Design Vision™

User Guide

Version X-2005.09, September 2005

Comments?
Send comments on the documentation by going to http://solvnet.synopsys.com, then clicking "Enter a Call to the Support Center."

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

The license agreement with Synopsys permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:

> "This document is duplicated with the permission of Synopsys, Inc., for the exclusive use of
> ____ and its employees. This is copy number __________."

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AMPS, Arcadia, C Level Design, C2HDL, C2V, C2VHDL, Cadabra, Calaveras Algorithm, CATS, CRITIC, CSim, Design Compiler, DesignPower, DesignWare, EPIC, Formality, HSIM, HSPICE, Hypermodel, iN-Phase, in-Sync, Leda, MAST, Meta, Meta-Software, ModelTools, NanoSim, OpenVera, PathMill, Photolynx, Physical Compiler, PowerMill, PrimeTime, RailMill, RapidScript, Saber, SiVL, SNUG, SolvNet, Superlog, System Compiler, Testify, TetraMAX, TimeMill, TMA, VCS, Vera, and Virtual Stepper are registered trademarks of Synopsys, Inc.

Trademarks (™)

Active Parasitics, AFGen, Apollo, Apollo II, Apollo-DPII, Apollo-GA, ApolloGAII, Astro, Astro-Rail, Astro-Xtalk, Aurora, AvanTestchip, AvanWaves, BCView, Behavioral Compiler, BOA, BRT, Cedar, ChipPlanner, Circuit Analysis, Columbia, Columbia-CE, Comet 3D, Cosmos, CosmosEnterprise, CosmosLE, CosmosScope, CosmosSE, Cyclelink, Davinci, DC Expert, DC Expert *Plus*, DC Professional, DC Ultra, DC Ultra Plus, Design Advisor, Design Analyzer, Design Vision, DesignerHDL, DesignTime, DFM-Workbench, Direct RTL, Direct Silicon Access, Discovery, DW8051, DWPCI, Dynamic Model Switcher, Dynamic-Macromodeling, ECL Compiler, ECO Compiler, EDAnavigator, Encore, Encore PQ, Evaccess, ExpressModel, Floorplan Manager, Formal Model Checker, FoundryModel, FPGA Compiler II, FPGA *Express*, Frame Compiler, Galaxy, Gatran, HANEX, HDL Advisor, HDL Compiler, Hercules, Hercules-Explorer, Hercules-II, Hierarchical Optimization Technology, High Performance Option, HotPlace, HSIM^{plus}, HSPICE-Link, i-Virtual Stepper, iN-Tandem, Integrator, Interactive Waveform Viewer, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, JVXtreme, Liberty, Libra-Passport, Libra-Visa, Library Compiler, Magellan, Mars, Mars-Rail, Mars-Xtalk, Medici, Metacapture, Metacircuit, Metamanager, Metamixsim, Milkyway, ModelSource, Module Compiler, MS-3200, MS-3400, Nova Product Family, Nova-ExploreRTL, Nova-Trans, Nova-VeriLint, Nova-VHDLlint, Optimum Silicon, Orion_ec, Parasitic View, Passport, Planet, Planet-PL, Planet-RTL, Polaris, Polaris-CBS, Polaris-MT, Power Compiler, PowerCODE, PowerGate, ProFPGA, ProGen, Prospector, Protocol Compiler, PSMGen, Raphael, Raphael-NES, RoadRunner, RTL Analyzer, Saturn, ScanBand, Schematic Compiler, Scirocco, Scirocco-i, Shadow Debugger, Silicon Blueprint, Silicon Early Access, SinglePass-SoC, Smart Extraction, SmartLicense, SmartModel Library, Softwire, Source-Level Design, Star, Star-DC, Star-MS, Star-MTB, Star-Power, Star-Rail, Star-RC, Star-RCXT, Star-Sim, Star-SimXT, Star-Time, Star-XP, SWIFT, Taurus, TimeSlice, TimeTracker, Timing Annotator, TopoPlace, TopoRoute, Trace-On-Demand, True-Hspice, TSUPREM-4, TymeWare, VCS Express, VCSi, Venus, Verification Portal, VFormal, VHDL Compiler, VHDL System Simulator, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (^{SM})

MAP-in, SVP Café, and TAP-in are service marks of Synopsys, Inc.

SystemC is a trademark of the Open SystemC Initiative and is used under license.
ARM and AMBA are registered trademarks of ARM Limited.
All other product or company names may be trademarks of their respective owners.

Printed in the U.S.A.

Document Order Number: 37175-000 ZA
Design Vision User Guide, version X-2005.09

Contents

Preface

This preface includes the following sections:

- What’s New in This Release
- About This User Guide
- Customer Support

What’s New in This Release

Information about new features, enhancements, and changes; known problems and limitations; and resolved Synopsys Technical Action Requests (STARs) is available in the *Design Vision Release Notes* in SolvNet.

To see the *Design Vision Release Notes*,

1. Go to the Synopsys Web page at http://www.synopsys.com and click SolvNet.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)
3. Click Release Notes in the Main Navigation section (on the left), click Design Vision, then click the release you want in the list that appears at the bottom.

About This User Guide

This section contains information about the target audience of this document, where to find other pertinent publications, and documentation conventions used in this manual.

Audience

This user guide is for logic design engineers who have some experience using Design Compiler and who want to use the visualization features of Design Vision for synthesis or analysis. To use this user guide, you should be familiar with

- Synthesis using Design Compiler
- VHDL or Verilog HDL
- The UNIX or Linux operating system

Related Publications

For additional information about Design Vision, see

- Synopsys Online Documentation (SOLD), which is included with the software for CD users or is available to download through the Synopsys Electronic Software Transfer (EST) system
- Documentation on the Web, which is available through SolvNet at http://solvnet.synopsys.com
- The Synopsys MediaDocs Shop, from which you can order printed copies of Synopsys documents, at http://mediadocs.synopsys.com

See also the documentation for the following Synopsys products:

- Design Compiler
- DFT Compiler
- BSD Compiler
- Power Compiler

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates command syntax.
`Courier italic`	Indicates a user-defined value in Synopsys syntax, such as *`object_name`*. (A user-defined value that is not Synopsys syntax, such as a user-defined value in a Verilog or VHDL statement, is indicated by regular text font italic.)
`Courier bold`	Indicates user input—text you type verbatim—in Synopsys syntax and examples. (User input that is not Synopsys syntax, such as a user name or password you enter in a GUI, is indicated by regular text font bold.)
[ ]	Denotes optional parameters, such as *`pin1`* `[`*`pin2 ... pinN`*`]`
|	Indicates a choice among alternatives, such as `low | medium | high` (This example indicates that you can enter one of three possible values for an option: low, medium, or high.)
_	Connects terms that are read as a single term by the system, such as `set_annotated_delay`
Control-c	Indicates a keyboard combination, such as holding down the Control key and pressing c.
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services including software downloads, documentation on the Web, and “Enter a Call to the Support Center.”

To access SolvNet,

1. Go to the SolvNet Web page at http://solvnet.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)

If you need help using SolvNet, click HELP in the top-right menu bar or in the footer.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a call to your local support center from the Web by going to http://solvnet.synopsys.com (Synopsys user name and password required), then clicking “Enter a Call to the Support Center.”
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at http://www.synopsys.com/support/support_ctr.
- Telephone your local support center.
  - Call (800) 245-8005 from within the continental United States.
  - Call (650) 584-4200 from Canada.
  - Find other local support center telephone numbers at http://www.synopsys.com/support/support_ctr.

1

Introduction to Design Vision

Design Vision is the graphical user interface (GUI) to the Synopsys synthesis environment. It provides analysis tools for viewing and analyzing your design at the generic technology (GTECH) level and the gate level. Design Vision provides two modes of operation: XG mode and DB mode. XG mode uses optimized memory management techniques that increase the tool capacity and can reduce runtime. DB mode uses the original Design Compiler memory management techniques.

The *Design Vision User Guide* discusses features supported in DB mode. For information about features supported only in XG mode, see the Design Vision online help. For information about Design Compiler features supported in XG mode, see the *XG Mode User Guide*.

This chapter contains the following sections:

- Features and Benefits
- User Interfaces
- Methodology
- Supported Formats
- Licensing
- Supported Platforms
- Design Vision and Other Synopsys Products

Features and Benefits

Design Vision has these features:

- Window- and menu-driven interface to Synopsys synthesis (Design Compiler)
- Analysis visualization capabilities that include
  - A hierarchy browser (logic hierarchy view) for easy visualization and navigation of your design hierarchy (including viewing area attributes)
  - Histograms for visualizing trends in various metrics, for example slack and capacitance
  - A timing analysis driver window for viewing details about the timing paths with the worst slack in the design
  - Path inspector windows for examining the path status and detailed information about clock network and path data elements on a timing path you select
  - Path schematics for visually examining any timing paths or design logic you select
  - Path profile views for visually examining the contributions of different cells and nets to the total delay of any timing path you select
  - Design schematics and symbol views for visually examining high-level and low-level design connectivity
  - Report views for viewing and saving design and timing reports
  - List views for examining, sorting, and filtering information about various types of design objects

  - A properties viewer for viewing object properties and editing attribute values

- Command-line interface integrated in the GUI
- Scripting support for all Design Compiler tool command language (dctcl) commands
- A man page viewer for viewing man pages for commands, variables, and error messages
- A dcsh emulation mode with scripting support for dc_shell (dcsh) commands

Using the features of Design Vision, you can

- Invoke Design Compiler synthesis and reporting
- Explore design structure
- Perform timing analysis for blocks you are synthesizing
- Visualize overall timing with Design Vision graphical analysis
- Perform detailed visual analysis of timing paths
- Insert scan cells and logic into a design, with or without test points

User Interfaces

Design Vision offers menus and dialog boxes for important Design Compiler functions. Menus also offer visual analysis features and other design viewing features that are not Design Compiler functions. Design Vision also includes a command-line interface that provides access to all the capabilities of Synopsys synthesis tools.

