cadence®

Virtuoso Liberate LV Reference Manual

Product Version 16.1
January 2017

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

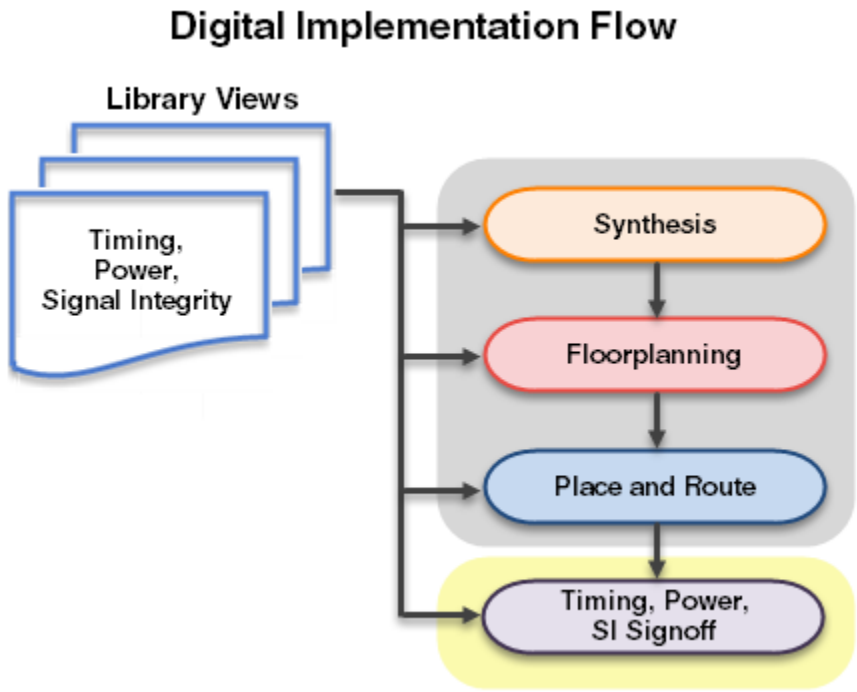
Contents

Preface

Introduction to Characterization

The Role and Importance of Libraries

Creation of electrical views is a pre-requisite for any digital design flow. The electrical information stored in the library views is used throughout design implementation from logic synthesis, through design optimization to final signoff verification. Accurate library view creation is essential to ensure close correlation between the design intent and the final silicon.

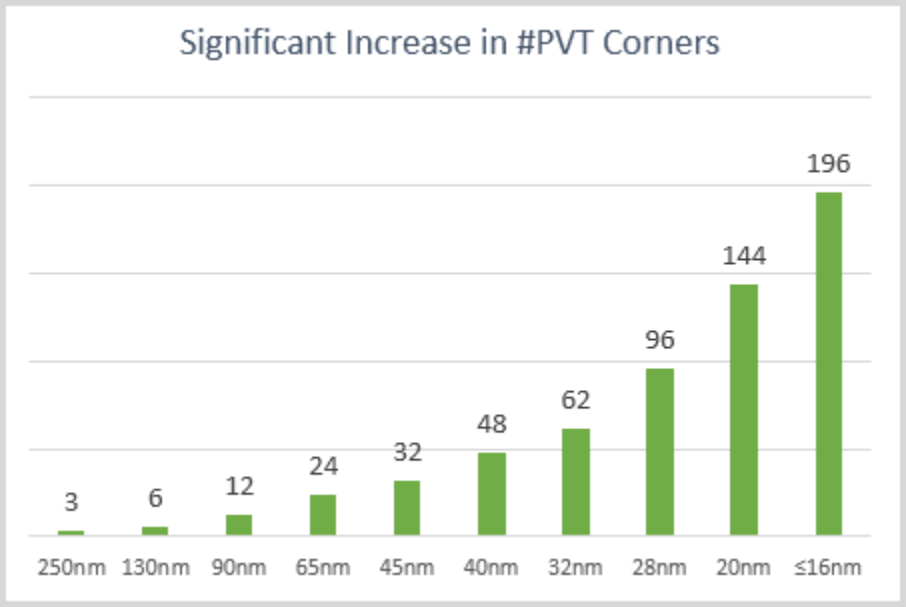


A Growing Problem

In nanometer geometries (65nm or below), the required number of library views is growing dramatically because of issues related to power leakage and process variation. To minimize

power leakage at deep submicron nodes, we see process variations such as LVT, RVT, and HVT (low/regular/high voltage) being utilized. For example, to manage power at 65nm, it is common to have library cells with two or three different threshold values (high threshold to reduce leakage power, lower thresholds to improve performance), and to use two or more on-chip supply voltages. In this scenario, the number of views needed for 65nm will be six times greater than what is needed for 130nm.

The figure below shows the growing trend that requires PVT corners to accurately model the circuit behavior:

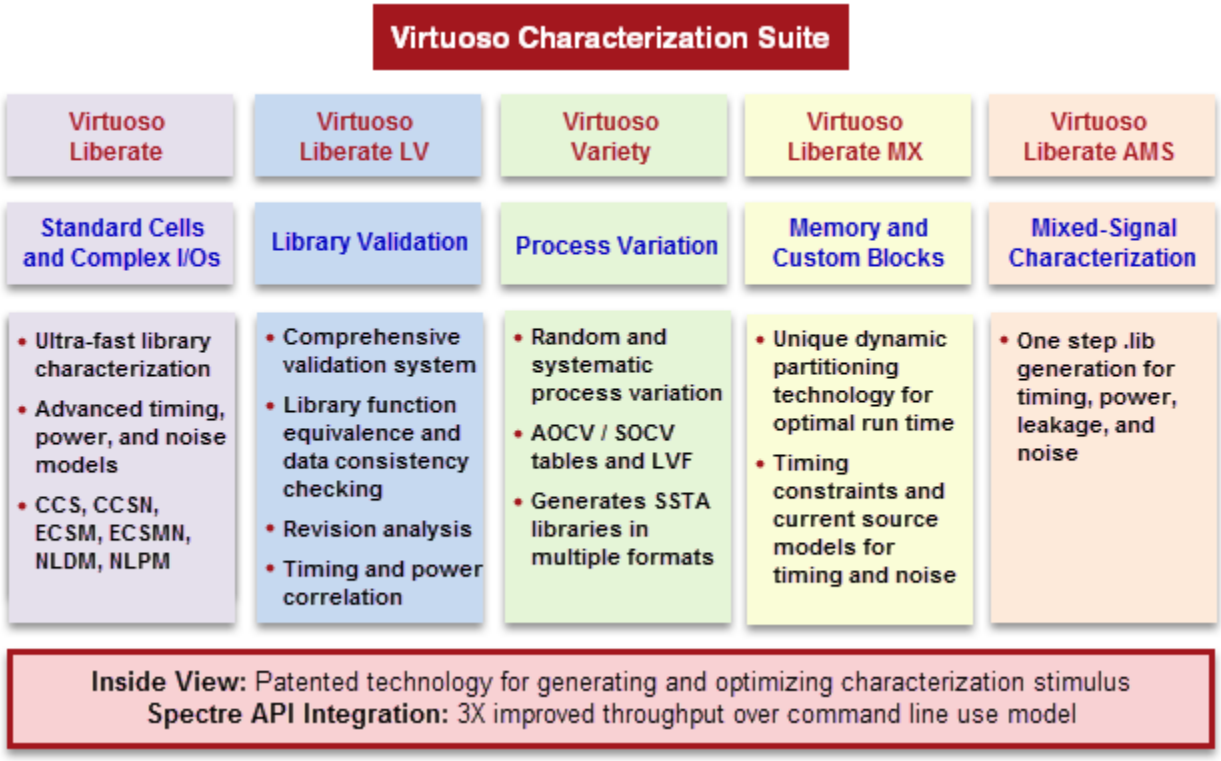


In addition, library views require more advanced models like:

- Current source models CCS and ECSM
- Statistical models – AOCV/SOCV/LVF
- Netlist extraction at various temperatures for Nanometer Process Nodes
- Support multiple foundries to assure flexibility for yield issues
- Support for many more functional designs – 1000+ STD cell, I/O, custom datapath, memory and Analog IP

Virtuoso Characterization Suite

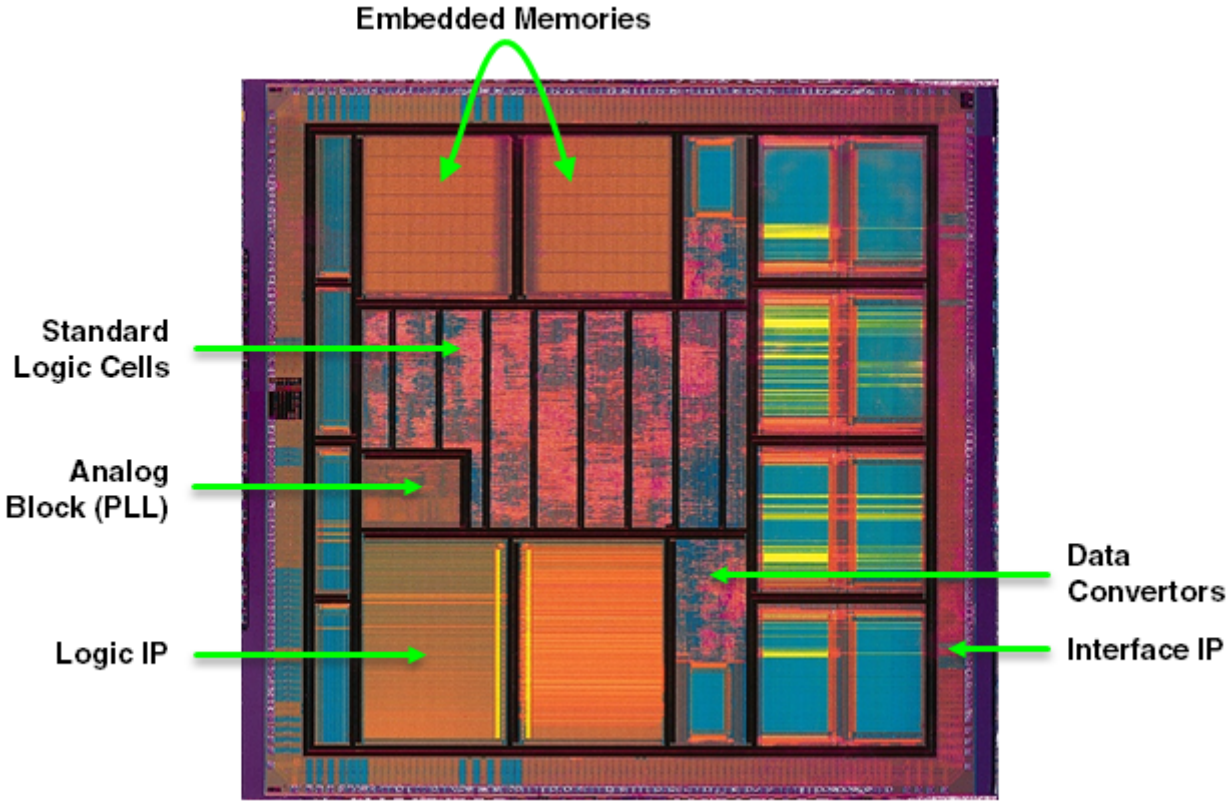
To address all the challenges, Cadence offers Virtuoso Characterization Suite that covers the complete portfolio of characterization solutions given below:



The Virtuoso Characterization Suite intends to provide highly efficient and automated electrical view creation and validation for all IP blocks that including the following:

- Logic and I/O cells (GPIO, PCI, SSTL, PECL, and so on)
- Embedded Memory (SRAM, ROM, Register files, CAM, and so on)
- Custom digital blocks (custom cells, datapath, cores, and so on)

- Interface IP and analog blocks (USB, Serdes, DDR, and so on)



System Requirements

Liberate, Variety, Liberate MX, Liberate LV, and Liberate AMS run exclusively on Linux operating system. The following table lists the supported platforms:

Architecture	Development OS	Supported Environments
x86_64 (32/64)	RHEL 5.5	RHEL 6
		SLES10
		SLES11

For detailed information about the requirements, see Computing Platforms.

Software and Licensing Requirements

- LIBERATE 15.1

  The following table lists the required server and client product numbers for each product in the Virtuoso Characterization Suite:

Product Name	Server Product Number	Client Product Number
Liberate	ALT110	ALT111
Variety	ALT210	ALT211
Liberate MX	ALT410	ALT411
Liberate LV	ALT610	ALT611
Liberate AMS	ALT810	ALT811 or ALT812

- MMSIM 14.1 or MMSIM 15.1

Product Name	Product Number
Spectre® XPS	91600 or 90004
Spectre® APS	3500 (restricted for characterization), 91050, or 90004

About This Manual

The *Virtuoso Liberate LV Reference Manual* describes the Cadence® Virtuoso® Liberate LV tool. The manual includes opening chapters that describe what Liberate LV does and how to get started with the tool. Later chapters discuss the commands and variables that can be used with Liberate LV.

Audience Profile

This manual is aimed at developers and designers who want to work on to validate and verify each library to ensure data consistency, accuracy and completeness. It assumes that you are familiar with:

- SPICE simulations

- Basic expected behavior of the design being used

Additional Documents for Reference

For information about known problems and solutions, see *Virtuoso Characterization Suite Known Problems and Solutions*.

For a list of new features in a release, see *Virtuoso Characterization Suite What's New*.

For information about other products in Virtuoso Characterization Suite, refer to the following manuals:

- *Virtuoso Liberate Reference Manual* describes the Liberate tool—an accurate, highly efficient and easy-to-use library characterizer that creates electrical views (timing, power, and signal integrity) in formats such as the Synopsys Liberty (`.lib`) format.
- *Virtuoso Variety Reference Manual* describes the Virtuoso Variety process variation cell characterizer—a tool that characterizes process variation aware timing models and generates libraries for multiple statistical static timing analyzers (SSTA) without requiring re-characterization for each unique format.
- *Virtuoso Liberate MX Reference Manual* describes Liberate MX—a tool that provides library creation capabilities to cover memory cores.
- *Virtuoso Liberate AMS Reference Manual* describes Liberate AMS—a tool that provides library creation capabilities for Analog Mixed Signal (AMS) macro blocks.
- *ALAPI Reference Manual* describes a Tcl interface that allows access to the Liberate characterized Library DataBase (LDB).

Rapid Adoption Kits

Cadence provides Rapid Adoption Kits that demonstrate how to use Virtuoso applications in your design flows. These kits contain design databases and instructions on how to run the design flow.

Typographic and Syntax Conventions

This section describes the typographic and syntax conventions used in this manual.

`literal`	Non-italic words indicate keywords that you must enter literally. These keywords represent command or variable names.
`argument`	Words in italics indicate text that you must replace with an appropriate value.
`< >`	Angle brackets indicate text that you must replace with a single appropriate value. When used with vertical bars, they enclose a list of choices from which you must choose one.
`|`	Vertical bars separate a choice of values. They take precedence over any other character.
`-`	Hyphens denote arguments of commands or variables. Usually arguments denoted in this way are optional but, as noted in the syntax, some are required. The hyphen is part of the name and must be included when the argument is used.
`{ }`	Braces indicate values that must be denoted as a list. When used with vertical bars, braces enclose a set of values from which you must choose one or more. When you specify a list, the values must be enclosed by either quotation marks or braces. For example, `{val1 val2 val3}` and `"val1 val2 val3"` are legal lists.

Some argument are positional and must be used in the order they are shown. Any positional arguments that are used must be given after any arguments denoted with hyphens.

Customer Support

For assistance with Cadence products:

- Contact Cadence Customer Support

  Cadence is committed to keeping your design teams productive by providing answers to technical questions and to any queries about the latest software updates and training needs. For more information, visit: https://www.cadence.com/support

- Log on to Cadence Online Support

Customers with a maintenance contract with Cadence can obtain the latest information about various tools at: https://support.cadence.com

Feedback about Documentation

You can contact Cadence Customer Support to open a service request if you:

- Find erroneous information in a product manual
- Cannot find in a product manual the information you are looking for
- Face an issue while accessing documentation by using Cadence Help

You can also submit feedback by using the following methods:

- In the Cadence Help window, click the *Feedback* button and follow instructions.
- On the Cadence Online Support Product Manuals page, select the required product and submit your feedback by using the *Provide Feedback* box.

