Figure 1-3 High-Level Design Flow



The high-level design flow shown in Figure 1-3 consists of the following steps:

1. Start with an HDL description (Verilog or VHDL) of your design.

2. In parallel, perform design exploration (preliminary synthesis) and functional simulation.

Design exploration uses the default synthesis algorithm to gauge the design performance against your goals. If the performance violates the timing goals by more than 10 percent, modify the HDL code or refine your goals and constraints.

Functional simulation verifies that the design performs the desired function. If the design does not function as desired, you must modify the HDL code.

Continue performing design exploration and functional simulation until the design functions correctly and is within 10 percent of the timing goals.

3. Perform design implementation (final synthesis) to implement the design.

Final synthesis uses the full power of Design Compiler to generate a design that meets your goals.

After you synthesize the design into a gate-level netlist, verify that the design meets your goals. If the circuit does not meet your goals, generate and analyze reports to determine the techniques you might use to correct the problems.

4. After you verify the functionality, timing, and design goals, complete the physical design (either in house or by sending it to your semiconductor vendor).

Analyze the design's performance using back-annotated data. If the results are not within your design goals, return to step 3. If the results meet your design goals, you are finished with the design cycle.