The Design Vision GUI and command-line interface use the dctcl command language by default. If you need to use dcsh command scripts, you can run Design Vision in DB mode with dcsh mode emulation. For details, see the Design Vision online Help.

Methodology

Design Vision allows you to use the same design methodology and scripts you currently use and to extend your methodology with Design Vision visual analysis. Many Design Compiler commands are available on Design Vision menus. All Design Compiler functions are available through the Design Vision command-line interface.

Supported Formats

With Design Vision you can access all the files supported by Design Compiler. Table 1-1 shows the supported design file formats. All netlist formats except .db, EDIF, equation, PLA, state table, Verilog, and VHDL require special license keys.

Table 1-1 Supported File Formats

Data	Formats
Netlist	EDIF
	LSI Logic Corporation netlist format (LSI)
	Mentor Intermediate Format (MIF)
	Programmable logic array (PLA)
	Synopsys equation
	Synopsys state table
	Synopsys database format (.db)
	Synopsys dc_shell database format (.ddc) for XG mode
	Tegas Design Language (TDL)
	Verilog
	VHDL
	Xilinx Netlist Format (XNF)
Timing	Standard Delay Format (SDF)
Command Script	dctcl, dcsh
Library	Synopsys internal library format (.lib)
	Synopsys database format (.db)
Parasitics	dc_shell command scripts

Licensing

To use Design Vision, you need the Design-Vision license.

Synopsys licensing software and the documentation describing it are now separate from the tools that use it. You install, configure, and use a single copy of Synopsys Common Licensing (SCL) for all Synopsys tools. Because it provides a single, common licensing base for all Synopsys tools, SCL reduces licensing administration complexity, minimizing the effort you expend in installing, maintaining, and managing licensing software for Synopsys tools.

For complete Synopsys licensing information, see the *Common Licensing Installation and Administration Guide*. This guide provides detailed information on SCL installation and configuration, including examples of license key files and troubleshooting guidelines.

Supported Platforms

Design Vision is supported on the same platforms that support Design Compiler and the other synthesis tools. For details, see the *Installation Guide*.

Your hardware and operating system vendor has required patches available for your system. For more information about the supported platforms and the operating system patches necessary to run Synopsys software on supported platforms, go to

http://www.synopsys.com/products/sw_platform.html

From this Web page you can navigate to the Supported Platforms Guide for your release.

Design Vision and Other Synopsys Products

As a visual analysis tool and GUI for Synopsys synthesis, Design Vision works with Design Compiler to synthesize and analyze your design.

The Design Vision and Synopsys PrimeTime tools have similar timing visualization features; however, the tools have different timing engines and differ in their application to analysis. Design Vision has the same static timing engine as Design Compiler. Use Design Vision to perform timing analysis and modification of blocks you are synthesizing. Use PrimeTime for static timing sign-off or for analyzing the timing of a chip or of large portions of a chip.

2

Before You Start

This chapter contains general and specific information you need to know before you use Design Vision for the first time. In this chapter you can find the following sections:

The Design Vision Documentation Set

You can find most of what you need to know to run Design Vision in the Design Vision documentation set.

The Design Vision documentation set is divided into these parts:

- *Design Compiler Tutorial Using Design Vision*
- *Design Vision User Guide*
- Design Vision online Help

Other sources of information include man pages, the SolvNet knowledge base, and the Customer Support Center. For information about accessing these sources of information, see “Customer Support” on page xi.

Design Compiler Tutorial Using Design Vision

The *Design Compiler Tutorial Using Design Vision* is an introduction to ASIC design using Synopsys Design Compiler and Design Vision.

The tutorial is a group of guided exercises in which you use Design Compiler to synthesize and analyze a simple design. The tutorial introduces Design Vision as the GUI for synthesis and provides some post-synthesis analysis that takes advantage of the Design Vision visual representation of analysis results.

Use the *Design Compiler Tutorial Using Design Vision* if you are new to Design Compiler. However, even if you are an experienced Design Compiler user, the tutorial is a useful introduction to the features of Design Vision.

In the tutorial, you can read about and perform exercises for basic synthesis and analysis. For example, you

- Read a design from disk into the tool’s workspace
- Define design constraints to direct synthesis results
- Compile the design to gates
- View analysis results
- Modify the design
- Resynthesize the design
- Reanalyze the design

Design Vision User Guide

The *Design Vision User Guide* assumes you are familiar with basic Design Compiler concepts.

The user guide provides guidance in solving particular problems. For example, it presents short procedures that use the analysis visualization features of Design Vision to locate and solve timing problems.

Sometimes steps in a procedure refer to actions without further explanation: for example, “Create a histogram” or “Create a path schematic.” Such steps refer to features of Design Vision that are explained in the Design Vision online Help.

The *Design Vision User Guide* does not contain specific information about individual menu items or dialog boxes. For such information, see the Design Vision online Help.

In Chapter 3, “Performing Basic Tasks,” experienced Design Compiler users can learn how to do certain familiar synthesis tasks using Design Vision. However, the user guide explains such topics only briefly.

Design Vision Online Help

The Design Vision online Help system is integrated in the Design Vision GUI. You access the Help system from the Help menu in the Design Vision window. Design Vision online Help explains the details of features and procedures. For example, if you need help performing a step in a procedure presented in the user guide, you can find the information you need in the online Help system.

Information in online Help is grouped in the following categories:

- Feature topics

  Overviews of Design Vision window components and tools.

- Design Vision Tour

  A visual “quick start” tour that introduces you to the basic visual analysis tools and other useful features in Design Vision.

- How-to topics

  Procedures for accomplishing general synthesis and analysis tasks.

- Reference topics

  Explanations for views, toolbar buttons, menu commands, and dialog box options.

To access the Design Vision online Help system,

- Choose Help > Online Help.

Design Vision online Help is a browser-based HTML Help system designed for viewing in Netscape Web browser versions 7.0 and above or 4.78 for UNIX or Linux. Design Vision Help is not supported in Netscape version 6.x.

Note:

> When you access Design Vision online Help from within Design Vision, the netscape executable file must be on your UNIX path.

The online Help system makes extensive use of Java, JavaScript, and style sheets. To configure your browser for viewing online Help, make sure the advanced preferences are set as follows:

- In Netscape 7, the Enable Java and XSLT options are both selected, JavaScript, and style sheets
- In Netscape 4.78, the Enable Java, Enable JavaScript, and Enable Style Sheets options are all selected, and the Enable Java Plugin option is deselected

If you reset preferences while the Help system is open, click the Reload button on the browser's navigation toolbar after you reset the preferences.

You can view the Design Vision online Help stand-alone by choosing File > Open File (Netscape version 7) or File > Open Page (Netscape version 4.78) and opening the file named index.html in the online Help directory: $SYNOPSYS/doc/syn/html/dvoh/enhanced.

Setup Files for Design Vision

When you start Design Vision, it reads the three standard Design Compiler setup files that dc_shell reads. These files have the same names, .synopsys_dc.setup, and reside in the installation, home, and design directories.

In addition, Design Vision reads two other sets of setup files:

- .synopsys_dv.tcl

  These files contain commands that apply to the Design Vision shell and GUI but do not apply to dc_shell. Design Vision reads these files when you start a session with or without the GUI.

- .synopsys_dv_gui.tcl

  These files contain commands that apply only to the GUI. Design Vision reads these files both when you start the session (except when you use the `-no_gui` option) and when you open or reopen the GUI from the Design Vision shell.

Setup files contain commands that perform basic setup tasks, such as initializing parameters and variables and declaring design libraries. Settings from the .synopsys_dv.tcl files override settings from the .synopsys_dc.setup files. In addition, you can use the setup file in your home or design directory to define dctcl or dcsh scripts and to set display options for design schematics.

These topics are covered in detail in “Exploring the Graphical User Interface” on page 2-9 and “Using Scripts” on page 2-26 respectively.

Besides reading the setup files, Design Vision also loads application preferences from a file named .synopsys_dv_prefs.tcl. Design Vision preferences include global options (such as font characteristics and various default controls) and schematic display options. Design Vision reads the preferences file when you start a session or when you open or reopen the GUI from the Design Vision shell.

For more information about the Design Vision setup and preference files, see the Design Vision online Help. For more information about the locations of setup files and initialization settings for synthesis, see the *Design Compiler User Guide*.