1

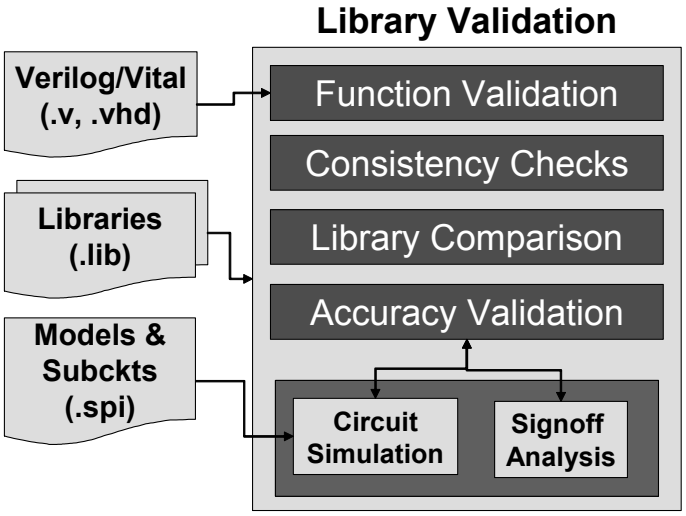
Introduction

This section gives an overview of the Liberate LV library validator.

Creation of library views is a complex process comprising many circuit simulations, data measurements and model transformations, typically distributed across a large computer network. In this process there are many opportunities to introduce errors due to network failures, simulation convergence problems, optimistic characterization assumptions, software version incompatibilities, measurement inaccuracies, and incorrect user inputs.

Because each set of libraries is used for multiple chip designs, it is paramount that the library data is complete and correct. Liberate LV provides a collection of capabilities to validate and verify each library to ensure data consistency, accuracy and completeness.

Figure 1-1 Liberate LV in the Design Flow



Function and State Coverage

When characterizing a cell library, the input directives, vectors, and stimuli might come from a previous library or be user-coded in the characterization tool's input language. However,

these vector and arc assumptions might be incomplete or inconsistent with the underlying transistor-level circuits that are extracted from the layout of each cell. Liberate LV is able to check all the function descriptions in the input library directly against the transistor-level circuit and report any differences, thus preventing potential functional errors later in the process, such as when the chip is being formally verified or tested after manufacturing.

Liberate LV provides the means to ensure that all the functional information stored in a library (`.lib`) (such as `function` and `next-state` attributes and `statetable` constructs) are consistent with the transistor-level SPICE subcircuits and the library's Verilog and Vital descriptions.

In addition, Liberate LV ensures that all the necessary timing, power, signal integrity, and leakage states are represented in the library and reports any missing states. Liberate LV also warns where insufficient distinct states exist, potentially causing inaccuracies in timing, power, and signal integrity analysis.

Consistency of Library Data

Liberate LV provides a number of data consistency checks such as comparing table-based delay data against current (CCS) data, checking for non-monotonic delays, or incorrect values such as negative values in rising current waveforms.

Liberate LV checks the consistency between the library (`.lib`), the library Verilog (`.v`) and the SDF generated by various timing tools to ensure that the SDF can be back-annotated to a logic simulator.

Library Revisions

Liberate LV provides the means to compare a new library against an existing library. It generates both graphical and text reports with the comparison data. Libraries can be compared with different indices, different function syntax, different states, and even different cell names. This allows verification of libraries created with different characterization systems or process models. Each new library release can be analyzed to identify significant changes in function, delay, power, leakage power, and noise immunity.

Accuracy of Library Data

To verify that the library data is accurate, Liberate LV performs a comparison of library models in the appropriate static timing analysis tool against results obtained from transistor-level circuit simulation. To ensure timing accuracy, Liberate LV invokes a static timing analyzer and

compares the resulting values against simulations of the test circuit using a SPICE simulator. Test circuits are automatically created for each check; for example, as a variable-length chain of cells with interconnect parasitics. For delay verification, every input-to-output arc for every input-slew and load condition can be verified. In addition, Liberate LV can measure the accuracy of constraint data, switching-power, and leakage. Liberate LV supports multiple Static Analysis tools for timing and power analysis, including Synopsys PrimeTime® SI, Cadence ETS, and numerous SPICE simulators such as Eldo®, HSPICE®, and Spectre®.

2

Getting Started

This chapter describes how to start using Liberate LV.

Before using Liberate LV, make sure that it is installed correctly and that all the necessary pre-requisite data is available. (Go to https://support.cadence.com for information about the InstallScape and license manager tools.)

Environment Variables

Path to Executable

Set the following environment variables to include Liberate LV in your executable path:

```
% setenv ALTOSHOME <install_dir>/<liberate_release_name>
% set path=($path $ALTOSHOME/bin)
```

Set the following to include integrated Spectre in your executable path:

```
% set path ($path $ALTOSHOME/tools.lnx86/spectre/bin)
```

64-bit Machine Support

Liberate LV also ships with support for 64-bit machines. To use the 64-bit version, set the ALTOS_64 environment variable before launching Liberate LV."

```
% setenv ALTOS_64 1
```

Environment Variables for Controlling Licensing Checks

- ALTOS_LIC_MAX_TIMEOUT
  `setenv ALTOS_LIC_MAX_TIMEOUT <value>`

  where;

  Value is time in seconds

This shell variable specifies how long Liberate (both Server and Client) will wait to obtain a license.

For a server process, if the ALTOS _QUEUE variable is enabled, Liberate will attempt to check out 1Server license. If the max timeout is reached, and no server license has been checked out, then Liberate will reset the timer and loop back to continue waiting for a license. For a client, when the max timeout is reached and at least one license was checked out, then the Liberate client will start to run with the licenses it has. No additional licenses are checked out.

- ALTOS_LIC_CHECK_ALT_TIMEOUT
  `setenv ALTOS_LIC_CHECK_ALT_TIMEOUT <value>`

  where;

  value is time in seconds

  Some Cadence characterization products can run using more than one product license. This variable controls both the server and client timeout before trying to check out an alternative license feature if there are any such licenses in the license pool.

Managing Licenses

Liberate LV uses a server/client licensing scheme. A server license is used for invoking and monitoring the validation run on the server machine while the client licenses are used for running simulations on the client machines. Each Liberate LV server can access all the available client licenses. For example, with two server licenses and forty client licenses the following configurations are all valid:

- A single characterization run using 40 client processes
- Two simultaneous characterization runs, each with 20 client processes
- Two simultaneous characterization runs, one with 30 client processes and one with 10 client processes

Note: It is important not to request more client licenses than are available because doing so can increase the wall-clock time required to complete the characterization.

Important

Ensure that the license daemon (`cdslmd`) and the license server (`lmgrd`) have the same version and that this version is the same as that required for a release. For example, v11.11.1 is required for the Liberate 15.1 release. If a mismatch is detected, unexpected license behavior might be observed. For example, the license search path can be reset to `<none>` after a failed license check out request. This

can result in incorrect license checking process.

On a 64-bit license host, the 64-bit `cdslmd` and `lmgrd` must be used instead of the default 32-bit ones.

Wait for Available License

When a Liberate LV job is submitted, a request is made for a license. If no license is available, Liberate LV terminates. To change this behavior so that Liberate LV waits until a license becomes available, set the following environment variable:

```
setenv ALTOS_QUEUE 1
```

Liberate LV clients run using different types of client license features, depending on the product names. Some product licenses can be mixed and matched together.

When a Liberate LV server starts, it checks out one client license, either `Liberate_LV_Client` or `Liberate_LX_Client` (if a bundled licence is available). Later, when simulations are ready to begin, Liberate LV checks out a server license, and attempts to check out N clients, where N is the number of threads specified by the `char_library -thread` argument. If Liberate LV

- Acquires all N licenses, it starts simulations using N threads.
- Acquires M licenses, 0 < M < N, and `ALTOS_QUEUE` is not set to 1, it starts M threads.
- Does not acquire a license and `ALTOS_QUEUE` is not set to 1, it quits.
- Acquires fewer than N licenses and `ALTOS_QUEUE` is set to 1, then it waits for up to the value of the `lic_max_timeout` variable, trying to get all N licenses. After `lic_max_timeout` is reached, if Liberate LV acquires M licenses and M > 0, then it starts M threads.

  If Liberate LV is unable to acquire a license, then it again waits for another `lic_max_timeout` seconds to acquire client licenses. This process is repeated until at least one client license can be checked out (because `ALTOS_QUEUE` is set to 1).

Invoking Liberate LV

Liberate LV utilizes stdout and stderr for all messages. By default, no log file is created. To invoke Liberate LV while creating a log file:

```
% liberate_lv  my.tcl  |&  tee  my.log
```

System Libraries

Liberate LV is shipped enabled with dynamically linked system libraries. To verify Liberate LV is capable of running on your system, try executing it. If Liberate LV fails to start properly, it may be possible that you have an old system and that there are missing or incorrect system libraries. If this occurs, and you have already checked your environment setup is correct, you can try using statically linked binaries by setting the following environment variable:

```
setenv ALTOS_USE_STATIC_BINARIES 1
```

Preparing For Validation

The following data is typically required to take advantage of the capabilities built into Liberate LV:

- SPICE level subcircuit descriptions for the cells to be validated

  For maximum accuracy, subcircuit descriptions should include layout parasitic elements.
- Foundry device models in the proper syntax for the target SPICE simulator
- A Liberate LV command file in Tcl format
- An existing library in Liberty® format

Extracted Cell Netlists

The transistors, diodes, resistors, capacitors, and extracted parasitic elements (RCs) comprising the cell are passed to Liberate LV in a SPICE format netlist. Extracted SPICE netlists can be created directly from the cell layout by device- and interconnect-parameter extraction tools. Standard SPICE, HSPICE, ELDO, and Spectre® netlist formats are supported. Multiple cells can be specified either in a single file or as a group of files. Each cell to be validated must have a `.subckt` definition in the files passed to Liberate LV.

Device Models

The device models represent the electrical parameters of the target process. The device models include models for transistors (P and N channel), diodes, capacitors, and resistors. Most device model files include different parameters for different process corners such as a typical, fast, and slow corners.

Tcl Command File

Liberate LV uses the Tcl scripting language to control the validation process. The Tcl script is used to specify the cell netlists, SPICE models, and library to be validated. In addition, the Tcl script defines the range of data that the validation is to be performed over, such as input-slew and output-loading conditions for timing validation.

A sample Tcl script for running Liberate LV is shown below. This script validates all the timing data from library `test.lib` using Eldo® and PrimeTime®.

```
# Validate the timing in library test.lib
set subckts {nand2x4.spi nor2x2.spi dffx1.spi}
validate_library -model models.spi -subckts $subckts \
-verbose -extsim eldo -timer primetime test.lib
```

Running Liberate LV

To perform validation, type `liberate_lv` followed by the Tcl command file. An example run of Liberate LV can be found at `$ALTOSHOME/examples`.

1. Create a Tcl command file called `val.tcl` with commands such as:

```
set rundir $env(PWD)
set_operating_condition -voltage 1.5 -temp 125
set cells {INVX1 NOR2X1 DFFX1}
set subckts {}
set csz [llength $cells]
for {set c 0} {$c < $csz} {incr c 1} {
  set cell [lindex $cells $c]
  lappend subckts spice/subckts/$cell.sp
}
set model $rundir/spice/include_SS.sp
validate_library \
    -model $model \
    -subckts $subckts example.lib
Then execute the Tcl file as follows:
```

```
% liberate_lv val.tcl |& tee val.log
```

2. Look into the `VAL` directory for report results.
3. Create the func.tcl file and add in it the command given below. The command compares the functions within the specified library with those in the specified Verilog netlist.

```
compare_function -verbose example.lib example2.lib
```

4. Then execute the Tcl file as follows:

```
% liberate_lv func.tcl |& tee func.log
```

5. View the results:

```
% vi example.v.cmp.txt
% vi example2.lib.cmp.txt
```

6. Create a command file to compare the data in the two libraries. For example, it might be called `comp.tcl` and contain:

```
compare_library -lcplot example.lib example2.lib
```

7. Then execute the Tcl file as follows:

```
% liberate_lv comp.tcl
% vi example2.cmp.txt
```

3

Parallel Processing

This chapter describes how to use Liberate LV across multiple CPUs.

To achieve good performance, Liberate LV can use multi-threading across all available CPUs. Furthermore, Liberate LV can use distributed processing with multi-threading across a network of machines. Parallel processing reduces the total turnaround time for characterization nearly linearly with the number of CPUs used.

Multi-threading

The simplest way to use parallel processing with Liberate LV is to use multiple threads on a single computer. Liberate LV automatically determines the optimal number of threads, based on the hardware characteristics of the available CPUs. The `-thread` argument to the `char_library` command can be used to increase or decrease the number of parallel processes that Liberate LV can use on a single machine.

Distributed Processing

Liberate LV partitions the characterization task into a group of related simulations (arc partitions) to be performed on each of the available CPUs. There are two forms of distributed processing:

- Set client (non-packet) mode

  This method pre-processes all cells in each client machine. This is suitable for a small number of (larger) cells due to its advanced load balancing capability.

- Packet mode

  This method pre-processes only the cells that are characterized on each client machine. This is more suitable for a large number of smaller cells.

Note: There are several variables that return strings (such as the machine name) that you can use to create unique directory names or commands:

`%B`	Bundle (packet) directory name
`%C`	Command name
`%L`	Packet log filename
`%M`	Machine name
`%N`	Client number
`%O`	List of options
`%P`	Liberate server process ID
`%S`	Server name
`%U`	User name

Set Client (non-packet) Mode

The `set_client` commands specify either the names, or number of client machines to be used. For each machine, a directory in which Liberate LV can temporarily store data must also be specified. If the `set_client -n` argument is used, Liberate LV submits the appropriate number of tasks to the named queuing system.

```
# Specify 10 client machines to use on lsf_queue
# Use /tmp/liberate_%N to store intermediate files
set_client -dir /tmp/liberate_%N -n 10 lsf_queue
```

The `rsh_cmd` variable (default `ssh`) can be used to specify the shell to use for starting remote jobs on a client machine. The `rcp_cmd` variable (default `scp`) can be used to set the command for copying files from the host to client machines. Before starting a parallel-processing job, make sure that the following commands can be performed without requiring any password or passphrase:

- `ssh` or `rsh` from the server to the client
- `scp` or `rcp` a file from the server to the client

The `rsh_cmd` string can reference the current client and the command to invoke Liberate LV by using `%M` (machine or queue name given to the `set_client` command) and `%C` (command). In addition, command line options that appear after the Liberate LV Tcl file name can be passed into the `rsh_cmd` string by using `%O` (options). These `%` overrides can be useful if your system is using load-balancing and scheduling software.

If using LSF to run remote jobs, use:

```
# Use LSF bsub to invoke jobs on remote clients
# %M is replaced with the queue name "lsf_queue"
set_client -dir /tmp/liberate_%N -n 10 lsf_queue
set_var rsh_cmd "bsub -q %M %C"
```

If using Sun Grid to run remote jobs, use:

```
# Use SunGrid qsub sub to invoke jobs on remote clients
# %M is replaced with the queue name "sungrid_queue"
set_client
    -dir /tmp/liberate_%N -n 10 sungrid_queue
set_var rsh_cmd "qsub -b y -q %M %C"
```

When using distributed mode it is important to make sure that each client machine can access the necessary external SPICE binaries and licenses. To ensure this, create an `altos_init` shell script (sh or bash) in your home directory that sets the path to the binaries and licenses. Each time Liberate LV starts on a client machine, this script is sourced. Example script:

```
export PATH=/home/spice_vendor/bin:$PATH
export LM_LICENSE_FILE=2860@linux1:$LM_LICENSE_FILE
```

Liberate LV requires that all the server and host machines are NFS-mounted and that all the files and directories use the same path. In addition, all the files referenced in the Liberate LV Tcl file must use absolute path names. If using the `-n` argument to `set_client`, then the Liberate LV Tcl file should also be a full path name. References inside SPICE netlist files using `.include` or `.lib` commands are assumed to be relative to the top-level SPICE netlist. For example, if a model file `models.spi` is in a `models` directory at the same level as the top-level SPICE netlist, then use `.include ./models/models.spi` in the top level SPICE netlist.

Liberate LV automatically recovers from any client failures that are caused by system failures, license failures, or clients dying. If a client is killed, the tasks for that client are re-assigned to another available client. If a client fails to communicate back to the server within a pre-defined heartbeat time (`heartbeat_timeout`), Liberate LV removes that client from the list of available clients and re-assigns that client's tasks to another client. If the re-assigned tasks also result in a client failure, the tasks are skipped and the current cell is omitted from the library database (LDB).

If the tool is using a queuing system that permits pre-emption (stopping and re-scheduling of active jobs), Liberate LV tries to re-schedule the stopped clients' current tasks to another free client. If no clients are free, these tasks are put back on the list of characterization tasks to be performed. When the pre-empted client re-starts it is given a new collection of tasks to be performed. The characterization process continues until all of the tasks have been performed.

During characterization in this mode, an LDB is created as a single file with the name of `altos.ldb.<`*`pid`*`>`. After characterization, the `write_ldb` command moves `altos.ldb.<`*`pid`*`>` to the specified file name.

Packet Mode

Packet Mode (also known as bundle mode), is an alternative method of distributed processing with Liberate LV. To enable the packet mode of distributed processing, set the `packet_clients` variable to a value greater than 0, specifying the number of client machine to be used.

By default, a packet contains one cell, and is characterized in a single client machine.

When `packet_clients` is enabled, the `set_client` command is automatically disabled. A queuing system must be employed with parallel packets. The Liberate LV built-in job distribution system with multiple `set_client` commands is not supported in packet mode.

To specify the queuing command, set the `rsh_cmd` variable. The option `%B` supports the packet directory name. (Options `%C`, `%M`, `%O` in `rsh_cmd` are not supported in packet mode.) An example `rsh_cmd` usage for packet mode is shown below:

```
set_var rsh_cmd "qsub -q linux64 -b y -o %B/log -e %B/log"
```

Each packet is characterized separately with individual LDBs being created. During characterization an LDB directory with the name of `altos.ldb.<`*`pid`*`>` is created. Each packet of cells is characterized and the resulting LDB is stored into this directory. After characterization, the `write_ldb` command moves `altos.ldb.<`*`pid`*`>` to the specified *`dir`* name. The process is summarized here:

1. Liberate LV is started, reads in the SPICE netlist, and estimates the memory usage. This is called the Liberate LV *master process*.
2. Liberate LV creates 1 packet per cell. This is the default unless the `bundle_count` variable is set to a non-zero number. The `bundle_count` divides the number of cells in the library into that number of packets. (For example, if a library contains 600 cells, and the `bundle_count` is set to 10, there are 10 packets of 60 cells each.)
3. For each packet, on the machine that is running the master process, another Liberate LV slave job process is started (as a server) using the same script used in step 1), but enabled to run only the cells in the associated packet. This Liberate LV slave server starts the corresponding Liberate LV slave clients.
4. Packets are handled sequentially. As each packet finishes, the LDB is written in subdirectories inside the packet directory.
5. When the last packet is completed, control returns to the original Liberate LV master process, and the `write_ldb` command moves the LDB directory to the specified directory name.

4

Liberate LV Commands

This chapter describes the Tcl commands that control library validation.

Note: The command arguments that are prefixed with a hyphen (`-`) are optional except where explicitly indicated. All commands have a `-help` option. When this option is included with a command name, a help message is printed out that lists all currently available arguments for that command. There may be arguments that get printed out when help is used but are not officially supported. The only supported arguments are those that are documented in this manual. When `-help` is used, all other command arguments are ignored.

set_driver_cell	set_pin_vdd
set_gnd	set_var
set_operating_condition	set_vdd
v...	
validate_ccsn_data	validate_monotonicity
validate_data_range	validate_scaling
validate_library	validate_sdf
validate_lvf_data	

compare_arcs

This command is deprecated. Use compare_structure instead.

compare_ccs_ecsm

Use this command to compare CCS and ECSM capacitance model data between libraries and report differences that exceed the defined tolerances. Timing waveforms are not compared.

Arguments

`-absolute_average` Reports averages using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`*` | < <`*`type`*`> <`*`value`*`> >`

Specifies the absolute tolerance limit for CCS vs. ECSM error comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.001 * *`default_unit`*

`value`	Specifies the absolute tolerance.
`type`	Specifies the type of comparison, for example, `cap` or `ccs_cap`.

`-cells {`*`list`*`}` Specifies a list of cells to compare. Default: all cells

`-format <txt | xls | htm>`

Specifies the format for the output report. Default: `txt`

`htm`	Requests a report formatted as HTML.
`txt`	Requests a report formatted as standard text.
`xls`	Requests a report in an output format that is suitable for import into Microsoft Excel.

`-gui <`*`filename`*`>` Generates, and specifies a name for, an intermediate file that can be used for graphical comparisons of data with the `lcplot` utility. For more information, see "lcplot" on page 141.

`-nworst <number>`	Specifies how many of the worst delay difference and worst percent difference outliers to include in the summary for each data type (For exmaple, delay or leakage) of each cell. For each cell included in the summary, the worst absolute and relative outlier is reported. Default: 5
`-percent_max_diff`	Reports the percent of the maximum difference. Default: reports the maximum of the percent difference.
`-reltol <value>`	Sets percentage tolerance for CCS versus ECSM error comparison. `-reltol` defines a relative tolerance limit for each comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.01 (1%).
`-report <filename>`	Specifies the filename to be used for the output comparison file. Default: `<library>.cmp.txt` An overall comparison summary is also written to the standard output.
`-verbose`	Generates a report showing every comparison including those that do not exceed a tolerance. The output is written to the `-report` filename.
`<ccs_library>`	(Required positional argument) CCS Library filename.
`<ecsm_library>`	(Required positional argument) ECSM Library filename.

Example

```
# Set relative tolerance to 1%, delay tolerance to 1ps
compare_ccs_ecsm -reltol 0.01 -abstol "cap 1e-15" ccs.lib ecsm.lib
```

compare_ccs_nldm

The `compare_ccs_nldm` command compares the CCS data to the NLDM data in a single library and reports any differences that exceed the defined tolerances.

Arguments

`-absolute_average` Reports average using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`*`>` Sets the absolute tolerance for the CCS vs. NLDM error comparison. `-abstol` and `-reltol` define absolute and relative tolerance limits for each comparison. Any comparison that exceeds both these tolerances is considered an outlier and is reported. Default: `0.001 *` *`time_unit`* (typically 1ns).

`-cells {`*`cell_names`*`}`

Specifies a list of cell names. Default: all cells

`-exclude` Reverses the meaning of the `-cells` list. This excludes the specified list of cells from validation.

`-format <txt | xls | htm>`

Specifies the format for the output report. Default: `txt`

`htm` Requests an HTML output format. The default directory name is `./html` and can be changed using the `-group` argument. A one page comparison is generated for each cell group. Open the file `index.html` in a web browser to view the report.

`txt` Requests a report formatted as standard text.

`xls` Requests a report in an output format that is suitable for import into Microsoft Excel.

`-group <dirname>`	Specifies the name of a directory to store cell comparisons for each cell group. `-group` requests a group-by-group comparison, storing the results in the given directory name. A cell group is determined by the `define_group` command or by the `cell_footprint` attribute. The comparison report for each group is stored in the file `<dir_name>/<group_name>.cmp.txt`. Default: all cells in a single report
`-gui <filename>`	Generates, and specifies a name for, an intermediate file that can be used for graphical comparisons of data with the `lcplot` utility. For more information, see "lcplot" on page 141.
`-lcplot`	Uses the `lcplot` utility to display the comparison results graphically. The `-gui` argument is not required because a comparison data file called `<library_lib>.gui` is automatically created.
`-nworst <number>`	Specifies how many of the worst delay difference and worst percent difference outliers to include in the summary for each data type (delay or leakage, for example) of each cell. For each cell included in the summary, the worst absolute and relative outlier is reported. Default: 5
`-percent_max_diff`	Reports the percent of the maximum difference. Default: reports the maximum of the percent difference.
`-reltol <value>`	Sets percentage tolerance for CCS versus NLDM error comparison. `-reltol` defines a relative tolerance limit for each comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.01 (1%).
`-report <filename>`	Specifies the filename to be used for the output comparison file. Default: `<library_name>.cmp.txt`
`-verbose`	Generates a report showing every comparison including those that do not exceed a tolerance. The output is written to the `-report` filename. An overall comparison summary is also written to the standard output.
`<library_file>`	(Required positional argument) Specifies the library filename.

compare_ecsm_nldm

The `compare_ecsm_nldm` command compares the ECSM data to the NLDM data in a single library and reports any differences that exceed the defined tolerances.

Arguments

`-absolute_average` Reports average using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`*`>` Sets the absolute tolerance for the ECSM vs. NLDM error comparison. `-abstol` and `-reltol` define absolute and relative tolerance limits for each comparison. Any comparison that exceeds both these tolerances is considered an outlier and is reported. Default: `0.001 * `*`time_unit`* (typically 1ns).

`-cells {`*`cell_names`*`}`

Specifies a list of cell names. Default: all cells

`-exclude` Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.

`-format <txt | xls | htm>`

Specifies the format for the output report. Default: `txt`

- `htm` Requests an HTML output format. The default directory name is `./html` and can be changed using the `-group` argument. A one page comparison is generated for each cell group. Open the file `index.html` in a web browser to view the report.
- `txt` Requests a report formatted as standard text.
- `xls` Requests a report in an output format that is suitable for import into Microsoft Excel.

`-group <dirname>`	Specifies the name of a directory to store cell comparisons for each cell group. `-group` requests a group-by-group comparison, storing the results in the given directory name. A cell group is determined by the `define_group` command or by the `cell_footprint` attribute. The comparison report for each group is stored in the file `<dir_name>/<group_name>.cmp.txt`. Default: all cells in a single report
`-gui <filename>`	Generates, and specifies a name for, an intermediate file that can be used for graphical comparisons of data with the `lcplot` utility. For more information, see "lcplot" on page 141.
`-lcplot`	Uses the `lcplot` utility to display the comparison results graphically. The `-gui` argument is not required because a comparison data file called `<library_lib>.gui` is automatically created.
`-nworst <number>`	Specifies how many of the worst delay difference and worst percent difference outliers to include in the summary for each data type (delay or leakage, for example) of each cell. For each cell included in the summary, the worst absolute and relative outlier is reported. Default: 5
`-percent_max_diff`	Reports the percent of the maximum difference. Default: reports the maximum of the percent difference.
`-reltol <value>`	Sets percentage tolerance for ECSM versus NLDM error comparison. `-reltol` defines a relative tolerance limit for each comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.01 (1%).
`-report <filename>`	Specifies the filename to be used for the output comparison file. Default: `<library_name>.cmp.txt`
`-verbose`	Generates a report showing every comparison including those that do not exceed a tolerance. The output is written to the `-report` filename. An overall comparison summary is also written to the standard output.
`<library_file>`	(Required positional argument) Specifies the library filename.

Example

```
# Set relative tolerance to 1%, delay tolerance to 1ps
compare_ecsm_nldm -reltol 0.01 -abstol "delay 1e-12" ecsm.lib
```

compare_function

Use this command to compare the library function information in `comp_filename` against the `ref_filename`.

Arguments

`-cells {`*`cell_names`*`}` Specifies a list of cell names. Default: all cells

`-conformal` Uses Cadence Conformal in the comparison run. A `.conformal` directory is created to hold the results.

Note: When using the `-conformal` option, only the `-cells`, `-define`, and `-model` options are supported currently.

`-define {`*`directives`*`}`

List of Verilog `` `ifdef `` directives. `-define` argument is used when comparing Verilog files to define the Verilog `` `ifdef `` compiler directives to use for the function comparison. For example, there may be different function descriptions for permitting negative timing checks denoted by `` `ifdef NTC … `else … `endif `` directives. By default, `compare_function` checks the `` `else … `endif `` section. Using `-define {NTC}` the `` `ifdef NTC … `else `` section is checked instead. The `-define` argument supports a list of directives. Default: `{}`

`-exclude` Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.

`-extra_comp_files` List of extra files for comparison..

`-extra_ref_files` List of extra files for reference.

`-map {`*`map_pairs`*`}` Specifies a list of name-map pairs to match equivalent variables that have different names in the two files. For example, the name `int` might be used in one library for a variable containing an internal state while another library might use the name `int_wire`. Setting `-map` to `{int int_wire}` maps all occurrences of variable `int` to `int_wire`. Default: `{}`

`-model <`*`filename`*`>` Specifies the SPICE model filename to be used to create the functions from transistor-level SPICE subckts.

Note: If the `-model` option is specified, then `-conformal` is enabled by default irrespective of whether the `-conformal` option is specifed or not.

`-report <filename>`	Specifies the filename to be used for the output report file. Default: `<comp_filename>.cmp.txt`
`-verbose`	Generates a report showing every comparison, not just mismatches.
`<ref_filename>`	(Required positional argument) Reference filename; a library (`.lib`), Verilog library file (`.v`) or a Vital library file (`.vhd`).
`<comp_filename>`	(Required positional argument) Comparison filename; `.lib`, `.v`, or `.vhd`

The `compare_function` command compares the library function information in *`comp_filename`* against the *`ref_filename`*. The comparison and reference files can be a library (`.lib`) in Liberty format, a Verilog file (`.v`), or a Vital library file (`.vhd`). A comparison of the function information stored in these files is performed and any functional mismatches are reported in the output report.

For example, the function attribute of each cell output in the comparison library is compared for Boolean equivalence against the equivalent function attribute in the reference library. Alternatively, two Verilog files can be compared, or a Verilog (`.v`) file can be compared against the equivalent library (`.lib`).

The function comparison compares the functions from the two files only when the functions are described in basic logic primitives (`not`, `and`, `or`, `xor`). Comparison can also be made for sequential cells, provided both the library and Verilog descriptions were generated by Liberate. Any pin whose function description contains a user-defined primitive (UDP) that is not defined in the input Verilog file or whose function is dependent on the output of a sequential UDP is flagged as unmatched. A summary is given reporting the number of matches, mismatches, and unmatched functions.

Examples

```
# Compare two libraries and report all the comparisons
compare_function -verbose ref.lib comp.lib
# Compare a .lib and .v and report mismatches to a file
compare_function -report lib_vs_v.txt ref.lib ref.v
```

compare_library

Use this command to compare the comparison library against the reference library and report differences that exceed the defined tolerances.

Arguments

`-absolute_average` Reports average using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`* | {*`type_and_value_list`*}`>`

Sets absolute tolerance differences for comparisons. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.001 times the default unit for each data type. For example, if the time_unit is in nS, the `-abstol` for delay defaults to 0.001nS or 1ps.

This argument accepts a single value or a paired list of type and value. Individual tolerances can be set for each different data type by assigning values to the following compare types:

```
all, cap, ccs, ccs_cap, ccsn_dc, ccsn_vout, constraint,
delay, ecsm, ecsm_cap, hyper, leakage, max_cap,
max_trans, miller_cap, noise, power, trans, timing,
capacitance, voltage, current
```

If the argument has only a single value, then the type for that value is assumed to be `all`. The `-abstol` value must be given standard units (not library units). For example, use `delay 5e=h12` to set the `-abstol` for delay to 5ps.

`-cells {`*`cell_names`*`}`

Specifies a list of cells to compare. Default: all cells

This argument supports the use of a wildcard. If the `-exclude` argument is used, then the cells in the *`cell_names`* list are excluded from the comparison.

`-comp_adjust_tristate_load <-1 | 0 | 1>`

Adjusts the tri-state load of the comparison library before comparing. Default: `-1`

`-1`	Does not override the setting of the `adjust_tristate_load` variable for the comparison library.
`0`	Overrides the setting of the `adjust_tristate_load` variable to equal 0 for the comparison library.
`1`	Overrides the setting of the `adjust_tristate_load` variable to equal 1 for the comparison library.

`-exact_match`

Compares arcs only when the logic (`when`) conditions are an exact match.

Note: The `-exact_match` argument overrides the `-multiple_matches` argument.

`-exclude`

Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.

`-format <txt | xls | htm>`

Specifies the format for the output report generated by the `compare_library` command. Default: `txt`

`htm`	Formats the report as HTML. The default directory name is "`./html`" but you can change the name by using the `-group` argument. Using this `htm` value generates a one-page comparison for each cell group. Open the file `index.html` in a web browser to view the report.
`txt`	Formats the report as standard text.
`xls`	Formats the report for import into Microsoft Excel.

`-group <dirname>` Directory name to store cell comparisons for each cell group. `-group` requests a group-by-group comparison, storing the results in the given directory name. A cell group is determined by the `define_group` command or by the `cell_footprint` attribute. The comparison report for each group is stored in the file: `<dir_name>/<group_name>.cmp.txt`. Default: all cells in a single report

`-gui <filename>` Generates, and specifies a name for, an intermediate file that can be used for graphical comparisons of data with the `lcplot` utility. For more information, see "lcplot" on page 141.