Starting and Exiting the Graphical User Interface

You can start a Design Vision session in either XG mode or DB mode. You cannot switch from mode to mode in the same session. To operate in either mode, Design Vision must start up and initialize in that mode.

To start Design Vision in XG mode,

- At the UNIX or Linux shell prompt, enter

```
% design_vision
```

To start Design Vision in DB mode,

- At the UNIX or Linux shell prompt, enter

```
% design_vision -tcl -db_mode
```

When you start Design Vision, the Design Vision window appears by default. In addition, the design_vision command-line prompt (design_vision-xg-t in XG mode or design_vision-t in DB mode) appears in the shell where you started Design Vision.

You can close and reopen the Design Vision GUI without exiting the Design Vision shell. In addition, you can start the Design Vision shell without opening the GUI.

To close the GUI without exiting Design Vision,

- Choose File > Close.

  Alternatively, you can enter `gui_stop` on the command line.

To open or reopen the GUI from the Design Vision shell,

- Enter the following on the design_vision command line.

```
design_vision-xg-t> gui_start
```

To start Design Vision in XG mode without opening the GUI,

- At the UNIX or Linux shell prompt, enter

```
% design_vision -no_gui
```

To set Design Vision to start without the GUI by default,

- Set the following variable in your .synopsys_dc.setup file:

```
set gui_auto_start 0
```

You can include other options on the command line when you start Design Vision, such as

- `-checkout` to access additional licenses

- `-f` to execute the specified script file
- `-x` to include a dc_shell command that executes at startup
- `-display` *host-name* to display the Design Vision window on the terminal named *host-name*
- `-h` to display a list of the available options (without starting Design Vision)

To exit Design Vision,

- Choose File > Exit.

  Alternatively, you can enter `exit` or `quit` on the command line or press Control-c three times in the UNIX or Linux shell.

Exploring the Graphical User Interface

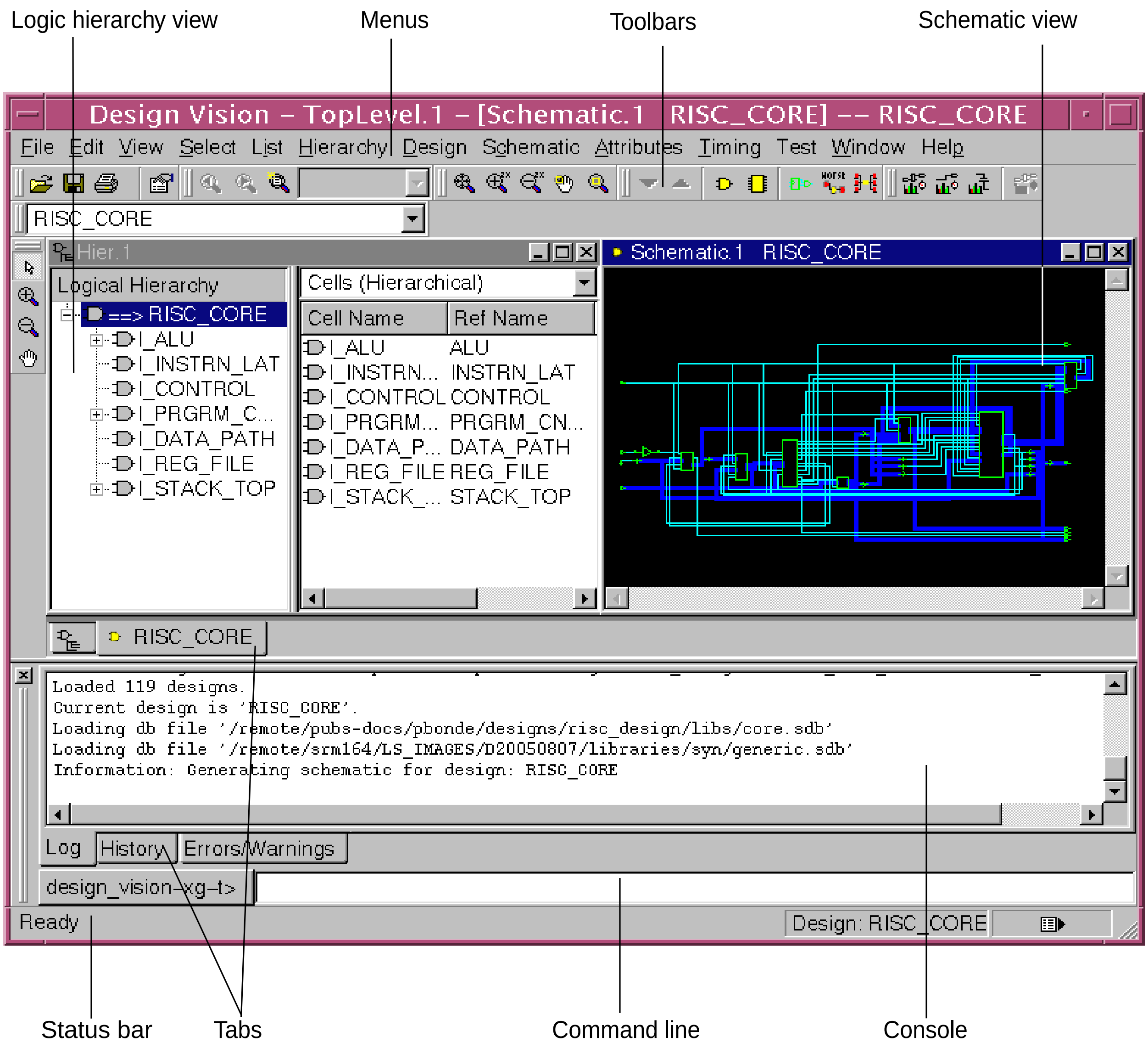
The Design Vision window is a top-level window in which you can display design information in various types of analysis views and windows. You can open multiple top-level windows, and you can resize, move, minimize, and maximize each window.

Each top-level window contains its own title bar, menu bar, toolbars, view windows, and status bar. You can enhance your viewing area by

- Displaying, hiding, moving, docking, or undocking individual toolbars
- Opening, closing, moving, resizing, minimizing, maximizing, docking, or undocking individual view windows
- Tiling or cascading all undocked view windows

Figure 2-1 shows the Design Vision window you see when you start Design Vision, read in a design, and open a design schematic for the top-level design.

Figure 2-1 The Design Vision Window



Cell Name	Ref Name
I_ALU	ALU
I_INSTRN...	INSTRN_LAT
I_CONTROL	CONTROL
I_PRGRM...	PRGRM_CN...
I_DATA_P...	DATA_PATH
I_REG_FILE	REG_FILE
I_STACK_...	STACK_TOP

```
Loaded 119 designs.
Current design is 'RISC_CORE'.
Loading db file '/remote/pubs-docs/pbonde/designs/risc_design/libs/core.sdb'
Loading db file '/remote/srm164/LS_IMAGES/D20050807/libraries/syn/generic.sdb'
Information: Generating schematic for design: RISC_CORE
```

The window consists of a title bar, a menu bar, and several toolbars at the top of the window and a status bar at the bottom of the window. You use the workspace between the toolbars and the status bar to display view windows containing graphical and textual design information.

You can open multiple Design Vision windows and use them to compare views, or different design information within a view, side by side. All open Design Vision windows share the same designs in memory and the same current timing information. However, you can configure the views independently for each window. To learn more, see the “Opening New Design Vision Windows” topic in online Help.

Title Bar

The title bar lists the product name (Design Vision) of the tool you are using, the name of the window you are viewing, and the name of the active view. The window name includes the unique number of the window. If you open multiple top-level windows, the windows are numbered sequentially throughout the session.

View windows display graphical or textual design information. The active view is the view window that has the mouse focus.

Menus

The menu bar provides menus with the commands you need to use the GUI. Menu commands that can also be used by pressing a toolbar button or typing a keyboard shortcut show representations of those alternatives with the command.

Some frequently used menu commands are also available on pop-up menus for individual views. To use a pop-up menu, right-click and choose a command.

Toolbars

Toolbars provide quick access to often used menu selections. The Design Vision toolbars include buttons and a design list. To determine the function of a toolbar button, hold the pointer over the button until an ToolTip message appears.

You can move the toolbars to other locations on or off the window or hide them to enhance your viewing area. To learn about the functions of each toolbar, see the “Toolbars” topic in online Help.

Status Bar

The status bar, located at the bottom of the Design Vision window, displays the information listed in Table 2-1.

Table 2-1 Information Displayed by the Status Bar

When you do this	**The status bar displays this information**
Select one object	Object name
Select multiple objects	Number of selected objects
Hold the pointer over a menu command, toolbar button, or tab	Information about the action it performs

You can quickly display the list of selected objects in the Selection List dialog box by clicking the ▤▸ button at the right end of the status bar.

You can hide or display the status bar by choosing View > Status Bar.

ToolTips

ToolTips display information about the GUI tool (toolbar button or dialog box option) under the pointer. You cannot disable these messages.

Views

View windows appear in the workspace area between the toolbars and the status bar in the Design Vision window.

Design Vision provides graphic views (such as histograms and schematics), hierarchy views (for traversing hierarchical structures and gathering design information at different hierarchy levels), and text views (such as reports and lists). In addition, console view windows provide a command line and views that display command status information, such as the command log and the command history list.