`-index1_range <range>`

Limits the comparison to a range of values. Default: `use all indices`

The *`range`* is either two values separated by a "`-`", (e.g. "`1-3`" to compare the first three indices) or a single value (e.g. "`2`" to compare the second index only.)

`-index2_range <range>`

Limits the comparison to a range of values. Default: `use all indices`

The *`range`* is either two values separated by a "`-`" (For example, "`1-3`" to compare the first three indices) or a single value (For example, "`2`" to compare only the second index.)

`-keep_bundle`

Not expand bundles.

`-keep_buses`

Not expand buses.

`-lcplot` Uses the `lcplot` utility to display the comparison results graphically. The `-gui` argument is not required because a comparison data file called `<library_lib>.gui` is automatically created.

`-lib <abs | rel>` Requests an output report formatted like the `comp.lib`, where the values in the data table represent the absolute or relative differences between the two libraries. The output report is named `<comp.lib>_<abs | rel>.cmp`.

	`abs`	Values represent the absolute differences between the two libraries.
	`rel`	Values represent the relative difference between the two libraries.

`-multiple_matches`

Reports the results of comparing all arcs that have functional overlap with a reference arc. Default: reports the table that gives the best match.

Multiple arcs are shown in the output file as (N of M) after the `when :` line. For example:

```
when : !M1 Vs (!(M1)*!(M2))(1 of 2),Timing : combinational
```

Note: The `-exact_match` argument overrides the `-multiple_matches` argument.

`-nldm_only`

Requests comparison of only the NLDM data. Comparison of the following data is ignored:

- CCS and ECSM timing
- Noise and power constructs

`-no_interpolation`

Disables the comparison of data groups that have different indices, that is, no interpolation occurs between index points. Default: if the index values are different, the comparison values are interpolated.

`-nworst <`*`number`*`>`

Specifies how many of the worst delay difference and worst percent difference outliers to include in the summary for each data type (delay or leakage, for example) of each cell. For each cell included in the summary, the worst absolute and relative outlier is reported. Default: 5

`-ocv_avg_early_late`

Enables comparison of the average of early and late `ocv_delay` or `ocv_trans` data.

`-ocv_include_nominal`

Includes the nominal delay when comparing the early and late sigma values. Default: compare `ocv_sigma` values.

The Liberty Variation Format (LVF) `ocv_sigma_* table` values can be very small. Comparing these values directly can lead to a significant number of outliers. Use this option to include the nominal delay in the comparison. This will reduce the number of outliers. The Tempus based `mean_shift`, skewness, and stddev values are also adjusted before being compared.

`-padding`

Pads delays, transitions, and constraints by ½ input slew and pads power by an additional $\frac{1}{2}CV^2$ (where `C`=output capacitance, `V`=Vdd for that pin) before comparison. The padding does not apply to hidden power because the output is not toggled.

Padding is useful when comparing very small or even negative delay values.

`-padding_index <end | mid | same>`

Selects the slew index to use when adding padding. Default: `same`

`end`	Uses the last slew index.
`mid`	Uses the mid slew index.
`same`	Uses the same slew index as the reference value.

`-percent_max_diff`

Reports the maximum difference as a percentage. Default: reports the maximum of the percent difference.

`-ref_adjust_tristate_load`

Adjusts the tri-state load of the reference library before comparing.

`-1`	Does not override the setting of the `adjust_tristate_load` variable for the reference library.
`0`	Overrides the setting of the `adjust_tristate_load` variable to equal 0 for the reference library.
`1`	Overrides the setting of the `adjust_tristate_load` variable to equal 1 for the reference library.

`-reltol <`*`value`* | `{`*`type_and_value_list`*`}>`

Sets percentage tolerance differences for comparisons. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.01 (1%).

This argument accepts a single value or a paired list of type and value. Individual tolerances can be set for each different data type by assigning values to the following compare types:

```
all, cap, ccs, ccs_cap, ccsn_dc, ccsn_vout, constraint,
delay, ecsm, ecsm_cap, hyper, leakage, max_cap,
max_trans, miller_cap, noise, power, trans, timing,
capacitance, voltage, current
```

If the argument has only a single value, the type for that value is assumed to be `all`.

`-report <`*`filename`*`>` Specifies the filename to be used for the output file. Default: `<`*`comp_lib`*`>.cmp.txt`

`-skip {`*`list`*`}` Specifies a list of data comparison types to skip. Default: none (do not skip any types).

The *`list`* can include the same values available for the `-type` argument.

The `-skip` and `-type` arguments separate dynamic power from hidden power. Specifying `-skip {power}` skips dynamic power, specifying `-skip {hidden_power}` skips hidden power, and `-skip {power hidden_power}` skips both.

`-type {`*`list`*`}`

Specifies a list of data comparison types to include. Default: `all` (include all types). However, the exception to the default behavior is that ccsp types must be specified explicitly; they are *not* included by default.

The valid comparison types are:

```
all, attributes, cap, ccs, ccs_cap, ccs_delay, ccs_trans,
ccs_ecsm_cap, ccs_retain, ccsn_dc, ccsn_prop, ccsn_vout,
ccsp, ccsp_cap, ccsp_dc, ccsp_lc, ccsp_res, clear,
constraint, constraint_variation, delay,
delay_variation, drv_wform, ecsm, ecsm_cap,
ecsm_cap_variation, ecsm_variation, em, em_maxcap,
groups, hidden_power, hyper, leakage, max_cap, max_trans,
miller_cap, noise power, retain, retain_trans, si_prop_h,
si_prop_w, siv, trans, trans_variation,
ocv_const_mean_shift, ocv_const_stddev,
ocv_const_skewness, ocv_delay_mean_shift,
ocv_delay_stddev, ocv_delay_skewness,
ocv_trans_mean_shift, ocv_trans_stddev,
ocv_trans_skewness, setup, hold, removal, recovery,
min_period, mpw, nonseq_setup, nonseq_hold, three_state,
three_state_enable, three_state_disable, tristate,
preset, ocv_const, ocv_delay, ocv_trans, ocv_retain,
ocv_retain_trans.
```

For convenience, you can also request subsets of these types by specifying the following values:

value	**subset**
`capacitance`	`{cap ccs_cap ecsm_cap ecsm_cap_variation in_cap max_cap miller_cap}`
`ccs`	`{ccs_delay ccs_trans}`
`ccsp`	`{ccsp_cap ccsp_dc ccsp_lc ccsp_res}`
`constraint`	`{setup hold recovery removal mpw nonseq_setup nonseq_hold}`
`current`	`{ccs ccsn_dc}`
`em`	`compare the electro migration (EM) max_toggle_rate data`
`em_maxcap`	`compare the electro migration (EM) max_cap data` **Note**: `By default, both max_toggle_rate and max_cap data will be compared.`
`ocv_constraint`	`Compare the constraint sensitivity data`
`timing`	`{delay delay_variation ecsm ecsm_variation max_trans time_const trans trans_variation}`

	`voltage {hyper noise ccsn_vout}`
`-unmatched`	Requests a report on reference library data that do not have equivalent entries in the comparison library.
`-upscale`	When the data for a particular arc have different data dimensions in two different libraries (for example, 7x1 vs. 7x7), the data dimension of the smaller table is scaled up to match the data dimension of the larger table.
`-verbose`	Generates a report showing every comparison including those that do not exceed a tolerance. The output is written to the `-report` filename. An overall comparison summary is also written to the standard output.
<*`ref_lib`*>	(Required positional argument) Reference library.
<*`comp_lib`*>	(Required positional argument) Comparison library.

The `compare_library` command compares data found in the reference library (`ref_lib`) to the matching data found in the compare library (`comp_lib`) and reports the differences that exceed the defined tolerances. The report includes the comparison of attributes, capacitance, leakage, delay, transition, power, timing constraints, and comparison of advanced model data such as ECSM, CCS, Electromigration (EM), Liberty Variation Format (LVF), and Normalized Driver Waveform (NDW). For CCS, the current waveforms are converted to voltage waveforms and the comparisons are performed using delay and slew thresholds, rather than for each current measurement. If the table indices in the comparison library are different from the reference library, bi-linear interpolation is used before performing the comparison. For CCSN, the following data types are supported: `ccsn_dc`, `ccsn_vout`, and `miller_cap` (propagation tables are not supported). For CCSN_DC and ECSM, five data points are compared: the first point, the last point, and three intermediate points.

The output reports when reference and comparison values are zero (including cap, max_tran, max_cap, and so on.). If the reference value is zero and the comparison value is not zero then the percent difference is reported using a `"/0"`. This "bad" data point is not included in the computation of the overall average, but it is counted as an outlier.

When comparing libraries, the data entries must have equivalent conditions. Two entries are deemed equivalent if they have the same or overlapping logic conditions, related pins, and data type. If comparing libraries with different cell names use the `-define_map` command to map the names in the comparison library to the reference library. Note that all the pin names must match.

When comparing two libraries that have different index values, slew thresholds, and units, the values in the *comp_lib* are scaled accordingly before comparison. The following characters are used to indicate that some form of data manipulation has occurred before the comparison:

*	Data were scaled due to slew thresholds or units.
^	Input slews were interpolated.
~	Output loads were interpolated.
!	The indices were switched.
+	Both the *ref_lib* and *comp_lib* values were padded.
/	Slews were extrapolated.
#	Loads were extrapolated.

If the number of indices (dimensions) differs between two data groups, the data in the smaller dimension table is expanded to fit the larger dimension table. For example, if comparing delay data based only on input slew versus delay data based on slew and load, the 1-D slew table is expanded to a 2-D slew/load table by using the first value of the load indices from the 2-D table.

When the reference and comparison library values are 0 (including for cap max_tran, max_cap, etc.) a report is generated. If the reference value is zero and the comparison value is non-zero, then the percent difference is reported as a `"/0"`. This point is not included in the overall average equation, but it is counted as an outlier.

Note: If present, the cell_leakage_power is listed as the first entry in the leakage power comparison table and has the state `When : ""`.

Examples

compare_library Command Example

```
# Set all relative tolerances to 2%, constraint tolerance
# to 3%, power tolerance to 5%. Set absolute tolerance
# values for constraint, transition, leakage, and power
compare_library \
    -reltol { all 0.02  constraint 0.03 power 0.05 } \
    -abstol { constraint 5e-12
            trans 5.0e-12
            leakage 2.e-15
            power 3e-15 } \
    ref.lib \
    comp.lib
```

Sample Output

```
Legend : * scaled, ! indices switched, ^ slews interpolated, ~ loads interpolated,
+ half slew padding


*** BEGIN INVX1 COMPARISON ***


INVX1 Delay Comparison in ns
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+
| Row #|              Pin Name |   Ref Value |  Comp Value |       Diff |   Diff % |    Type |  Index_1 |  Index_2 |
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+
|     1|         INVX1:A->ON FR |    0.181790 |    0.171756 |  -0.010034 |   -5.52% |   delay |    0.304 |    0.058 |
|     2|         INVX1:A->ON FR |    0.239880 |    0.227162 |  -0.012718 |   -5.30% |   delay |    0.612 |    0.058 |
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+
|     3|         INVX1:A->ON RF |    0.149020 |    0.138183 |  -0.010837 |   -7.27% |   delay |    0.612 |    0.058 |
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+

INVX1 Delay SUMMARY
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      delay(ns) |       98 |     -0.00166 |       -2.30% |        4.27% |     -0.01272 |       -7.27% |        3 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
Worst delay outlier: Max Abs:     -0.01272, Row # :        2; Max Rel:       -7.27%, Row # :        3


INVX1 Transition Comparison in ns
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+
| Row #|              Pin Name |   Ref Value |  Comp Value |       Diff |   Diff % |    Type |  Index_1 |  Index_2 |
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+
|     1|         INVX1:A->ON FR |    0.219420 |    0.201528 |  -0.017892 |   -8.15% |  rising |    0.004 |    0.058 |
|     2|         INVX1:A->ON FR |    0.219220 |    0.201360 |  -0.017860 |   -8.15% |  rising |    0.013 |    0.058 |
|     3|         INVX1:A->ON FR |    0.219550 |    0.201632 |  -0.017918 |   -8.16% |  rising |    0.032 |    0.058 |
|     4|         INVX1:A->ON FR |    0.219330 |    0.201455 |  -0.017875 |   -8.15% |  rising |    0.072 |    0.058 |
|     5|         INVX1:A->ON FR |    0.219460 |    0.202950 |  -0.016510 |   -7.52% |  rising |    0.148 |    0.058 |
|     6|         INVX1:A->ON FR |    0.238460 |    0.225337 |  -0.013123 |   -5.50% |  rising |    0.304 |    0.058 |
|     7|         INVX1:A->ON FR |    0.316770 |    0.301474 |  -0.015296 |   -4.83% |  rising |    0.612 |    0.058 |
+------+----------------------+-------------+-------------+------------+----------+---------+----------+----------+

INVX1 Transition SUMMARY
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      trans(ns) |       98 |     -0.00242 |        -2.1% |        2.94% |     -0.01792 |       -8.16% |        7 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
Worst trans outlier: Max Abs:     -0.01792, Row # :        3; Max Rel:       -8.16%, Row # :        3

*** END INVX1 COMPARISON ***


Overall LIBRARY SUMMARY
```

```
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|    leakage(nW) |        2 |      0.00000 |        0.00% |        0.00% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+

+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|        cap(pf) |        2 |      0.00000 |        0.00% |        0.00% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+

+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      delay(ns) |       98 |     -0.00166 |       -2.30% |        4.27% |     -0.01272 |       -7.27% |        3 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+


Worst delay outlier (one per cell):
+-----+------------------+--------------+----------+------------------+--------------+----------+
|   # |             Cell |     Max Diff |    Row # |             Cell |    Max Diff% |    Row # |
+-----+------------------+--------------+----------+------------------+--------------+----------+
|   1 |            INVX1 |     -0.01272 |        2 |            INVX1 |       -7.27% |        3 |
+-----+------------------+--------------+----------+------------------+--------------+----------+



+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      trans(ns) |       98 |     -0.00242 |        -2.1% |        2.94% |     -0.01792 |       -8.16% |        7 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+

Worst trans outlier (one per cell):
+-----+------------------+--------------+----------+------------------+--------------+----------+
|   # |             Cell |     Max Diff |    Row # |             Cell |    Max Diff% |    Row # |
+-----+------------------+--------------+----------+------------------+--------------+----------+
|   1 |            INVX1 |     -0.01792 |        3 |            INVX1 |       -8.16% |        3 |
+-----+------------------+--------------+----------+------------------+--------------+----------+



+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      power(pJ) |       98 |      0.00003 |        1.99% |        6.54% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+

+----------+---------------+---------------+----------+
|  Entries |     Avg Diff% |        Sigma% | Outliers |
```

```
+----------+---------------+---------------+----------+
|      298 |        -0.82% |         5.18% |       10 |
+----------+---------------+---------------+----------+

*** LIBRARY Comparison of comp.lib to ref.lib completed on Wed May 31 14:39:41 PDT 2006
```

Sample Library Summary

```
Overall LIBRARY SUMMARY
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      Data Type |  Entries |     Avg Diff |    Avg Diff% |       Sigma% |     Max Diff |    Max Diff% | Outliers |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|    leakage(nW) |        2 |      0.00000 |        0.00% |        0.00% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|        cap(pf) |        2 |      0.00000 |        0.00% |        0.00% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      delay(ns) |       98 |     -0.00166 |       -2.30% |        4.27% |     -0.01272 |       -7.27% |        3 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      trans(ns) |       98 |     -0.00242 |       -2.18% |        2.94% |     -0.01792 |       -8.16% |        7 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
| constraint(ns) |        0 |      0.00000 |        0.00% |        0.00% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
|      power(pJ) |       98 |      0.00003 |        1.99% |        6.54% |      0.00000 |        0.00% |        0 |
+----------------+----------+--------------+--------------+--------------+--------------+--------------+----------+
+----------+---------------+---------------+----------+
|  Entries |     Avg Diff% |        Sigma% | Outliers |
+----------+---------------+---------------+----------+
|      298 |        -0.82% |         5.18% |       10 |
+----------+---------------+---------------+----------+
```

compare_spice

Use this command to compare the function information extracted from a cell-level SPICE netlist (`subckts`, `model`) to the function attributes in the `.lib`. The `compare_spice` command also checks that all the timing, power, constraint, and leakage arcs (including all necessary states) are represented in the library.

Arguments

`-cells {`*`cell_names`*`}`

Specifies a list of cells to compare. Default: all cells

This argument supports the use of a wildcard.