You can adjust the sizes of view windows for viewing purposes, and you can move them to different locations within the Design Vision window. You can arrange the open, undocked view windows by tiling or cascading them within the workspace in the Design Vision window. You can also dock or undock individual view windows by attaching them at edges of the Design Vision window or separating them from the edge so they can float within the workspace. When you click anywhere within a view window, Design Vision highlights its title bar to indicate that it has the focus (that is, it is the active view) and can receive keyboard and mouse input.

Design Vision displays a tab in the workspace for each undocked view you open. For more details, see the “View Windows” topic in online Help.

Viewing Design Information

When you start a Design Vision session, the view windows described in the following sections appear in the workspace:

- Logic Hierarchy View
- Command Console

In addition, you can display design information in the views, windows, and dialog boxes described in the following sections:

- Histogram Views
- Timing Analysis Driver Window
- Path Inspector Windows
- Path Profile Views
- Schematic and Symbol Views
- DRC Violation Browser and RTL Source Browser
- DFT Hold Time Analysis Window
- Report Views
- Design and Object Lists
- Properties Dialog Box
- Selection List Dialog Box

For additional information about these views, windows, and dialog boxes, see online Help.

Logic Hierarchy View

The logic hierarchy view helps you navigate your design and gather information. The view is divided into the following two panes:

- Instance tree
- Objects list

The instance tree lets you quickly navigate the design hierarchy and see the relationships among its levels. If you select the instance name of a hierarchical cell (an instance that contains subblocks), information about that instance appears in the object table. You can Shift-click or Control-click instance names to select combinations of instances.

By default, the object table displays information about hierarchical cells belonging to the selected instance in the instance tree. To display information about other types of objects, select the object types in the list above the table. You can display information about hierarchical cells, all cells, pins and ports, pins of child cells, and nets.

If some of the text in a column is missing because the column is too narrow, you can hold the pointer over it to display the information in an InfoTip. You can also adjust the width of a column by dragging its right edge left or right.

For more details about the logic hierarchy view, see the “Browsing the Design Hierarchy” topic in online Help.

Command Console

The console provides a command-line interface and displays information about the commands you use during the session in the following three views:

- Log view
- History view
- Errors and warnings view

You can enter dc_shell commands on the command line at the bottom of the console. Enter these commands just as you would enter them at the dc_shell prompt in a standard UNIX or Linux shell. When you issue a command (by pressing Return or clicking the prompt button to the left of the command line), Design Vision echoes the command output (including processing messages and any warnings or error messages) in the console log view.

The log view is displayed by default when you start Design Vision. The log view provides the session transcript. The history view provides a list of the commands that you have used during the session. The errors and warnings view displays error and warning messages. To select a view, click the tab at the bottom of the console.

You can display, edit, and reissue commands on the console command line by using the arrow keys to scroll up or down the command stack and to move the insertion point to the left or right on the command line. You can copy text in the log view and paste it on the command line the same way you would in a UNIX or Linux shell, by selecting the text with the left mouse button and pasting it with the middle button. You can also select commands in the history view and edit or reissue them on the command line.

You can expand the command line to display multiple lines (for example, when you need to enter a dctcl procedure or a multiple-line command) by pressing Shift-Return or by entering a backslash (\) and pressing Return. The command line automatically shrinks to a single line when you issue the command.

For more details about the command line and the console views, see the “Using dc_shell Commands” topic in online Help.

Histogram Views

Histograms are focal points of visual timing analysis. You can use histograms to view the overall timing performance of your logic design. Design Vision provides the following types of histograms: endpoint slack, path slack, and net capacitance.

- Endpoint slack histograms show a distribution of timing slack values for all endpoints in the design. You can choose maximum or minimum delay (setup or hold). The slack distribution provides an overall picture of how close the design is to meeting requirements.
- Path slack histograms show a distribution of timing slack values for selected paths or for the paths with the worst slack in the design. You can select a maximum or minimum delay type (setup or hold), set the maximum number of paths and the number of worst paths per endpoint, and select a path group. You can also specify individual paths to, from, or through selected objects (similar to the way you specify paths for timing reports).
- Net capacitance histograms show a distribution of net capacitance values for selected nets or for all nets in the design.

When you use the timing analysis driver window to view timing path details, you can also generate histograms that show the distribution of values for certain types of path details listed in the window.

For more details about histogram views, see the “Viewing Histograms” topic in online Help.

Timing Analysis Driver Window

The timing analysis driver window displays path details in a list of paths with the worst slack in the design. You can configure the list of paths and the details you want to view. You can also select paths in the window for further examination with other analysis tools.

The timing analysis driver window contains a timing path table and a button bar. The timing path table displays a list of paths found by the `get_timing_paths` command. The table columns show the startpoint name, endpoint name, and other details about each path. The button bar below the table lets you load a different list of paths, save the path details in a text file, customize the table columns, and access other analysis tools.

Path Inspector Windows

Path inspector windows let you examine detailed information about selected timing paths. When you select a path and load it into a path inspector window, you can view the path status, path and clock summaries, clock network and path data elements, path delay profiles, and a path schematic with clock path highlighting. The window provides a variety of tools for viewing different aspects of the selected path.

A path inspector window has its own menu bar, toolbars, and status bar. You can move or resize the window and manipulate the toolbars and status bar the same way you do with the Design Vision window. The path information appears on three panels between the toolbars and the status bar:

- The Status panel displays the path startpoint, endpoint, delay type, and slack. It also provides controls for automatically or manually loading another path.
- The Tab panel displays path summaries, clock network and path data element tables, and path delay profiles in a set of tabbed pages. You can copy the summaries and paste them into text files, export the tables to text files, customize the element tables, and generate delay calculation reports for selected cells or nets.
- The Schematic View panel displays a path schematic of the complete path (datapath, clock launch path, and clock capture path) with data and clock path overlays. You can display or hide the paths and the overlays. You can also add fanin and fanout logic to the path schematic.

The Status panel contains an automatic follow-selection mechanism that is enabled by default. This mechanism, when enabled, causes the contents of the path inspector window to “follow” the current selection in the Design Vision window. If you select a different path, the path inspector window immediately loads the data for the new path.

Path Profile Views

Path profile views help you examine the contributions of individual cells and nets to the total delay of a timing path.

The view window contains a table that shows path and pin data. For each path, the table displays the path name, total delay time, relative pin delay contributions, and full path name. For each pin on each path, the table displays the pin name, individual delay time, relative contribution, clock edge (rising or falling), and full pin name. The combined delays for each path and the relative delay contribution for each pin appear graphically in bar graphs that represent the percentages of the total path delay.

Schematic and Symbol Views

Schematic views show graphical representations of the logic design. Symbol views provide graphic overviews of logic designs.

Design Vision provides the following types of schematic views:

- Path schematics that display individual timing paths or selected design logic (cells, ports, or nets) in a flat, single-sheet schematic

  A path schematic can contain cells, pins, ports, and nets. Hierarchy crossings indicate places where the path traverses the design hierarchy. You can display an InfoTip showing information about an object by holding the pointer over the object. The type of information displayed depends on the object type.

You can generate a path schematic to display timing paths, selected logic, or the fanin or fanout logic for selected objects. You can display or hide logic in a path schematic. You can also add timing paths, add fanin or fanout logic levels, or remove selected logic.

- Design schematics that display a design as a schematic composed of cell instances, pins, nets, and ports

  A design schematic shows both leaf-level logic (gates) and subdesigns (blocks). You can move up or down in the design hierarchy to view the schematic for each block in the hierarchy.

- DRC violation schematics (available only in DB mode) that display timing paths or individual cells with unified DRC rule violations in a flat, single-sheet schematic

Symbol views provide graphic overviews of logic designs.

You can use schematic and symbol views to visually analyze timing and logic in the optimized design and to gather information that can help you guide later optimization operations. Objects you select in a schematic or symbol view are cross-selected in other views. This capability allows you to efficiently analyze the logic and timing aspects of your design.

DRC Violation Browser and RTL Source Browser

If you are using DFT Compiler in DB mode, you can use the DRC violation browser to search for and view violation messages in an RTL TestDRC report or a Unified DRC report. The violation browser groups the warning and error messages into categories that help you find the problems you are interested in.

When you select a violation message in an RTL TestDRC report, you can view the source HDL code in the RTL source browser. Use the RTL source browser to view the causes of RTL design rule violations in your HDL source files.

When you select a violation message in a Unified DRC report, you can investigate the violation in a DRC violation schematic. Use this schematic to view the simulation values for the pins or paths where the violations occur and trace them through the design.

DFT Hold Time Analysis Window

If you are using DFT Compiler, you can use the hold time analysis window to view information about scan cells that have hold time violations. You can also select scan cells in the window for further examination with other analysis tools.

The hold time analysis window contains a scan cell table that displays a list of scan cells that have hold time violations. The table columns show the scan cell names and other details about each cell. A button bar below the table lets you load a different list of scan cells, save the scan cell details in a text file, customize the table columns, and access other analysis tools.