`-dir <`*`directory`*`>` Directory name to store intermediate files used in the lib-to-SPICE comparison. Default: `SPI`

`-exclude` Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.

`-extsim <`*`sim_name`*`>` Specifies an external SPICE simulator, for example, Spectre to be used by *inside_view* for resolving "collisions." Collisions occur when Liberate LV cannot logically resolve node values to a logical 1 or 0. To resolve the values, Liberate LV uses circuit simulation. Default: uses the Alspice simulator.

`-extsim_format <"spice" | "spectre">`

Type of netlist format. Default: "spice"

`-map {`*`map_pairs`*`}` Specifies a list of name-map pairs to match equivalent variables that have different names in the two files. For example, the name of internal pins used in the reference library might not be the same as those generated by Liberate LV from the SPICE subckts. Default: `{}`

`-model <`*`filename`*`>` (Required) SPICE model filename to be used to create function from transistor-level SPICE subckts.

`-overlap_when` Permit overlapping `when` conditions when comparing arcs. Default: require exact logic equivalence for all `when` conditions.

`-report <`*`filename`*`>` Specifies the filename to be used for the comparison report file. Default: `<`*`dir`*`>/<`*`comp_filename`*`>.cmpspi.txt`

`-subckts {<`*`filenames`*`>}`

	(Required) Specifies the list of files containing the SPICE subckts for each cell to be compared.
`-verbose`	Generates a report showing every comparison, not just mismatches.
<ref_filename>	(Required positional argument) Reference library (`.lib`).

The `compare_spice` command is used to compare the function information extracted from a cell-level SPICE netlist (`subckts`, `model`) to the function attributes in the `.lib`. It also checks that all the timing, power, constraint and leakage arcs (including all necessary states) are represented in the library. The function and arc information is extracted from SPICE netlists and stored in library (`.lib`) form in a file named *<dir>*/*<libname>*`.arc.lib`. The `compare_spice` command then calls `compare_arcs` to verify the arcs and calls `compare_function` to verify the function information of the extracted library with the reference library (*ref_filename*). The arc-comparison result is written to file *<dir>*/*<libname>*`.arc.cmp.txt`, while the function-comparison report is written to the `-report` filename (default *<dir>*/*<libname>*`.spi.cmp.txt`). Note that the `set_operating_condition` command is required before `compare_spice` to enable the correct use of the SPICE model.

Example

```
# Compare the arcs and function in library test.lib
# against the SPICE subckts description
set subckts {nand2x4.spi nor2x2.spi dffx1.spi}
set_operating_condition -temp 25 -voltage 1.2
compare_spice -model models.spi -subckts $subckts \
    -verbose test.lib
```

compare_structure

Use this command to compare the structure of two libraries.

Arguments

`-cells {`*`cell_list`*`}`	Specifies a list of cells to include for comparison. Default: all cells
`-cell_group_trend`	Compares the library structure amongst cells belonging to the same group. A cell group is created by grouping cells based on the `group_attribute` variable (default is the `cell_footprint` attribute) and then sorting them alphabetically. When this option is set, the first cell in the group is compared to the second cell in the group and then the second cell is compared to the third cell and so on. A single reference library should be given to `compare_structure` when this option is used. If a cell group has only one cell, then no comparison is performed for that group.
`-exclude`	Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.
`-nldm_only`	Requests that only the `nldm` data is compared. The ccs and ecsm timing and the noise and power constructs are ignored.
`-overlap_when`	Allows overlapping 'when' conditions to be considered equivalent. By default it allows only exact functional equivalence of 'when' conditions.
`-report {`*`filename`*`}`	Specifies the name to be used for the output file. Default: <base name of *`comp_lib`*>`.struct.tx`t

`-type {`*`list`*`}`

Specifies a list of data comparison types to include. Default: `all` (include all types). However, the exception to the default behavior is that ccsp types must be specified explicitly; they are *not* included by default.

The valid comparison types are:

```
all, attributes, cap, ccs, ccs_cap, ccs_delay, ccs_trans,
ccs_ecsm_cap, ccs_retain, ccsn_dc, ccsn_prop, ccsn_vout,
ccsp, ccsp_cap, ccsp_dc, ccsp_lc, ccsp_res, clear,
constraint, constraint_variation, delay,
delay_variation, drv_wform, ecsm, ecsm_cap,
ecsm_cap_variation, ecsm_variation, em, em_maxcap,
groups, hidden_power, hyper, leakage, max_cap, max_trans,
miller_cap, noise power, retain, retain_trans, si_prop_h,
si_prop_w, siv, trans, trans_variation,
ocv_const_mean_shift, ocv_const_stddev,
ocv_const_skewness, ocv_delay_mean_shift,
ocv_delay_stddev, ocv_delay_skewness,
ocv_trans_mean_shift, ocv_trans_stddev,
ocv_trans_skewness, setup, hold, removal, recovery,
min_period, mpw, nonseq_setup, nonseq_hold, three_state,
three_state_enable, three_state_disable, tristate,
preset, ocv_const, ocv_delay, ocv_trans, ocv_retain,
ocv_retain_trans.
```

For convenience, you can also request subsets of these types by specifying the following values:

value	**subset**
`capacitance`	`{cap ccs_cap ecsm_cap ecsm_cap_variation in_cap max_cap miller_cap}`
`ccs`	`{ccs_delay ccs_trans}`
`ccsp`	`{ccsp_cap ccsp_dc ccsp_lc ccsp_res}`
`constraint`	`{setup hold recovery removal mpw nonseq_setup nonseq_hold}`
`current`	`{ccs ccsn_dc}`
`em`	`compare electro migration data`
`ocv_constraint`	`Compare the constraint sensitivity data`
`timing`	`{delay delay_variation ecsm ecsm_variation max_trans time_const trans trans_variation}`
`voltage`	`{hyper noise ccsn_vout}`

`-verbose`

Specifies that all structural data in both libraries are itemized in the report.

<ref_lib>	(Required positional argument) Specifies the reference library name.
<comp_lib>	Specifies the name of the library to compare against the reference library (`ref_lib`). This argument is not used when the `-cell_group_trend` option is specified; otherwise it is required.

Structure comparison checks to see that both libraries have the same arcs, groups, and attributes. Values are *not* compared. (Use compare_library to compare values.) A report entry is made for any arc, group, or attribute that is present in one library, but not the other.

The output report has two sections:

- Reports on groups and attributes.

  A group or attribute difference is tagged with a "`?`" at the end of the comparison line. For attributes that represent Boolean logic functions (e.g. function, state_function, next_state, etc.) the comparison checks for functional equivalence, and flags differences.

- Reports on arcs.

  An attribute or group missing from either library has a "`?`" in the comparison or reference column for that attribute or group.

By default, only differences between the two files are reported.

Note: The `compare_structure` command replaces the functionality of `compare_arcs` within Liberate LV. However, `compare_arcs` is still supported for backward compatibility.

Sample Output

```
+------+-------------------------+-------------------------+-------------------------+-------------------------+
| Arc# |                 Arc Name |                Data Type |                Ref Type |                Comp Type |
+------+-------------------------+-------------------------+-------------------------+-------------------------+
|   377|   ALPHA_CKENOA12:phi->i0 |          rise_constraint |             setup_rising |            setup_rising |
|   378|   ALPHA_CKENOA12:phi->i0 |          fall_constraint |             setup_rising |            setup_rising |
|   379|       ALPHA_CKBUF09:i->o |               dc_current |         ccsn_first_stage |                       ? |
|   380|       ALPHA_CKBUF09:i->o |               dc_current |          ccsn_last_stage |                       ? |
+------+-------------------------+-------------------------+-------------------------+-------------------------+
```

define_arc

Use this command to specify a user-defined arc to override the Liberate LV automatic arc determination. An arc represents library data between a given pin and a related pin.

Arguments

`-constraint <"`*`function`*`">`

Specifies the logic condition applied to vectors but does not place the actual states in the library.

`-delay_threshold { <`*`in_rise`* `| cross> <`*`in_fall`* `| cross> <`*`out_rise`* `| cross> <`*`out_fall`* `| cross> }`

Defines a list of delay percentage measurement points (a ratio of VDD normalized to between 0 and 1) for the arc. Each argument consists of a list of four values representing the input_rise_delay threshold, the input_fall_delay threshold, the output_rise_delay threshold, and the output_fall_delay threshold in that exact order. If not specified, then all delays are measured at the values defined by the `delay_inp_rise`, `delay_inp_fall`, `delay_out_rise`, and `delay_out_fall` variables.

When differential pairs are specified for inputs using the `-dual_related` argument or for outputs using the `-dual_pin` argument, the delay measurements can be made using the voltage crossover between the differential signals. To request that the delay measurements use the crossover point, use the value of `cross` for the `-delay_threshold` argument instead of a ratio. For example:

```
-delay_threshold { 0.5 0.5 cross cross }
```

`-dependent_load <`*`value`*`>`

Specifies the load to add to dependent side pins. Use this argument to control the load applied to side outputs that impact the arc. Dependent loads specified with the `-dependent_load` argument take precedence over those specified by the `set_dependent_load` command.

`-dual_dir <U | D | B>`

Specifies the switching direction of dual pin used to set load direction. The `-dual_dir` argument is the equivalent of the `-load_dir` argument as it defines the direction of the load circuitry to apply to the `-dual_pin` of this `define_arc` command.

`U`	Sets a direction of up.
`D`	Sets a direction of down.
`B`	Sets a direction of both.

`-dual_pin <`*`name`*`>` Specifies the other pin in a pair of differential *output* pins.

`-dual_related <`*`name`*`>`

Specifies the other pin in a pair of differential *input* pins.

`-extsim_deck_header`

Allows to provide external simulator commands directly to the external simulator on an individual arc basis without using the Liberate process or reviewing them. This argument is intended to be used when an external simulator is used (refer to the `-extsim` argument of the **char_library** command). It is a local arc specific version of the variable `extsim_deck_header`. As Liberate does not parse the string specified by this argument, ensure that the contents are valid and consistent with the arc simulation. The value string can contain the return character ("\n"). The value string is included at the top of simulation deck. For example:

```
define_arc -extsim_deck_header ".ic n128 0" -related_pin
ck -pin Q ..
```

`-ic <"`*`ic_list`*`">` Specifies initial condition voltage values for each pin in the pinlist.

`-ignore` A flag that prevents simulation of all arcs originating from the related pin and ending at the pin. When this argument is used, only the `-pin` and `-related_pin` arguments are required. All other arguments, including the `-vector` argument, are ignored. This argument can be used to disable the internal view in Liberate LV from analyzing the specified arc.

`-load_dir <U | D | B>`

Specifies whether the pullup resistance, the pulldown resistance, or both resistances should be applied to this arc.

`U`	Applies the pullup resistance.
`D`	Applies the pulldown resistance.
`B`	Applies both the pullup and pulldown resistances.

`-metric <delay | glitch>`

Specifies the metric to use for measuring timing constraints.

`delay`	Produces a violation when a delay change at the probed pin exceeds the `constraint_delay_degrade` variable.
`glitch`	Produces a violation when the glitch-peak at the probed pin exceeds the `constraint_glitch_peak` variable.

`-pin {`*`pins`*`}` (Required) Specifies a list of destination pins for the arc (typically output pins for combinatorial arcs, input pins for timing constraint or hidden power arcs).

`-pin_dir <R | F>` Specifies the transition direction of pins.

`R`	Specifies a rising transition.
`F`	Specifies a falling transition.

`-pin_gnd {`*`pin voltage`* ...`}`

Specifies a list of paired values, each consisting of a pin and the arc-specific voltage that represents a logic *zero* for that pin.

`-pin_load <`*`value`*`>` Specifies additional circuitry to be applied to all the destination pins of the `define_arc` command. The `-pin_load` *`value`* argument refers to a template that defines the loading circuitry to be placed prior to the loading capacitance for the pin. The loading template must be pre-defined using the `define_pin_load` command. The additional circuitry can include pullup and pulldown resistances and series resistance.

`-pin_vdd {`*`pin voltage`* ...`}`

Specifies a list of paired values, each consisting of a pin and the arc-specific voltage that represents a logic *one* for that pin.

`-probe <{`*`names`*`} | altos_internal>`

Specifies a list of names of nodes to monitor for constraints in sequential cells. The `-probe` argument is used for timing constraints and defines the nodes to monitor when determining the constraint. It can be an external pin such as the `Q` pin in a flip-flop or an internal node name. Use the `-probe altos_internal` argument when a constraint can be measured at both an internal node and an output pin. This instructs Liberate LV to use the internal probe node. If the `-probe` argument is not specified then the pin defined by the `constraint_output_pin` variable is probed.

`-related_pin {`*`pins`*`}` Specifies a list of related pins (typically input pins for combinatorial arcs, clock pins for timing constraint arcs) while the `-pin` argument is a list of destination pins for the arc.

`-related_pin_dir <R | F>`

Transition direction of related pins.

`R` Specifies a rising transition.

`F` Specifies a falling transition.

`-slew_threshold {` *`lower_rise upper_rise lower_fall upper_fall`* `}`

Specifies a list of slew percentage measurement points (a ratio of VDD normalized to between 0 and 1) for the arc. Each argument consists of a list of four values. For `-slew_threshold` the values in the list represent the lower_rise_slew measurement threshold, the upper_rise_slew measurement threshold, the lower_fall_slew measurement threshold, and the upper_fall_slew measurement threshold in that exact order. If not specified, then all slews are measured at the values defined by the `measure_slew_lower_rise`, `measure_slew_upper_rise`, `measure_slew_lower_fall`, and `measure_slew_upper_fall` variables.

`-type <async | combinational | disable | edge | enable | hidden | hold | mpw | non_seq_hold |non_seq_setup | power | recovery | removal | setup>`

Specifies the type of arc. Default: `combinational`

`async` An `async` arc corresponds to a preset or clear transition.

`combinational`

	The arc is a `combinational` path from input pins (`related_pin`) to output pins (`pin`) for combinational cells.
`disable`	The `disable` type is used for specifying arcs that disable tri-state gates.
`edge`	An `edge` arc between an input and an output pin is an edge-triggered transition.
`enable`	The `enable` type is used for specifying arcs that enable tri-state gates.
`hidden`	A `hidden` arc is an arc that doesn't cause an output transition and is used to simulate hidden power for that `pin`.
`hold`	The arc is a timing constraint of type `hold` between data (`pin`) and a clock (`related_pin`) for sequential cells.
`mpw`	The arc is a timing constraint of type `mpw` between data (`pin`) and a clock (`related_pin`) for sequential cells.
`non_seq_hold`	The `non_seq_hold` type is used for specifying hold arcs between a pin and a non-clock related pin.
`non_seq_setup`	The `non_seq_setup` type is used for specifying setup arcs between a pin and a non-clock related pin.
`power`	The `power` type is used for specifying power-related arcs.
`recovery`	The arc is a timing constraint of type `recovery` between data (`pin`) and a clock (`related_pin`) for sequential cells.
`removal`	The arc is a timing constraint of type `removal` between data (`pin`) and a clock (`related_pin`) for sequential cells.
`setup`	The arc is a timing constraint of type `setup` between data (`pin`) and a clock (`related_pin`) for sequential cells.

`-value <value>` Use this to override the simulation and force a value for all entries into the data table for the specified arc. Default: use simulated values

`-vector <"stimulus">`

Specifies the stimulus to simulate this arc. It is defined as a string of bits (digits) where each bit corresponds to one string in the pinlist. Each bit can have the values `R` (rising), `F` (falling), `X` (don't care), `1` (logic high), `0` (logic low). The order of the bits must correspond one-ton-one to the pin list order defined by the **define_cell** `pinlist` argument or the `define_arc pinlist` argument. White space is permitted in the vector for readability. The vector value for a pin must be logically consistent with the `when` and `constraint` arguments. Else, the **define_arc** command is rejected. If a side input is specified as X it is overridden by the state of the pin as specified in the `when` or `constraint` argument. If busses are in the pinlist, there should be 1 bit in the vector for each bus. The bit value is applied to all elements in the bus. If different logical values are required for each bit in the bus, then the bus bits must be separately enumerated in the pinlist.

`-when <"function">` Specifies the logic conditions of the other pins of the cell to enable this arc using the Liberty `when` syntax. It corresponds to the Liberty `when` attribute.

`{cell_names}` (Required positional argument) List of cells.

The `define_arc` command specifies a user defined arc to override the automatic arc determination otherwise performed by Liberate LV. An arc represents library data between a given pin and a related pin. Typically this command is only required for the simulation of complex I/O cells.