Report Views

Report views display design and timing reports. You can use a report view to view report information and select reported objects. You can also

- Save or append the report in a text file
- Display a report previously saved in a text file

In reports that list object names (cell, net, port, and worst-path timing reports), you can select an object by clicking its name (blue text) in the report view. When you select an object in a report view, Design Vision displays the design schematic for the design in which the object is located and magnifies the schematic to fit the selected object in the view. The name of the selected object also appears in the selection list, and the object is displayed in the selection color (white) in all open schematic views.

Design and Object Lists

You can generate a list of objects (designs, cells, nets, or ports and pins) and display information about them in a list view. You can display information about

- The designs loaded in memory
- Selected objects (such as selected cells)
- Objects related to other selected objects (such as pins of selected cells)
- Objects with a common function or attribute (such as fixed cells)

You can select some or all of the objects in a list by clicking or dragging the pointer across their names in the list view. (You can also use Shift-click or Control-click to select multiple names.) You can also

- Sort the information alphabetically by clicking a column heading. Click the heading again to reverse the sort.
- Scroll up and down in the table by pressing the Up Arrow and Down Arrow keys.

- Filter the objects listed in the table, limiting it to objects based on a character string or regular expression that you define.

If some of the text in a column is missing because the column is too narrow, you can hold the pointer over it to display the information in an InfoTip. You can also adjust the width of a column by dragging its right edge left or right.

Properties Dialog Box

You can use the Properties dialog box to view object properties for selected designs, design objects (cells, pins, ports, nets, and buses), or timing paths. You can also set, change, or remove values for certain properties.

The dialog box lists the object properties in a table with two columns (for property names and property values) and a row for each property. The properties you can view include object names, attribute values, and certain timing values. The list of properties differs depending on the type of object you select.

Note:

> Timing attribute values do not appear until you perform an operation that updates timing information (such as generating a timing report or opening a histogram).

When you select multiple objects, the property lists are displayed separately for each object. You can click the previous and next arrow buttons to navigate from one list of object properties to another. Alternatively, you can select an option to list the property values for all the selected objects together in a single table.

Selection List Dialog Box

The Selection List dialog box displays a list of the names and object types of all selected objects in the current design. When you select objects in other views, their names automatically appear in the selection list.

You can deselect object names and remove their names from the list. You can also filter the selection list, limiting it to information based on a character string or regular expression that you define.

Changing the Appearance of Schematic and Symbol Views

For display purposes only, when Design Vision generates a schematic, it applies the same display characteristics (visibility, line color, line style, and line width) to all the objects of a given object type: cell, port, pin, net, bus, bus ripper, or hierarchy crossing. This convention allows you to define how similar objects are displayed in a schematic or whether they appear at all.

You can change schematic settings in either of the following ways:

- Choose View > Preferences to open the Application Preferences dialog box, select the Schematic Settings category, set the desired options, and click OK.
- Edit the .synopsys_dv_prefs.tcl file and run it as a Tcl script.

Using Scripts

Designers often use scripts to accomplish routine repetitive tasks such as setting constraints or defining other design attributes. You can use your existing scripts in Design Vision. You can define your scripts in your setup files or in separate script files.

You can run dctcl scripts, in either DB mode or XG mode. If you have legacy dcsh scripts that you need to run, you must start Design Vision in DB mode with dcsh mode emulation. For details, see “Using dcsh Mode Emulation” in the Design Vision online Help.

To run scripts in Design Vision,

- Choose File > Execute Scripts.

  The Execute File dialog box opens. Use the dialog box to navigate to the appropriate directory and run your script.

Alternatively, you can run scripts from the design_vision command line by using the `source` command (for dctcl scripts) or the `include` command (for dcsh scripts in dcsh mode emulation).

Table 2-2 is a short list of some commands that are useful when you are writing scripts for use with Design Vision.

Table 2-2 Commands for Scripting

Command	Type	Description
`get_selection`	dctcl	Returns a collection that is currently selected in the GUI
`change_selection`	dctcl	Changes the selection in the GUI to the collection passed to it as a parameter
`gui_create_schematic`	dctcl	Generates a design schematic or a symbol view of the selected design or cell

For detailed descriptions of commands and options, see the man page for each command.

Viewing Man Pages

Design Vision provides an HTML-based browser window that lets you view, search, and print man pages for commands, variables, and error messages. You can use it to

- Display a man page
- Search for text on the man page you are viewing
- Print the man page you are viewing
- Browse back and forth between man pages you have already viewed

To view a man page in the man page viewer,

1. Choose Help > Man Pages.

   The man page viewer appears. The home page displays a list of links for the different man page categories.

2. Click the category link for the type of man page you want to view: Commands, Variables, or Messages.

   The contents page for the category displays a list of title links for the man pages in that category.

3. Click the title link for the man page you want to view.

Note:

> You can also display man pages in the man page viewer by using the `man` command (or the `gui_show_man_page` command) on the console command line.

You can browse back and forth between pages you previously viewed the same way you browse Web pages in a Web browser. For more information about viewing man pages, see online Help.

3

Performing Basic Tasks

This chapter describes how to perform basic presynthesis and synthesis tasks using Design Vision menus. This chapter is useful for experienced users of Design Compiler who need to accomplish familiar basic tasks in Design Compiler using Design Vision windows and menus.

If you are new to Synopsys synthesis tools, see the *Design Compiler Tutorial Using Design Vision* to learn about basic tasks by performing them on a simple design.

This chapter contains the following sections:

- Specifying Libraries
- Reading In Your Design
- Setting the Current Design
- Defining the Design Environment

- Setting Design Constraints
- Compiling the Design
- Saving the Design Database
- Working With Reports
- Printing Schematic and Symbol Views

Specifying Libraries

Before you start work on a design, specify the location of your libraries. You can define your library locations directly in the .synopsys_dc.setup file or indirectly by entering the locations in the Application Setup dialog box. Either method is acceptable—they both accomplish the same thing. (You can also specify library locations by running a script when you start Design Vision or by using the Execute Script dialog box. For details, see "Starting and Exiting the Graphical User Interface" in Chapter 2 and "Using Scripts" in Chapter 2.)

The link and target libraries are technology libraries that define the semiconductor vendor's set of cells and related information, such as cell names, cell pin names, delay arcs, pin loading, design rules, and operating conditions.

The symbol library defines the symbols for schematic viewing of the design.

You must specify any additional, specially licensed, DesignWare libraries with the `synthetic_library` variable (you do not need to specify the standard DesignWare library).

To define libraries in the .synopsys_dc.setup file, see the *Design Compiler User Guide*.

To specify the library location in the Application Setup dialog box,

1. Select File > Setup.

   The Application Setup dialog box opens.

2. Select the Defaults category if the Defaults page is not displayed.

3. Enter the appropriate path in the Search Path box.
4. Enter the library file names for the link, target, and symbol libraries you need to use.
5. (Optional) Enter the library file names for any specially licensed Synopsys or third-party DesignWare libraries you need to use.

   Note:

   > Synopsys provides a standard DesignWare library with components that implement many of the built-in HDL operators. You do not need to specify the standard DesignWare library.

6. Click OK.

For more information about the options on the Defaults page, see the “Setting Library Locations” topic in the online Help. For information about the options on the Variables page, see the “Setting Variables” topic.

To understand more about the function of link libraries, target libraries, symbol libraries, and DesignWare libraries, see the *Design Compiler User Guide*.

Reading In Your Design

To begin working on your design, read the design from disk into the tool’s active memory. This is where all changes in the design take place before you save the design by writing it back to disk.

The File menu contains the commands for reading in a design:

- Analyze and Elaborate

Use Analyze and Elaborate to read HDL designs and convert them to Synopsys database (.db) format. These commands open dialog boxes in which you can set options that are equivalent to the `analyze` and `elaborate` command-line options. For details, see the “Analyzing Files” and “Elaborating a Design” topics in online Help.

- Read

  Use Read (`read_file` is the command-line equivalent) to read designs that are already in .db format. This command opens a dialog box in which you can set options that are equivalent to the `read_file` command-line options. For details, see “Reading in a Design” topic in online Help.

The Analyze command checks the HDL designs for proper syntax and synthesizable logic, translates the design files into an intermediate format, and stores the intermediate files in the directory you specify.

The Elaborate command first checks the intermediate format files before building a.db design. During this process, Elaborate determines whether it has the necessary synthetic operators to replace the HDL operators, and it also determines correct bus size.

If you use Read to read in HDL files, the Analyze and Elaborate read functions are combined; however, Read does not do certain design checks that Analyze and Elaborate do.

Setting the Current Design

When you start a Design Vision session and read a design, the current design is automatically set to the top-level design. Some commands require you to set the current design to a subdesign before you issue them (the man pages provide such information).

To set the current design,

1. Click the drop-down list on the Design List toolbar to display the design names.
2. Select a design name.

Alternatively, you can open a design list view (by choosing List > Designs View), select a design name in the list, right-click, and choose Set Current Design. The command-line equivalent is `set current_design`.

Defining the Design Environment

Design Compiler requires that you model the environment of the design to be synthesized. This model comprises the external operating conditions (manufacturing process, temperature, and voltage), loads, drive characteristics, fanout loads, and wire loads. It directly influences design synthesis and optimization results.

Defining the design environment can involve using a large number of commands. Many designers find it convenient to define the design environment by using the default target technology library settings and by running scripts to define differences or additions. To define the design environment by using Design Vision menus, choose commands on the Operating Environment submenu under the Attributes menu. Online Help has more information about particular commands and submenus under the Attributes menu.

Setting Design Constraints

Setting design constraints can involve using a large number of commands. Most designers find it convenient to use scripts to set design constraints.

Design Compiler uses design rule and optimization constraints to control the synthesis of the design.

Setting Design Rule Constraints

Design rules are provided in the vendor technology library to ensure that the product meets specifications and works as intended. Typical design rules constrain transition times, fanout loads, and capacitances. These rules specify technology requirements that you cannot violate. (You can, however, specify stricter constraints.)

To set design rule constraints for the current design by using the GUI,

- Choose Attributes > Optimization Constraints > Design Constraints, set options as needed, and click OK.

To set design rule constraints for certain input ports by using the GUI,

- Select the ports, choose Attributes > Optimization Directives > Input Port, set options as needed, and click OK.

Setting Optimization Constraints

Optimization constraints define the design goals for timing (clocks, clock skews, input delays, and output delays) and area (maximum area). During optimization, Design Compiler attempts to meet these goals; however, it does not violate your design rules. To optimize a design correctly, you must set realistic optimization constraints.

To set optimization constraints by using the GUI,

- Click Attributes in the menu bar to open the Attributes menu.

  Choose Specify Clock to set clock periods and waveforms. Other optimization constraints and settings are in the submenus under the Attributes menu: Operating Environment (input and output delays), Optimization Constraints (maximum and minimum delays and maximum area), and Optimization Directives (design attributes, object attributes, and timing exceptions). Explore these submenus to find the settings you need.

For more information about menu items in the Attributes menu, see the “Attributes Menu” topic in the online Help.

Compiling the Design

Use Design Vision to initiate Design Compiler synthesis and optimization, thus compiling your high-level design description to your target technology.

Design Vision supports standard synthesis methodology: either a top-down compile or a bottom-up compile. For more information about compile methodologies, see the *Design Compiler User Guide*.

To compile the current design,

1. Choose Design > Compile Design.

   The Compile Design dialog box opens.

2. Select or deselect options as you require.

   For details about using the Compile Design dialog box, see the “Optimizing the Design” topic in online Help. Use the default settings for your first-pass compile. For most designs, the default settings provide good initial results. For more information about compile options, see the “Optimizing the Design” topic in online Help, the *Design Compiler User Guide* and the *Design Compiler Reference Manual: Optimization and Timing Analysis*.

3. Click OK to begin compiling.

After compiling the design, save the design as described in the next section.

The command log file records your Design Vision session. This log file is a record of all commands invoked. It includes commands you enter on the command line, commands initiated by your menu and dialog box selections, and tool commands (such as initialization commands and commands needed to execute user-entered commands).

You can use the command.log file to create a script file. Copy the file and use a text editor to add or remove commands as necessary.

Saving the Design Database

Design Vision does not automatically save designs before exiting.

To save the current design and each of its subdesigns in separate Synopsys database (.db) format files named *design_name*.db, where *design_name* is the name of the design,

- Choose File > Save.

To save the current design and all of its subdesigns in a single file with a different file name or file format,

- Choose File > Save As, enter or select a file name, select a file format, and click OK.

For more details, see the “Saving the Design” topic in online Help.

Working With Reports

Textual reports are available from the Design menu and the Timing menu. Use commands in the Design menu to generate design information and design object reports. Use commands in the Timing menu to generate timing and constraint reports. If these menus do not have the report you need, you can generate any Design Compiler report by issuing dc_shell report commands on the command line.

You can generate reports and display them in the report view (the default), save them in text files, or both. In a report view, you can select objects, save the report in a text file, or load a report from a file. In a report that contains object names (such as a design, cell, port, or net report), if you select an object name (blue text) in the report view, Design Vision cross-selects the object in a design schematic and magnifies the schematic to fit the object in the schematic view. For more information, see the “Viewing Reports” topic in online Help.

Generating Reports for Selected Objects

You can generate reports for selected objects.

To generate a report for a particular design object or group of design objects,

1. Select an object.
2. Choose a report command in the Design menu to open the associated dialog box.

   You can generate reports for cells, ports, and nets.
3. Click Selection in the report dialog box.

   The names of the selected objects appear.
4. Set other options as needed.
5. Click OK.

Printing Schematic and Symbol Views

You can print the path schematic or design schematic displayed in the active schematic view or the symbol diagram displayed in the active symbol view. Be sure that a default printer is set in your .cshrc file.

To print the active schematic or symbol view,

1. Generate a path schematic, design schematic, or symbol view.
2. Make sure the view you want to print is the active view.

   Click the corresponding tab at the bottom of the workspace if you need to make the view active.
3. Choose File > Print Schematic.

   The Select Printer Settings dialog box opens.
4. Select the print options you require and click OK.

You can also save an image of the schematic or symbol view in a PostScript file for printing later from a UNIX or Linux shell. For details about the options in the Print dialog box, see the “Printing Schematic and Symbol Views” topic in online Help.

4

Solving Timing Problems

This chapter presents procedures and suggestions for solving timing problems by using the features of Design Vision. The chapter does not provide details about exercising particular features of Design Vision, such as how to create a histogram or how to create a path schematic. For detailed information about Design Vision features, see the Design Vision online Help system.

This chapter contains the following sections:

- Before You Analyze
- Creating a Timing Overview
- Choosing a Strategy for Timing Closure

Before You Analyze

Before you analyze your design with Design Vision, follow your normal compile methodology to create a constrained gate-level design. A constrained gate-level design is a prerequisite to any timing analysis.

For more information about using Design Vision to create a gate-level design, see Chapter 3, “Performing Basic Tasks.”

Creating a Timing Overview

Creating an overview of the timing of your design is a valuable way to start any analysis of your design’s timing problems. A timing overview can help you decide what strategy to follow in gaining timing closure.

For example, a timing overview can help answer such questions as

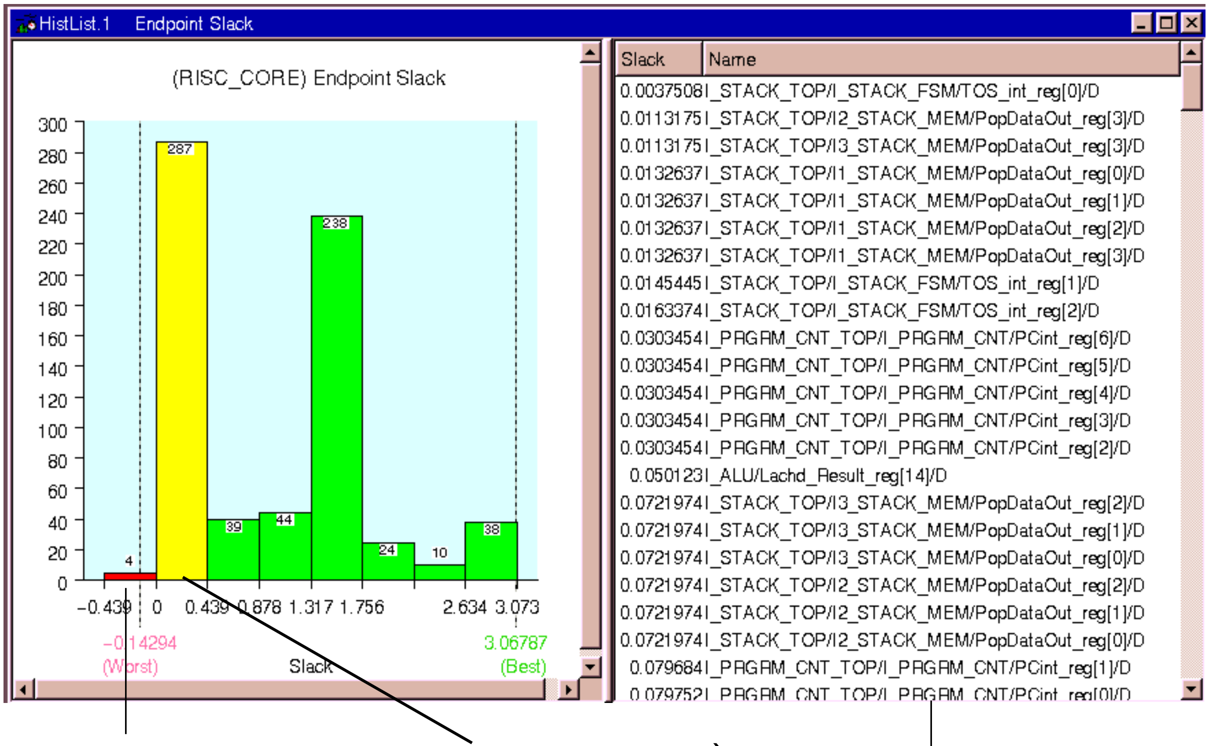
- Do I have many failing paths or just a few?
- Can I apply a local strategy for gaining timing closure?
- Do I need a global strategy for gaining timing closure?

To create a timing overview of your design,

1. Start with a constrained gate-level design.
2. Generate an endpoint slack histogram.

Figure 4-1 is a typical endpoint slack histogram for a design with a 4-ns clock cycle.