The `define_arc` command can be applied to a single cell or a list of cell names. The template used for each arc defaults to the template defined for the cell unless a `define_index` command is specified for that particular arc.

Example

```
# Define the IOCELL
define_cell \
    -pinlist {IN OEN PAD OUT} \
    IOCELL

define_arc \
```

```
    -vector {XXRR} \
    -related_pin PAD \
    -pin OUT \
    IOCELL

define_arc \
    -vector {XXFF} \
    -related_pin PAD \
    -pin OUT \
    IOCELL

define_arc \
    -vector {R0RX} \
    -related_pin IN \
    -pin PAD \
    IOCELL

define_arc \
    -vector {F0FX} \
    -related_pin IN \
    -pin PAD \
    IOCELL
define_arc \
    -pinlist { A B C[5:0] OUT } \
    -vector { R 0 1 F } \
    -related_pin A \
    -pin OUT \
    myCell
define_arc \
    -pinlist { A B C[5] C[4] C[3] C[2] C[1] C[0] OUT } \
    -vector { R 0 101110 F } \
    -related_pin A \
    -pin OUT \
    myCell
```

define_cell

Use this command to specify how a cell is to be simulated by `validate_library`. By default, Liberate LV determines how to simulate each cell from the information in the input library and the cell's transistor level netlist. The `define_cell` command combined with `define_arc` commands can be used to augment and or override the automatic vector generation process.

Arguments

`-async {`*`pin_names`*`}` Specifies that the listed pins are asynchronous.

`-bidi {`*`pin_names`*`}` Specifies that the listed pins are bi-directional.

`-clock {`*`pin_names`*`}` Specifies that the listed pins are clocks.

`-constraint <`*`name`*`>` Specifies a template, pre-defined using the `define_template` command, that enables validation of timing constraints (setup, hold, recovery, removal). The template defines the range of input slews to use for the data and clock signals.

`-delay <`*`name`*`>` Specifies a template for delay tables, pre-defined using the define_template command, to be used for simulating each library construct. The template defines the range of input slews and output loads to use for the construct.

`-input {`*`pin_names`*`}` Specifies that the listed pins are inputs.

`-internal_supply {`*`supply_names`*`}`

Specifies a list of switched supply pin names. Use the `-internal_supply` argument for cells such as power switch cells to identify output pins that are to be treated as switched power nets. The internal supply net must be a port in the `subckt` definition of the cell. When this argument is used, all internal supply pins must also be identified as a supply using the `set_vdd`, `set_pin_vdd`, `set_gnd`, or `set_pin_gnd` commands.

`-output {`*`pin_names`*`}`

Specifies that the listed pins are outputs.

`-pinlist {`*`pin_names`*`}`

	Specifies the pin-order list. This information is used by the `-vector` argument of the `define_arc` command when specifying a user-defined timing arc. The pin list can contain internal pins as well as input, inout, and output pins.
`-power <`*`name`*`>`	Name of template for power tables. define which template to use for simulating each library construct: `power` enables simulation of switching power and hidden power (power dissipated when the output doesn't switch). The range of input slews and output loads to use for this construct is defined by the given template name where the template is pre-defined using the define_template command.
`-when <"`*`function`*`">`	Specifies user-defined cell level logic constraints using the Liberty format `when` syntax, constraining the Liberate LV automatic vector generation for this cell. The `define_cell -when` logical condition applies only to steady state signals such as leakage states and side input states that are non-switching. Liberate LV does not automatically infer simultaneous switching inputs based on the logical condition. You can use `define_arc` to specify simultaneous switching inputs, or specify a truth table to be translated automatically.
`{`*`cell_names`*`}`	(Required positional argument) Specifies the list of cells to be simulated.

The `define_cell` command defines how a cell is to be simulated by `validate_library`. By default, Liberate LV determines how to simulate each cell from the information in the input library and the cell's transistor level netlist. The `define_cell` command combined with `define_arc` commands can be used to augment or override the automatic vector generation process.

All pins of a cell must have a defined pin type. If a pin name or pin type does not apply to a particular cell it is ignored. For example, combinatorial cells such as NOR or NAND gates might not have `clock` or `async` pins so any definition for these pins is ignored. Likewise, if a pin name is specified but not used by a particular cell, it is ignored by that cell. The same pin name cannot appear in multiple pin types within a single `define_cell` command. For example, if one cell has an input Y and another has an output Y, then they must be defined uniquely with separate `define_cell` commands.

Liberate LV permits the re-use of Liberate style `define_cell` commands ignoring the arguments that are not required or supported by Liberate LV.

This command must be used before `validate_library`.

Examples

```
define_cell
    -input {A1 A2} \
    -output {Z} \
    -delay delay_3x3
    -power power_3x3 \
{NAND2X4 NOR2X2}

define_cell \
    -input {D} \
    -output {Q QN} \
    -clock {CK}\
    -async {SN} \
    -delay delay_5x5 \
    -power power_5x5 \
    -constraint constraint_3x3 \
{DFFX1}

define_cell \
    -input {A1 A2 A3 A4 SLP} \
    -output {Y}\
    -pinlist {A1 A2 A3 A4 SLP Y} \
    -delay delay_5x5 \
    -power power_5x5 \
    -when "!SLP" \
{MTAND2 MTAND3 MTAND4}
```

define_leafcell

Use this command to define the level of hierarchy that resides at the bottom of a cell level netlist.

Arguments

`-area <"string">`
Use this argument to provide the name of the diode area parameter in the cell. Default: `'area'`

`-extsim_model`
Loads the model files for the leafcells, allowing for partial include and partial use of the `read_spice` command. If this argument is used, the leafcell being defined also needs to have the `extsim_deck_header` variable insert a `.inc '<path>/modelfile.inc'` to load a model (probably a Verilog model) for this cell.

If a leafcell *does not have* the `-extsim_model` argument and the `extsim_model_include` variable is missing, the tool outputs an error requesting use of the `extsim_model_include` variable and quits.

If a leafcell *does have* the `-extsim_model` argument, you can load model files for it by using either:

- The `extsim_model_include` variable.
- The `extsim_deck_header` variable.

All other device models can be loaded by using the `read_spice` command.

`-length <"string">`
Use this argument to provide the name of the mos length parameter in the cell. Default: `'l'`

`-multiple <"string">`
Use this argument to provide the name of the multiple mos parameter. Default: `'m'`

`-pin_position {list_of_pin_positions}`

(Required) Use this argument to map the pin positions in this device to the nodes in the model, specifying one number for each pin in the cell.

The first pin is designated by `0`, where `0` is associated with drain, `1` with gate, `2` with source, and `3` with bulk.

For example,

```
define_leafcell -type nmos -pin_position {0 1 2 3} nch
```

`-pj <"string">` Use this argument to provide the name of the pj diode parameter in the cell. Default: `'pj'`

`-scale <"value">` Use this argument to provide the mos parameter scale factor in the cell. Default: `1.0`

This scale factor is used only by the Liberate *Inside View* to determine device sizes, and is not applied to the device sizes in the simulation netlist.

`-type <nmos | pmos | diode | r | c | nmos_stk | pmos_stk>`

(Required) Specifies the type of the cell.

`nmos`	Specifies the cell as an NMOS semiconductor.
`pmos`	Specifies the cell as a PMOS semiconductor.
`diode`	Specifies the cell as a diode.
`r`	Specifies the cell as a resistor.
`c`	Specifies the cell as a capacitor.
`nmos_stk`	Specifies the cell as an nmos stack. This type supports 5 pin stacked NMOS transistors. For 7 pin stacked MOS, the extra 2 pins are internal pins. Note that the pin position for stacked MOS is: d g1 g2 s b.
`pmos_stk`	Specifies the cell as a pmos stack. This type supports 5 pin stacked PMOS transistors. For 7 pin stacked MOS, the extra 2 pins are internal pins. Note that the pin position for stacked MOS is: d g1 g2 s b.

`-width <"string">` Use this argument to provide the name of the mos width parameter in the cell. Default: `'w'`.

{*cell_names*}	(Required positional argument) Use this argument to specify the list of leaf cell names.

Using the `define_leafcell` command allows Liberate LV to correctly identify devices in the cell netlist even when the process model file cannot be parsed. This command can be used in combination with the `extsim_model_include` variable to enable external simulation with the process models and the compiled netlist. This command supports identification of mosfets, diodes, resistors, and capacitors.

This command must be used before the `read_spice` command. If the `-extsim_model` argument is used with this command, the `define_leafcell` command must be used before the `char_library` and `write_library` commands as well.

Examples

Example 1:

```
# Define the cell NCH_MAC as a leafcell
define_leafcell \
    -type pmos \
    -pin_position { 0 1 2 3 } \
    PCH_MAC
# Define the cell PCH_map as a leafcell.
# first node (gate) in netlist must be swapped with the
# second node (drain) to match drain,gate,source,bulk order
define_leafcell \
    -type pmos \
    -pin_position { 1 0 2 3 } \
    PCH_map
```

Example 2:

```
set_var extsim_deck_header ".hdl /support/diode.va"
define_leafcell -extsim_model -type diode\
    -pin_position {0 1} {diodeva}
set spicefiles "netlist.sp"
lappend spicefiles "/support/sp_models.inc"
# Read in spectre netlists
read_spice -format spectre $spicefiles
```

define_map

Use this command to define a file for mapping cell names prior to writing out the library.

Arguments

<map_filename>	(Required positional argument) Defines a file that maps the names of cells between libraries.

The **define_map** command defines a file for mapping cell names prior to writing out the template, library, Verilog, VITAL, or datasheet files. It can also be used to map cell names when doing a library comparison using the **compare_library** command. It also changes the cells name(s) returned by the API functions: **ALAPI_inputs**, **ALAPI_outputs**, **ALAPI_inouts**, **ALAPI_internals**, **ALAPI_clocks**, **ALAPI_pinnames**, **ALAPI_name,** **ALAPI_cellnames,** and **ALAPI_cellgroups**.

If the specified file contains only cell name mapping, this command can be used before model generation (see **write_library**, **write_verilog**, and **write_vital**) and before the **write_template** command. However, if the specified file contains pin mapping, it must be used before the **read_ldb** and **read_library** commands.

The specified file should contain separate lines of one of the following formats:

- *<original_cell_name> <new_cell_name>*
- *<original_cell_name:pin_name> <new_pin_name>*

Example:

Define a mapping file before writing the library
```
read_ldb      my.ldb.gz
define_map  my_cell.map
write_library my_mapped.lib
```

The map_filename would contain the following information:
```
cell_1     cell1_new
cell_1:ck CLK
```

Liberate maps the cell named `cell1` to `cell1_new` and the pin named `ck` in `cell_1` to `CLK`.

define_validate_cell

Use this command to override the `validate_library` options for the specified list of cells.

Arguments

`-chain_length <number>`

Specifies the length of the cell chain. Default: same as the `-chain_length` option of `validate_library`.

`-cross_cap <value>`

Specifies the value of the coupling capacitor (in Farads) used in the `-xtalk` validation. Default: use `-wire_cap * 0.5`. If not locally specified then follow `-wire_cap` option of `validate_library`.

`-driver_cell <string>`

Specifies the cell to drive the current cell in the chain. Default is no driver cell.

`-driver_depth <integer>`

Specifies the number of additional driver cells to attach to the front of the chain. Default is `1` if the `-driver_cell` option is specified. Else, the default is `0`.

`-fanout <value>` Specifies the number of fanout cells to be added to the output of each cell in the simulation chain. The fanout cell is specified by the `-fanout_cell` option. Default: 0 (no fanout cells will be added).

`-fanout_cell <cell_name>`

Specifies the cell to be added to the output of each cell in the simulation chain. See the `-fanout` option for the number of fanout cells to be added. The specified fanout cell must have only one input and one output. Default: Current cell

`-receiver_cell <string>`

Specifies the cell to be used as a receiver to the current cell. Default is no receiver cell.

`-receiver_depth <integer>`

Specifies the number of additional receiver cells to attach to the end of the current cell. Default is `1` if the `-receiver_cell` option is specified. Else, the default is `0`.

`-second_level_fanout <value>`

Specifies the number of equivalent loads of the fanout cell to attach to the output pin of each fanout cell. This is often referred to as the second order load. Default: `0` (no second level fanout cell equivalent load).

`-wire_cap <`*`value`* `| min | mid | max>`

Specifies the wire capacitance applied between cells in the simulation chain. Set to the min, mid, or max load from the delay table for the arc under test, or to a specific value (in Farads). Default: follow the `-wire_cap` option `validate_library` .

`value`	Specifies a load (in Farads)
`min`	Sets the capacitance to the min load from the delay table.
`mid`	Sets the capacitance to the mid load from the delay table.
`max`	Sets the capacitance to the max load from the delay table.

`-wire_res <`*`value`*`>` Specifies the wire resistance in ohms for each side of a Pi network applied between each cell in the simulation chain. Default: 0.01 ohms.

`{`*`cells`*`}` Use this argument to specify the list of cells to which this will apply. Default: There is no default since this argument is required.

Use this command to specify unique fanout requirements on a per-cell basis. For cells or arguments that are not specified with the `define_validate_cell` command, the equivalent settings used with the `validate_library` command are automatically applied.

Examples

Example 1

```
# Use a chain length of 3 and a fanout of 4 INVX1 cells for all INV cells
define_validate_cell
    -fanout_cell INVX1
```

```
    -fanout 4
    -chain_length 3
    *INV*
```

Example 2

```
# Use a chain length of 2 and a fanout of 2 INVX2 cells for all BUF cells.
define_validate_cell
    -fanout_cell INVX2
    -fanout 2
    -chain_length 2
    *BUF*
```

get_var

Use this command to return the current value of a Liberate LV variable, whether it be the default value or a value set using the `set_var` command.

Arguments

<variable_name>	(Required positional argument) Use this argument to specify the name of a Liberate LV variable for which you want to determine the value. You can generate a list of Liberate LV variables by using the `printvars` command.

Example

```
# Get the value of default_timing
get_var default_timing
```

lv_summary_report

Read report files and present them in a Web browser.

Arguments

`-cells {list}`	List of cells
`-compare_arcs_rpt_filenames {file name}`	compare_arcs report file(s)

`-compare_ccs_nldm_rpt_filenames {file name}`

compare_ccs_nldm report file(s)

`-compare_ccsp_nlpm_rpt_filenames {file name}`

compare_ccsp_nlpm report file(s)

`-compare_function_rpt_filenames {file name}`

compare_function report file(s)

`-compare_library_rpt_filenames {file name}`

compare_library report file(s)

`-compare_spice_rpt_filenames {file name}`

compare_spice report file(s)

`-dir <directory name>`

Directory for report files. Default: RPT

`-library_file_name <lib file name>`

Library file name

`-open_browser_name <browser name>`

Web browser. Default: Mozilla

`-report <file name>`

HTML file name. Default: index.html

`-validate_data_range_rpt_filenames {file name}`

validate_data_range report file(s)

`-validate_monotonicity_rpt_filenames {file name}`

validate_monotonicity report file(s)

read_ldb

Arguments

`{<dir>/<libname>.ldb.gz}`

The read_ldb command can be used to recover from network failures during the simulation phase. An ldb (library database) is created under the validation directory in the file <libname>.ldb.gz. If the simulation phase is aborted or fails, a new run that starts where the previous one stopped can be done by using this command.

read_library

Use this command to read existing library files, in Liberty format, into memory where they can be used for validation and checking.

Arguments

`{`*`library_names`*`}` (Required positional argument) List of library files to be read in.

Examples

```
# Check the monotonicity and data range of a library
read_library test.lib
validate_data_range -warn_zero 2
validate_monotonicty test.lib
```

set_client

Use this command to define a machine or a queue to be used for distributed simulations during library validation.

Arguments

`-dir <directory_name>{%N%U%P%S}`

(Required) Use this argument to define a directory on the client machine to use as a temporary workspace for simulation jobs performed on that machine. Liberate LV creates the directory if it does not exist. You can incorporate the following objects into the name to create unique scratch directories for each individual validation run.

`%N`	Inserts the client number.
`%U`	Inserts the user name.
`%P`	Inserts the Liberate LV server process id.
`%S`	Inserts the server name.

`-n <number_of_clients>`

Use this argument to specify that Liberate LV is to submit jobs to this number of clients via the specified queue name.

When you use this argument, all file names within the Tcl file must be full path names and the full pathname for the Tcl file must be used when running Liberate LV.