Figure 4-1 Endpoint Slack Histogram



Slack	Name
0.0037508	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[0]/D
0.0113175	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0113175	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0145445	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[1]/D
0.0163374	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[5]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[4]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[3]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[2]/D
0.050123	I_ALU/Lachd_Result_reg[14]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D
0.079684	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[1]/D
0.079752	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[0]/D

Endpoints of
failing paths (red)

Selected bin (yellow)

Endpoint names for
selected bin

Using information such as that in Figure 4-1, you might decide on a local strategy if just a few paths are failing by a small margin (failing path endpoints are in one or more red bins to the left of 0 on the horizontal axis). Conversely, if you find that many paths are failing, or that the design is failing your timing goals by a large margin, you might choose a higher-level, or global, strategy for problem solving.

Choosing a Strategy for Timing Closure

There is no single strategy that ensures quick and easy timing closure; however, a strategy based on the size and number of timing violations can be useful.

Assessing the Relative Size of Your Timing Violations

This section suggests guidelines for describing the relative size of timing violations in your design. After you create an endpoint slack histogram, you can use these size guidelines to help you judge what strategy to use for timing closure.

What you consider to be small or large violations depends on the requirements of your design and your design process; however, assessing violation size as a percentage of clock cycle can be useful.

- Small violations

  Some designers consider small violations to be about 10 percent of the clock cycle or less.

- Large violations

  Some designers consider large violations to be about 20 percent of the clock cycle or greater.

- Medium violations

  Medium-size timing failures fall between the limits you set for large and small failures in your design or design process.

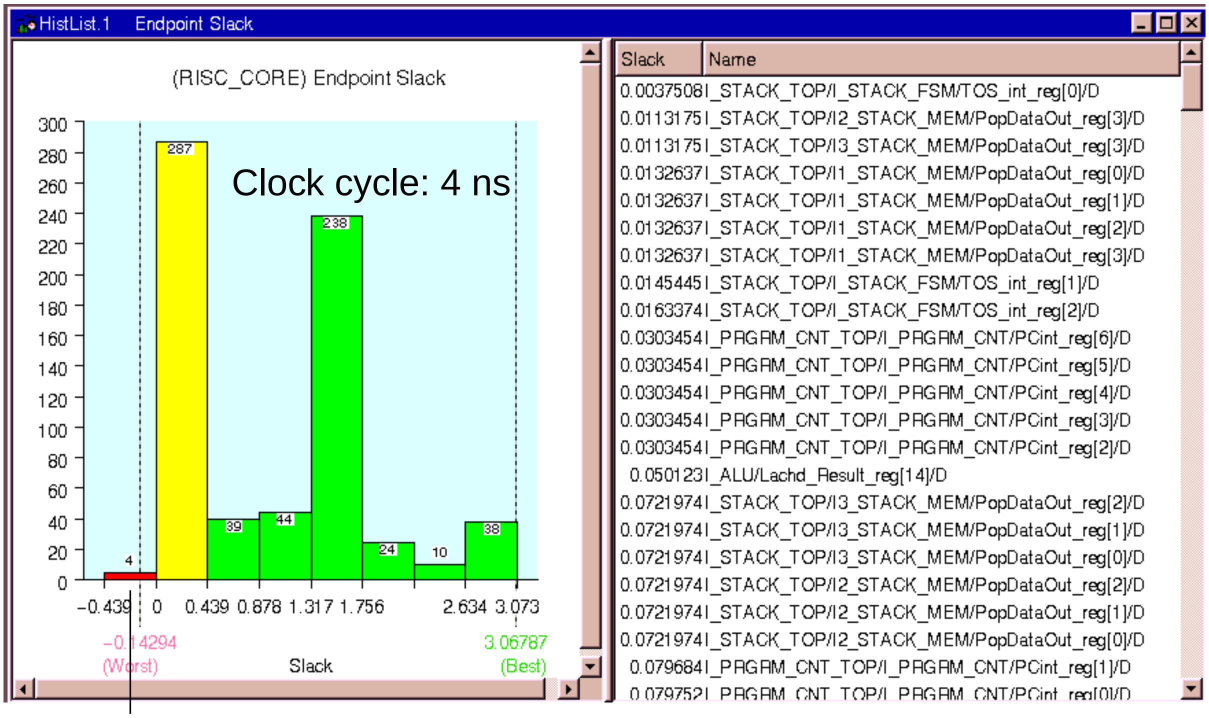
Whether your design is failing timing goals by large or small margins, the best strategy for timing closure is one that uses the least amount of runtime or number of design iterations to achieve timing goals.

This principle underlies the methodology suggestions in this chapter. For more information about creating a timing overview, see “Creating a Timing Overview” on page 4-2.

When Timing Violations Are Small

Figure 4-2 is a histogram of a design that is failing timing goals by a small margin. For example, no path is failing by more than 0.14 ns—that is, less than 10 percent of the 4-ns clock cycle (ignoring input and output delay). You can click any bin to see the endpoint names for the paths in the bin.

Figure 4-2 Design With Small Timing Violations



Slack	Name
0.0037508	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[0]/D
0.0113175	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0113175	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0132637	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0145445	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[1]/D
0.0163374	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[5]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[4]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[3]/D
0.0303454	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[2]/D
0.050123	I_ALU/Lachd_Result_reg[14]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0721974	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D
0.079684	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[1]/D
0.079752	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[0]/D

Endpoints of failing paths (red)

Designs that fail by a small margin can have many failing paths or just a few. The endpoint slack histogram helps you to recognize quickly which case you have. Whether you have just a few failing paths or many, you can follow a global or local strategy in fixing the violations.

If suggestions for fixing small violations (either globally or locally) don’t meet your timing goals, try applying the suggestions in “When Timing Violations Are Medium Size” on page 4-8 or “When Timing Violations Are Large” on page 4-11.

Working Globally to Fix Small Violations

To apply a global methodology for fixing small violations, consider recompiling your design using the incremental option and a higher map effort. The incremental option saves runtime by using the current netlist as the startpoint for design improvements.

The incremental compile with higher map effort has the advantage of simplicity—that is, it requires little or no time spent in analyzing the source of timing problems. However, this method can change much of the logic in the design.

Working Locally to Fix Small Violations

If you have a small number of paths with small violations, or if your violations seem to come from a limited set of problems on a few paths, a local strategy can be effective.

To use a local strategy for fixing small violations,

- Check hierarchy on failing paths

Design Compiler does not optimize across hierarchical boundaries. Thus, snake paths limit Design Compiler's ability to solve timing problems on such paths.

- Look for excessive fanout on failing paths

  Because higher fanout causes higher transition times, excessive fanout can worsen negative slack on failing paths.

To check for hierarchy problems on failing paths,

1. Generate an endpoint slack histogram.
2. Click a bin that contains a failing path.

   A list of endpoints for failing paths is displayed.
3. Select the endpoint for the path you are interested in.
4. Generate a path schematic to see which leaf cells are in which levels of hierarchy.

   If your critical path, for example, crosses multiple subblocks of a level of hierarchy, consider ungrouping these subblocks. Design Compiler does not optimize across hierarchy boundaries. Thus, a subsequent compile has further opportunity to optimize the critical path when you ungroup such blocks.

To look for excessive fanout on failing paths,

1. Generate an endpoint slack histogram.
2. Select the endpoints for failing paths.

   Select the failing bin to see the endpoints.

3. Generate a timing report with the following options:

   - net

   - trans

   Send the report output to the report view. For more information on report generation, see “Working With Reports” on page 3-10 and the “Viewing Reports” topic in online Help.

4. Examine the report for pins with high transition times and nets with high fanout.

   Such paths are candidates for buffering or drive-cell resizing.

5. Create path schematics of any paths you would like to see.

   A path schematic provides contextual information and details about the path and its components. Such information is often a prerequisite to understanding problems on the path.

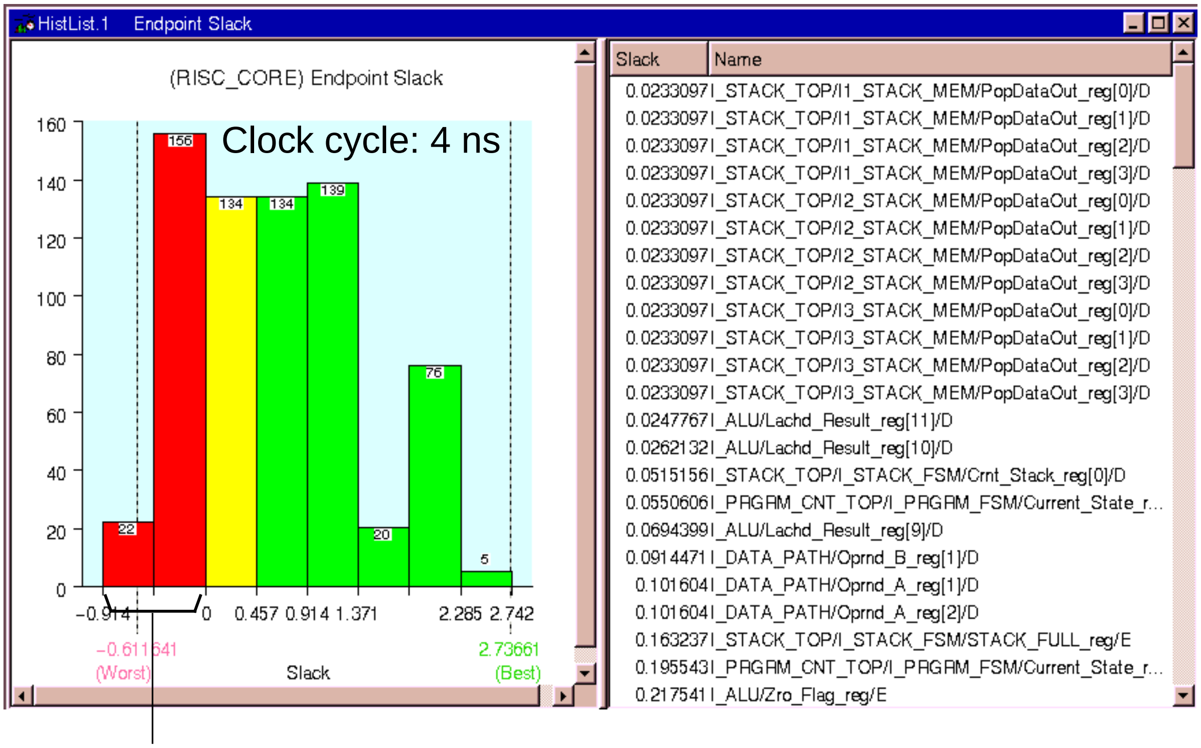
6. View fanin and fanout for path schematics.

   This step can provide useful information about the logic that drives, or is driven by, the problem path. For example, after viewing fanin or fanout, you might choose to resize cells in those logic cones.

When Timing Violations Are Medium Size

Figure 4-3 is a histogram of a design that is failing timing goals by margins that are between the large and small limits that are appropriate to your design methodology (for example, between 10 and 20 percent of the clock cycle). You can click a bin to see the endpoint names for paths the bin contains. A bin is yellow when selected. In Figure 4-3, one of the four bins containing endpoints of failing paths is selected.

Figure 4-3 Design With Medium Timing Violations



Slack	Name
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I1_STACK_MEM/PopDataOut_reg[3]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I2_STACK_MEM/PopDataOut_reg[3]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[0]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[1]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[2]/D
0.0233097	I_STACK_TOP/I3_STACK_MEM/PopDataOut_reg[3]/D
0.0247767	I_ALU/Lachd_Result_reg[11]/D
0.0262132	I_ALU/Lachd_Result_reg[10]/D
0.0515156	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[0]/D
0.0550606	I_PRGRM_CNT_TOP/I_PRGRM_FSM/Current_State_r...
0.0694399	I_ALU/Lachd_Result_reg[9]/D
0.0914471	I_DATA_PATH/Oprnd_B_reg[1]/D
0.101604	I_DATA_PATH/Oprnd_A_reg[1]/D
0.101604	I_DATA_PATH/Oprnd_A_reg[2]/D
0.163237	I_STACK_TOP/I_STACK_FSM/STACK_FULL_reg/E
0.195543	I_PRGRM_CNT_TOP/I_PRGRM_FSM/Current_State_r...
0.217541	I_ALU/Zro_Flag_reg/E

Endpoints of failing paths

When negative slack values are medium, you can use Design Vision to investigate further and focus your recompile on a critical range of negative slack values for path groups. Focusing your compile effort on a critical range can improve worst negative slack and total negative slack.

Defining a critical range for path groups offers the advantage of concentrating compile effort and runtime on those areas that most need it.

To investigate and focus a recompile by defining a critical negative slack range for path groups,

1. Create a path slack histogram for each path group in your design.

   Start with an arbitrary value of 1000 for the number of paths to include in each histogram. Raise or lower this value depending on the number of failing paths. The goal is to choose a value that shows you all or nearly all of the failing paths.

2. Decide on a critical range for each path group (note the values for use in step 3).

   When deciding on a critical range, choose a range that allows Design Compiler to focus on the worst endpoint violations without too large an increase in runtime. For example, some designers apply one of the following guidelines to decide on a critical range:

   - Use a range that includes the worst 50 paths in a group.
   - Use a range equal to one generic cell delay in your technology.

   These are rough guidelines; for subsequent compiles you can adjust your critical range as necessary.

3. Set a critical range for each path group.

   Using the values you decided on in step 2, set the critical ranges for each path group with the `group_path` command. For example,

```
design_vision-t> group_path -name my_clock -critical_range 0.25
```

4. Recompile the design.

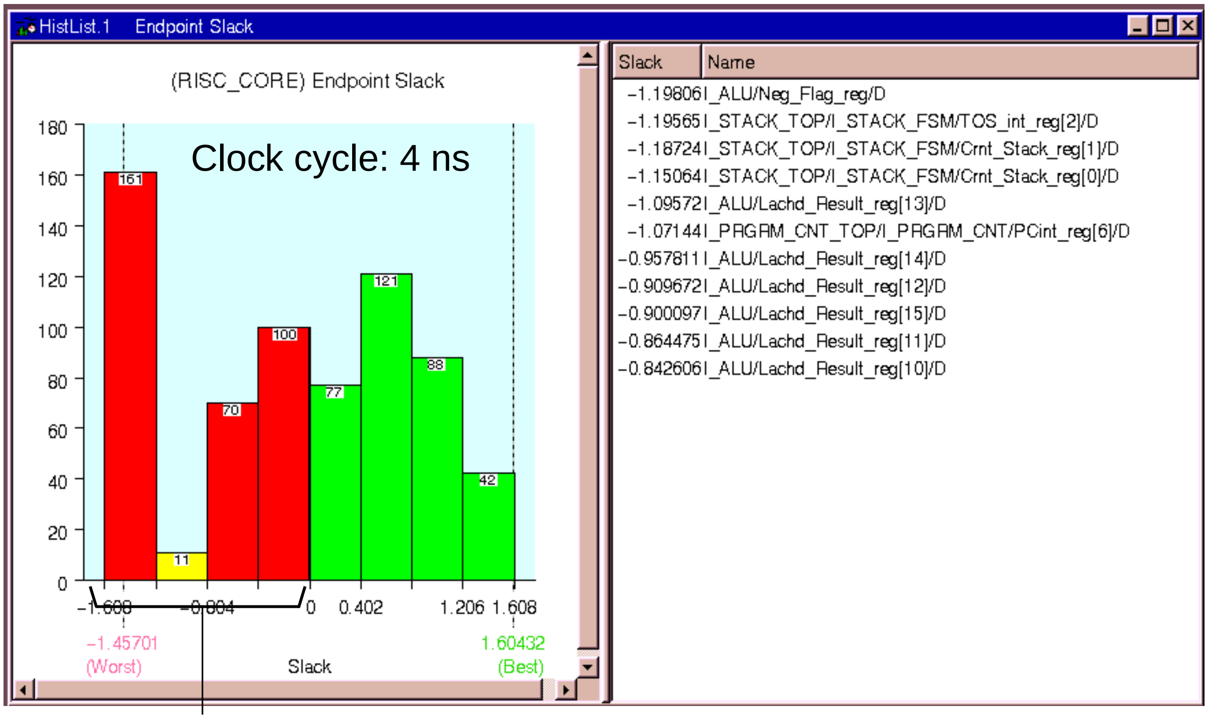
   With a critical range defined, the compile effort is now focused. However, you can also choose to increase the compile effort over your previous compile. In the Compile Design dialog box, select medium or high effort. Select the “Incremental mapping” option to direct Design Compiler to use the current netlist as a starting point for design improvements. An incremental compile can save runtime when your design does not require fundamental changes.

If suggestions in this section don’t meet your timing goals, try applying the suggestions in “When Timing Violations Are Large” on page 4-11.

When Timing Violations Are Large

Figure 4-4 shows a design that is failing timing goals by a large margin. You can click a bin to see the endpoint names for the paths it contains.

Figure 4-4 Design With Large Timing Violations



Slack	Name
−1.19806	I_ALU/Neg_Flag_reg/D
−1.19565	I_STACK_TOP/I_STACK_FSM/TOS_int_reg[2]/D
−1.18724	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[1]/D
−1.15064	I_STACK_TOP/I_STACK_FSM/Crnt_Stack_reg[0]/D
−1.09572	I_ALU/Lachd_Result_reg[13]/D
−1.07144	I_PRGRM_CNT_TOP/I_PRGRM_CNT/PCint_reg[6]/D
−0.957811	I_ALU/Lachd_Result_reg[14]/D
−0.909672	I_ALU/Lachd_Result_reg[12]/D
−0.900097	I_ALU/Lachd_Result_reg[15]/D
−0.864475	I_ALU/Lachd_Result_reg[11]/D
−0.842606	I_ALU/Lachd_Result_reg[10]/D

Endpoints of failing paths

Fixing large violations can require a high-level strategy to improve your design’s performance. To fix large timing problems, consider any of the following changes:

- Modify your constraints.

  For example, increase the clock cycle or adjust time budgeting for the block or chip.

- Change the target technology.

  For example, target a higher performance technology.

- Modify the RTL.

  For example, you can move late-arriving signals such that you minimize their path length.

Index