<machine_or_queue_name>

(Required positional argument) Use this argument to specify the name of a client machine or a queue name.

As an alternative approach, you can instruct Liberate LV to perform distributed processing by explicitly defining the names of each of the client machines. To specify multiple machines, use multiple `set_client` commands. The network port number to be used can also be set using the `set_network_port` command. For more details on distributed parallel processing see Chapter 3, “Parallel Processing.”

Examples

```
# Set 20 machines for use with the LSF queue
set_client -dir /tmp/liberate_lv_%N -n 20 liberate_lv_lsf

# Or explicitly set the machines to use (no queue)
set_client -dir /tmp/scratch/%U_%N_%S_%P linux1
set_client -dir /tmp/scratch/%U_%N_%S_%P linux2
```

set_driver_cell

Use this command to define a pre-driver to be used to determine the input waveform for validation, overriding the default behavior, which is to use a linear ramp.

Arguments

`-char_pin <pin>`
: Specifies the primary validation pin, the one that is used to measure the transition. If `-char_pin` is specified, then `-pin_map` is required.

`-input_transition <value>`
: This pre-driver cell is driven at it's input by a linear ramp defined by the `-input_transition` argument. Liberate LV determines the loading of the pre-driver such that the output transitions of the driver cell are equivalent to the input transitions specified in the template when measured at the measure_slew* voltage levels. Input transition time, in seconds. Default: `5e-12`

`-instantiate`
: Allows the instantiation of driver cells for constant side input pins during validation. This argument causes this driver cell to be used for the specified cell/pin in the SPICE deck when the specified pin is a side pin and is static. The use of this functionality can result in a significant (~20%) run time penalty.

`-pin_map { <driver_pin>, <cell_pin>}`
: Use this argument to specify the driver cell pin that drives each pin in the `-pinlist`. The `-pin_map` maps by position to the `-pinlist` pins with the first `-pin_map` pin driving the first `-pinlist` pin, etc. If `-char_pin` is specified, then `-pin_map` is required.

`-pinlist {<cell> <pin>}`
: If a `-pinlist` is given, then the driver cell is used for only the specific cell and pin pairs in the list. The cell names can be wild-carded with a `*`.

`<driver_cell>`
: (Required positional argument) Specifies the name of the driver cell.

You can specify as many `set_driver_cell` commands as you wish.

Liberate LV supports an active driver that drives multiple inputs to a cell simultaneously. This capability allows multiple inputs to include delay offsets between related signals such as CK and CKN.

In nanometer technologies, it is common to have non-linear signal transitions. Using a pre-driver cell ensures that the simulated delay values and output slew more realistically model the typical on-chip behavior. A good choice for a pre-driver is to use a strong buffer cell.

When characterizing CCS data, Synopsys recommends using a CCS predriver waveform. For more information about this, see the variable predriver_waveform on page 134.

This command must be used before the `char_library` command.

Example

```
# Set the default pre-driver with a 10ps input ramp
set_driver_cell -input_transition 10e-12 bufx16
# Set the default pre-driver for all CLK pins, GATER:CLKIN
set_driver_cell -input_transition 10e-12  \
    -pinlist {* CLK GATER CLKIN } Clkbufx4
# connect driver cell output X to DFF1 inputs CK and SE, and
# connect driver cell output Y to DFF1 inputs CKN and SEN.
set_driver_cell \
    -input_transition 6e-12 \
    -char_pin X \
    -pinlist { DFF1 CKN DFF1 SEN DFF1 CK DFF1 SE } \
    -pin_map { Y Y X X } \
active_driver_2
```

set_gnd

Use this command to define the names of ground nets. You can specify multiple `set_gnd` commands if necessary.

Arguments

`-cells`	Specifies a list of cells that use this supply specification.
`-ignore_power`	Use this argument to have the tool ignore the contribution of the specified supply net. That is, the current in this supply net is not summed into any power measurement. The `-ignore_power` argument is skipped if the `-cells` argument is specified.
<gnd_net_name>	(Required positional argument) Specifies the name of the ground supply net.
<voltage_value>	(Required positional argument) Specifies the ground value (in volts).

Liberate LV automatically identifies `0`, `GND`, and `VSS` (case insensitive) as ground supplies and sets them to zero volts. Use the `set_gnd` command to set them to alternative values.

Examples

```
# Set VDD3 to 3 volts, BULK_GND to 0 volts.
set_vdd VDD3 3
set_gnd BULK_GND 0
```

set_operating_condition

Use this command to define the process corner, temperature, and default voltage to be used for library creation.

Arguments

`-process <name>`	(Required) Defines the name of the process corner to be simulated. The name must correspond to a `.LIB` name in the SPICE models.
`-temp <value>`	(Required) Specifies the temperature to use for simulation, in °Celsius.
`-voltage <value>`	(Required) Specifies the default positive supply voltage. This voltage is assigned to any VDD pin name. The default negative supply voltage is `0V`. To specify additional power- or ground-supply nets and their appropriate values, use the `set_vdd` and `set_gnd` commands. The default supply names are `VDD` for the positive supply; `VSS`, `GND`, and `0` for the negative supply.

Note: If the voltage or temperature specified by the `set_operating_condition` command is different than that set in the *first* library in the `validate_library` list, then validation results reflect the impact of voltage and temperature scaling.

Example

```
# Validate at 25°C, 1.2 Volts
set_operating_condition -temp 25 -voltage 1.2
```

set_pin_gnd

Use this command to associate a pin of a cell with a particular ground supply voltage.

Arguments

`-add_supply`	Creates and adds `-supply_name <name>` to the cell ground list.
`-supply_name <name>`	(Required) Specifies the name of the supply that drives this pin.
`<cell_name>`	(Required positional argument) Specifies the name of the cell.
`<pin_name>`	(Required positional argument) Specifies the name of the pin.
`<gnd_value>`	(Required positional argument) Specifies the ground supply value.

This command is useful for setting ground supplies on cells that have multiple power connections, such as level shifters. Typically, `set_pin_gnd` is used in conjunction with `set_pin_vdd`.

set_pin_vdd

Use this command to associate a pin of a cell with a particular supply voltage.

Arguments

`-add_supply`	Creates and adds `-supply_name <name>` to the cell vdd list.
`-supply_name <name>`	(Required) Specifies the name of the supply that drives this pin.
`<cell_name>`	(Required positional argument) Specifies the name of the cell.
`<pin_name>`	(Required positional argument) Specifies the name of the pin.
`<vdd_value>`	(Required positional argument) Specifies the power supply value.

This command is useful for setting power supplies on cells that have multiple power connections, such as level shifters. Typically, `set_pin_vdd` is used in conjunction with `set_pin_gnd`.

Example

```
# Set the voltage swing on the input pin of a level shifter
set_pin_vdd level_shifter_3to1 A1 3.0
```

set_var

Use this command to set variables that are specific to Liberate LV.

Arguments

`-cells`	List of cells. Default: all cells
`-pin {`*`pins`*`}`	List of destination pins for the arc (typically, output pins for combinational arcs, input pins for timing constraint, or hidden power arcs). (REQUIRED)
`-pin_dir <R | F>`	Transition direction of pin(s).
`-related_pin {`*`pins`*`}`	List of related pin names (typically input pins for combinational arcs, clock pins for timing constraint arcs).
`-related_pin_dir <R | F>`	Transition direction of related pin(s).
`-type < constraint | delay | delay and power | hold | leakage | mpw | nochange | power | recovery | removal | setup >`	Type of arc (Default: *all types*)
<*`name`*>	Variable name (REQUIRED)
<*`value`*>	Variable value (REQUIRED)

The options `-cells`, `-type`, `-pin`, `-pin_dir`, `-related_pin`, and `-related_pin_dir` are used to specify local cell and arc specific variables and their corresponding values. All options are not valid for all parameters. If an option is not allowed, an error is issued and the setting is ignored. If an option is omitted, any value for that option is allowed. The options `-cells`, `-pin`, and `-related_pin` support the usage of the wildcards `*` and `?`. Some variables can only be set at a global level. If you specify local cell and arc variable that can only be applied globally, a warning is issued in the log file and `set_var` is ignored.

The available Liberate LV variables are defined in Chapter 5, “Liberate LV Variables.”

Example

```
set_var -cells DFF* write_logic_function false
```

set_vdd

Use this command to define the names of power supplies. You can specify multiple `set_vdd` commands if necessary.

Arguments

`-cells`	Specifies a list of cells that use this supply specification.
`-ignore_power`	Use this argument to have the tool ignore the contribution of the specified supply net. That is, the current in this supply net is not summed into any power measurement. The `-ignore_power` argument is skipped if the `-cells` argument is specified.
<vdd_net_name>	(Required positional argument) Specifies the name of the power supply net.
<voltage_value>	(Required positional argument) Specifies the voltage value (in volts).

Liberate LV automatically identifies the net name `VDD` (case insensitive) as a power supply and sets it to the default voltage specified by the `set_operating_condition` command. Use the `set_vdd` command to set the `VDD` power supply to a different value.

Examples

```
# Set VDD3 to 3 volts
set_vdd VDD3 3
set_gnd BULK_GND 0
```

validate_ccsn_data

Use this command to check the CCSN data items in a given library.

Arguments

`-cells {`*`list`*`}`	Specifies a list of cells to check. Default: all cells
`-exclude`	Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from validation.
`-expand_buses`	Supports CCSN data validation for bundle or bus groups. It specifies that the library outputs the individual pins in the report and no buses or bundles.
`-match_timing`	Reports if a timing arc does not have a corresponding CCSN stage with the same '`when`' condition.
`-verbose`	Generates a report of all CCSN arcs in the library, regardless of pass or fail.
`-report "`*`file_name`*`"`	Specifies the filename to be used for the information that is returned by the `validate_ccsn_data` command. Default: `stderr`

The `validate_ccsn_data` command checks the following CCSN data conditions in a given library.

- Arc checks
  - Each CCSN group under the timing group must contain a `ccsn_first_stage`.
  - If an arc contains a `ccsn_last_stage` it must also contain a `ccsn_first_stage`.
  - A timing arc must have a corresponding CCSN stage with the same 'when' condition.
  - CCSN arc data should not be duplicated.
- Pin checks
  - An input pin requires a `ccsn_first_stage`.
  - An output pin requires a `ccsn_last_stage`.

- An inout pin requires both a `ccsn_first_stage` and a `ccsn_last_stage`.

- Group attributes
  - The attribute "`is_inverting`" must be present and be either true or false.
  - The "`is_inverting`" attribute when it appears in an arc based CCSN construct must be consistent with the unateness of the timing arc it is under. For example, if a timing arc is negative_unate and there is only a single `ccsn_first_stage` then the "`is_inverting`" attribute must be true. For a two stage arc that is positive_unate then the "`is_inverting`" attribute must be either true for both the `ccsn_first_stage` and the corresponding `ccsn_last_stage` or must be false for both.
  - The attribute "`stage_type`" must be present and be one of "l", "`pull_down`" or "`both`".

- Miller caps
  - If stage_type is "`pull_up`", `miller_cap_rise` is required.
  - If stage_type is "`pull_down`", `miller_cap_fall` is required.
  - If stage_type is "`both`", `miller_cap_rise` and `miller_cap_fall` are required.
  - The `miller_cap_rise` and `miller_cap_fall` fields must be non-negative.

- dc_current
  - The field `ccsn_dc_current` is required.
  - The dimensions of the DC table must be at least 10x10.
  - The values in index 1 of the table (input voltage) and in index 2 of the table (output voltage) must be in increasing order.
  - The first entry in the table index for both the input voltage and output voltage must be less than VSS.
  - The last entry in the table index for both the input voltage and output voltage must be greater than VDD.
  - Max dc_current must be greater than 1uA.

- Output voltage
  - If stage_type is "`pull_up`", `output_voltage_rise` must be present.
  - If stage_type is "`pull_down`", `output_voltage_fall` must be present.

- If stage_type is "`both`", `output_voltage_high` and `output_voltage_low` must be present.
- The `output_voltage_low` and `output_voltage_high` fields must have at least one vector.
- The index 1 (time values) must be monotonically increasing and >= 0.
- There should be at least 5 time entries.

- Propagated noise
  - The input noise height must be greater than 0.0 and less than or equal to VDD.
  - The input noise width must be greater than 0.0.
  - The net capacitance must be non-negative.
  - The index 1 (time values) must be monotonically increasing and >= 0.
  - There should be at least 5 time entries.
  - The values in the table (voltage) must be greater than 0.0 and less than or equal to VDD.
  - If stage_type is "`pull_up`", `noise_propagation_low` must be present.
  - If stage_type is "`pull_down`", `noise_propagation_high` must be present.
  - If stage_type is "`both`", `noise_propagation_low` and `noise_propagation_high` must be present.
  - The `noise_propagation_low` and `noise_propagation_high` fields must have at least one vector.

The `validate_ccsn_data` command must be used after `read_library`.

Example

```
read_library my_ccsn.lib
validate_ccsn_data -verbose -report my_ccsn_report.txt
```

validate_data_range

Use this command to check the range and quality of data in the data tables of cells.

Arguments

`-at_least_one` Check that at least one value in each table is greater than the minimum or is less than the maximum.

`-cells {`*`cell_names`*`}`

Specifies a list of cells to compare. Default: all cells

This argument supports the use of a wildcard. If the `-exclude` argument is used, then the cells in the list are excluded from checking.

`-direction` Allows for different range checking for rise/fall/high/low data. Default: "" (all directions)

`-exclude` Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from validation.

`-index <`*`integer`*`>` Use this argument to specify that table indexes, rather than table values, should be checked. Default: check values, except for `ccsn_vout`, where time, `index_3`, is checked. For example,

```
validate_data_range -index 1
```

`-intra_table <`*`value`*`>`

Use this argument to compare the rise_power and fall_power data, and report discrepancies beyond a specified threshold. Default: `0.0` (Do not compare the data.)

This argument and `-intra_table_min` are used to compare the rise_power and fall_power data, and report discrepancies for differences that satisfy this equation:

$$\frac{\text{abs}(\text{rise} - \text{fall})}{\min\langle\ \text{rise}|\ \text{fall}\rangle} \times 100 > \text{value}$$

`-intra_table_min <`*`value`*`>`

Use this argument to specify the minumum value of a table entry to be considered for intra_table comparison; if the value is below that threshold, it is ignored. Default: `1.0e-5`.

This argument and `-intra_table` are used to compare the rise_power and fall_power data, and report discrepancies beyond a specified threshold.

`-max_range {`*`values`*`}`

Specifies either a single maximum value, or a list of maximum values corresponding to the load index in any two-dimensional data-table. Default: no maximum

If the data being checked is not a two-dimensional table with an index based on capacitive load, then the first value in the `-min_range` list is used as the minimum and the last value in the `-max_range` list is used as the maximum. The *`values`* should be given in library units. For example, if the library timing unit is `ns` then to set the maximum value to 1ns set `-max_range` to `1`.

`-min_range {`*`values`*`}`

Specifies either a single minimum value, or a list of minimum values corresponding to the load-index in any two-dimensional data-table. Default: no minimum

If the data being checked is not a two-dimensional table with an index based on capacitive load, then the first value in the `-min_range` list is used as the minimum and the last value in the `-max_range` list is used as the maximum. The *`values`* should be given in library units.

For example,

```
validate_data_range -type cap -min_range 0.0001 \
     -max_range 10
```

`-report <`*`filename`*`>` Specifies the filename to be used for the information that is returned by the `validate_data_range` command. Default: `stderr`

`-type <type>`

Specifies the type of data to check, because different types of range-checking can be applied to different data types. Default: `all`

Valid types include:

```
all area cap ccs ccs_cap ccsn_dc ccsn_prop ccsn_vout
ccsp_cap ccsp_dc ccsp_lc ccsp_res constraint delay ecsm
ecsm_cap hidden hold leakage max_cap max_trans miller_cap
min_period mpw noise nonseq_hold nonseq_setup ocv_const
ocv_delay ocv_trans power recovery removal setup
switching_power trans
```

where,

`ccsp_cap` = intrinsic capacitance
`ccsp_dc` = dynamic current
`ccsp_lc` = leakage current
`ccsp_res` = intrinsic resistance
`hidden` = enables range checking for hidden power, i.e., power consumed when the outputs of a cell are not switching.
`power` = applies range checks to all internal power data, i.e., both hidden and switching power.
`switching_power` = enables range checking for switching power, that is, power consumed when the outputs of a cell are switching.

`-verbose`

Generates a report of the min and max value for every check (one per arc), regardless of pass or fail.

`-warn_zero <integer>`

Generates a warning if the number of consecutive zeros equals the `warn_zero` argument. Default: `-1` (No warning is issued).

If *`integer`* is set to 1, all zeros are flagged as warnings. If *`integer`* is set to 2, than any table sequence of "`0, 0`" generates a warning, and so on.

Multiple `validate_data_range` commands can be issued in the same Liberate LV session to specify different checks for different data-types. A `read_library` command is required before `validate_data_range` to read the library to be checked.

Example

```
read_library test.lib
validate_data_range -type cap -min_range 0.0001 \
    -max_range 10
```

```
validate_data_range -type delay -min_range 0
validate_data_range -type trans -report slew.range.txt \
    -max_range {0.4 0.5 0.6 0.7 0.8 0.9 1 2}
```

validate_library

This command performs library timing and power validation.

Arguments

`-absolute_average` Reports averages using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`*`>` Sets the absolute tolerance for comparison. Default: For delay `2e-12`; power `2e-15`; trans `1e-11`; constraint `1e-11`; leakage `2e-12`

`-auto_arc` Derives all arcs from the SPICE sub-circuit, ignoring the arcs defined in the input library. This may highlight potential inaccuracies due to incomplete state coverage in the input library. If user-specified `define_arc` commands are also given, `-auto_arc` applies only to arcs that do not have `define_arc` commands. To limit the validation to only user-defined arcs set the `user_arcs_only` variable.

`-cdb {`*`filename`*`}`

Specifies a cdb file to be used in ETS crosstalk analysis. It requires `-xtalk` and `-timer` “ets” options to be set.

`-cells {`*`cell_names`*`}`

Specifies a list of cells to validate. Default: all cells.

`-chain_length <`*`number`*`>`

Specifies the length of the cell chain. Default: `1`

`-constraint` Enables validation of timing constraints.

To validate a constraint, a linear search is performed around the original library constraint value. This search starts at `original_lib_value + (constraint_steps/2 * constraint_step_size)` and stops at `original_lib_value - (constraint_steps/2 * constraint_step_size)` (default ±20ps around the original value). If the constraint criteria (set using the `set_constraint_criteria` command) is violated at the start of the range, the constraint value is too optimistic. If the constraint criteria is never violated, the constraint value is too pessimistic. In the constraint validation output report, <*dir*>/<*libname*>.const.<*extsim*>.cmp.txt, constraints that are too pessimistic are marked with "`>>`" and constraints that are too optimistic are marked with "`<<`". To find the true value of a constraint that is optimistic or pessimistic, increase the `-constraint_steps` or `-constraint_step_size` variables and re-run those cells that have these failures.

`-constraint_report_style <"all" | "separate">`

Controls the constraint reporting style. Default: "`all`"

`all`	Combines all data into one report.
`separate`	Produces separate reports for setup, hold, recovery, removal, nonseq_setup, and nonseq_hold.

`-constraint_step_size`

Specifies the spacing of constraint sweep steps. Default: `4e-12`

`-constraint_steps` Specifies the number of constraint sweep steps. Default: `10`

`-cross_cap <value>`

Specifies the value of the coupling capacitor (in Farads) used in the `-xtalk` validation. Default: use `-wire_cap * 0.5`.

`-db {`*`library_name.db(s)`*`}`

Specifies one or more Synopsys compiled library database files to use for SDF validation. Default is to recompile the input libraries.

This avoids having to recompile the library for each run of `validate_library` when the timer is PrimeTime.

`-dir <directory>` Specifies the name of a directory to store the results and data used for the validation run. Default: "`VAL`".
The directory must be specified as a full path if the `set_client` command is also used.

`-exact_match`

Enables the comparison of arcs that have identical states (“when” condition). An unconditional arc is never compared to a conditional one.

`-exclude`

Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from validation.

`-extrapolate` Specifies that loads and slews that are outside the range given in the table are to be extrapolated from the delay template. It uses the first slew/load minus half the difference between the second and first slew/load and the last slew/load plus half the difference between the last and next-to-last slew/load.

`-extsim <ski | spectre>`

Specifies the name of an external simulator to use. Default: `ski`.

`ski` Uses the Spectre simulator with SKI flow.

`spectre` Uses the Spectre simulator.

`-extsim_format <spice | spectre>`

Specifies the format of the netlist. Default: `spectre` (However, when `-extsim` is set to `hspice`, the default format is `spice`.)

`spice` Uses the SPICE format.

`spectre` Uses the Spectre format.

`-fanout <value>` Specifies the number of fanout cells to be added to the output of each cell in the simulation chain. Default: `0`

The cell used for fanout is either the current cell in the chain or the cell defined by the `-fanout_cell` argument.

`-fanout_cell <cell_name>`

Specifies a cell to be added to the output of each cell in the simulation chain. Default: Current cell

The specified cell must have only one input and one output.

`-format <htm | txt | xls>`

Specifies the format for the output report. Default: `txt`

`htm`	Requests a report formatted as HTML.
`txt`	Requests a report formatted as standard text.
`xls`	Requests a report in an output format that is suitable for import into Microsoft Excel.

`-glitch` Performs glitch only comparison.

`-glitch_last` Performs output glitch comparison.

`-glitch_noise <{list}>` [peak, width] value pairs.

`-interpolate` Enables validation with slews and loads created by interpolating the indices specified in the library. The first interpolated index is the first index minus the difference between the first and second indices. For a 7×7 delay table, this means that 6 slews and 6 loads are used. For example, if the library has characterized the slews 100, 300 and 500, when the `-interpolate` argument is specified, slews of 200 and 400 would be validated.

`-io` Enables IO cell validation.

`-leakage` This option enables validation and reporting of leakage only. It can be used instead of -power, which will validate switching, hidden power and leakage power. The leakage validation report will be created under the <dir>/REPORTS directory and will be named <libname>.lkg.<timer_type>_<spice>.cmp.<format>. Example: ./VAL/REPORTS/mylib.lkg.pt_spectre.cmp.txt

`-lef <file_name>` Specifies a LEF file to be used for ECSMP model validation. If the file is not specified, an error is generated.

`-lvf` Compares Monte Carlo to Tempus using LVF data on a chain of gates. The library must have `ocv_sigma_*` data. Currently, this argument supports only tempus timing analyzer.

`-load_range <value>` Specifies a range of loads to use for timing validation. Default: Use all loads.

For example: `-load_range "2-5"` uses the 2nd, 3rd, 4th and 5th loads (if they exist), ignoring other existing loads such as 1st, 6th, 7th, etc.

`-loads {<values>}` Specifies a list of load values to use. Default: Use the loads defined in the library.

`-max_percent_diff` Reports the maximum percentage difference between all the compared values. (In other words, the tool calculates all the differences, then reports the largest percentage difference.) Default: Off

Without this argument, the tool reports the percentage of the maximum difference. (It determines the largest absolute difference, then reports what percentage that difference represents.)

`-model <filename>` Specifies a SPICE model filename to be included in the SPICE netlist as `".inc <model>"`. Default: "" (none)

`-monte_carlo_trials <value>`

Specifies the number of Monte Carlo trials to perform for SSTA validation. Both the nominal and sigma results are compared. Default: `1000`

`-noise` Includes comparison of Propagated Noise (CCSN only).

`-noise_immunity` Includes comparison of noise immunity.

`-power` Enables power data validation. PrimeTime-PX, EPS, and Voltus are supported. A chain length of one is used for power validation. Also, a license is required for the power analysis tool. The -timer argument specifies which power tool will be used.

The support data formats for power validation are:

-timer setting	**data formats**
primetime	nlpm, ccsp
ets	nlpm, ecsmp
tempus	nlpm, ecsmp, ccsp

`-power_combined_rise_fall`

Compares the sum of rise and fall power in the SPICE simulation with static power analysis results. This is useful for validating libraries that have only power groups rather than distinct rise/fall_power groups.

`-power_dont_add_load`

Does not add output load. Default: Add load.

For `nlpm` switching power comparisons, the power reported by the power analyzer is modified before it is compared with the total switching power measured during the SPICE simulation. The power due to all the switching outputs, caused by switching the input pin, and the power due to each output load (`0.5*C*V*V`) are added. The `-power_dont_add_load` argument prevents the power due to the output load (`0.5*C*V*V`) from being added. It should be used if the `nlpm` library power data includes the load component.

`-power_models {<nlpm | ccsp | ecsmp>}`

Specifies which power models should be validated. The comparison includes leakage power, hidden power and switching power. Default: For PrimeTime px, `{ccsp nlpm}`; for ETS/EPS, `{nlpm}`.

`nlpm` Validates the NLPM power model.

For `nlpm` switching power comparisons, the power reported by the power analyzer is modified before it is compared with the total switching power measured during the SPICE simulation. The power due to all the switching outputs, caused by switching the input pin, and the power due to each output load (`0.5*C*V*V`) are added. The `-power_dont_add_load` argument prevents the power due to the output load (`0.5*C*V*V`) from being added. It should be used if the `nlpm` library power data includes the load component.

`ccsp` Validates the CCSP power model.

`ecsmp` Validates the ECSMP power model.

`-reltol <value>` Sets a relative tolerance limit for each comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: for delay `0.02`; power `0.02`; delay_variation `0.1`; trans `0.1`; constraint `0.1`; leakage `0.02`.

`-report_dir <dir_name>`

Creates a directory to store all the report files. Default: `"dir/REPORTS"`

`-resize_table <1x1 | 2x2 | 3x3>`

Reduces the data table size. Default is "" (do not apply any reduction). The setting overrides the variable `debug_flow`.

`-second_level_fanout`

Specifies the number of equivalent loads of the fanout cell to attach to the output pin of each fanout cell. This is often referred to as the second order load. Default: `0` (no second level fanout cell equivalent load).

`-sigma_factor <value>`

Specifies a sigma scale factor to apply to the ocv_sigma_* data before it is summed to the corresponding nominal data of the specified by `-type`. Default: 1.0.

`-slew_range <value>` Specifies the slew range to use for the related pin during constraint validation. If this option is specified, it will override the -slew_range. Default: Use the same slew range as specified by `-slew_range`.

For example: `-slew_range "2-5"` uses the 2nd, 3rd, 4th and 5th slews (if they exist), ignoring other existing slews such as 1st, 6th, 7th, etc.

`-slew_range_rpin`

Specifies a range of slews to be used for the related pin during constraint validation. The default is to use the same slew range as the `-slew_range` option. For example, `validate_library -slew_range_rpin 2-5`.

`-slews {<values>}` List of slew values to use. Default: use the slews defined in the library

`-ssta` Enables validation of statistical timing analysis (SSTA) against Monte Carlo circuit simulation.

`-start_from <spice | timer | compare>`

Used to repeat or re-start the validation from a specific phase: spice, timer or compare. Default: Performs all the validation steps.

For example, if the timer license was not available, this argument can be used to run starting from the timing analysis without needing to re-run all the simulations.

`spice`	Begins at the SPICE simulation phase, skipping the netlist generation phase.
`timer`	Begins at the timer phase, skipping the netlist generation and SPICE simulation phases.
`compare`	Begins at the compare phase, skipping the netlist generation, SPICE simulation, and timer phases.

`-subckts {<`*filenames*`>}`

Specifies the list of files containing the SPICE subckts for each cell in the library. Default: "" (none)

`-thread <`*number*`>` Number of different CPU threads to use. Default: use all available threads

`-timer <primetime | ets | tempus | timer_cmd_option>`

Specifies the static timing analyzer (STA) and power analyzer to be used for validation. When power analysis is requested (see -power), the STA timer is mapped to its associated power tool as per the table below:

	STA	Power Analyzer
`primetime`	PrimeTime	PrimeTime-PX
`ets`	ETS (Encounter Timing System)	EPS (Encounter Power System)
`tempus`	Tempus	Voltus

`primetime` Uses PrimeTime from Synopsys.

Note: Setting `-timer` to primetime uses PrimeTime as a timing signoff solution and PrimeTime-PX for the power simulator.

`ets`	Uses Encounter Timing System from Cadence.
`tempus`	Uses Tempus Timing Signoff Solution from Cadence. **Note:** Setting `-timer` to Tempus uses Tempus as a timing signoff solution and Voltus as the power simulator.
`timer_cmd_option`	A general timing option to generate timer command string.

`-timing_mode <gba | pba | pba_wave>`

Allows the timing validation to be done using graph-based timing analysis (default), path based analysis or path based analysis with waveform propagation.

Path based analysis (`pba`) is less pessimisitc than graph-based but may increase timer runtimes. Path based analysis with waveform propagation (`pba_wave`) effects are important for advanced nodes (28nm or below) when there are long interconnects or nets with many fanout stages.

When `-timing_mode pba_wave` is set for PrimeTime it will use CCSN data from the library to augment CCS timing and hence will use a PrimeTime SI license. We recommend `driver_waveform` information be present in the input library for this mode. For `pba_wave` to have some impact the `chain_length` should be at least 3, the `wire_res` at least 100ohms and `wire_cap` at least 10ff or alternatively the `fanout` should at least 3.

The `pba` and `pba_wave` timing modes are also supported by Tempus and ETS using version 13.1 or later.

`gba`	Graph-based analysis (Default)
`pba`	Path-based analysis
`pba_wave`	Path-based analysis with waveform propagation.

`-timing_models {<ccs | nldm | ecsm | none>}`

Specifies a list of timing models to check. Default: `{ccs nldm}` for PrimeTime and `ecsm` for ETS.

For example, to perform only constraint validation use `-constraint -timing_models` {`none`}.

`ccs`	Checks CCS models.
`nldm`	Checks NLDM models.
`ecsm`	Checks ECSM models.
`none`	Turns off timing validation.

`-transition` — `R`eports a timing comparison for both delay and transition. Default tolerance for transition is 1e-11 (10ps) and 0.1 (10%).

Report a timing comparison that includes both delay and transition

`-user_env` — Specifies a file containing environment variables. This file is sourced to set up the required environment. The value must include the complete path to the file.

`-user_tcl_file <filename>` — Specifies a Tcl file to source in the `-timer` or `-power` tool run script. The Tcl file will be sourced from `validate_library` at the begining of the processing.

`-verbose` — Generates a report showing every comparison of all data, including those that do not exceed a tolerance.

`-wire_cap <`*`value`*` | min | mid | max>`

Sets the wire capacitance between cells to the min, mid, or max load from the delay table for the arc under test, or to a specific value (in Farads). Default: `min`

`value`	Sets the capacitance to the specified value.
`min`	Sets the capacitance to the min load from the delay table.
`mid`	Sets the capacitance to the mid load from the delay table.
`max`	Sets the capacitance to the max load from the delay table.

`-wire_res <`*`value`*`>` — Specifies the wire resistance in ohms for each side of a T network between cells. Default: 0.01 ohms

`-xtalk < min | max>`

Enables validation of the noise model in the input library for crosstalk delay analysis. The input library should contain advanced noise model data in either CCSN or ECSMN format. Alternatively, a cdb file (`-cdb`) for crosstalk analysis can be provided if using ETS (`-timer ets`).

`min`	Validate the impact of crosstalk decreasing path delay.
`max`	Validate the impact of crosstalk increasing path delay.

{*`library_name(s)`*} (Required positional argument) Library files to validate.

The `validate_library` command performs library timing and power validation. This either validates all cells in the library, or the cells designated by the `-cells` and `-exclude` arguments. Validation involves comparing a SPICE simulation using the `-extsim` simulator against reports generated by a static timer using the given set of libraries.

Validation is performed as a four phase process:

1. *Netlist generation phase*: During this phase, the tool generates all the required netlists and input files to run the SPICE simulations and the static timing analyses. The circuit used for validation can be a single cell instance or a series of cell instances of the length specified by `-chain_length`. Between each cell instance is a T network consisting of two resistors (with the resistance specified by `-wire_res`) with an intermediate capacitance (specified by `-wire_cap`).

   Every input-to-output arc for each cell is instantiated as a unique chain or "supercell." For example, a three input, two output cell results in up to six unique supercells. For combinational cells, the output pin is chained to the same input on the next cell in the series. Sequential cells and cells with mixed clock to data or data to clock paths are restricted to a single cell instance in the chain (`chain_length=1`). The exception is, clock gaters where a fully clocked path or a fully data path can have a chain length greater than 1. Supercells are named as follows:

   `<cellname>__X<chain_length>__<input_pin>__<output_pin>`

   The netlists are stored in the directory `dir/netlists`. The input files for timing analysis are stored in `dir/timer`.

2. *SPICE simulation phase*: During this phase, the tool generates vectors and performs distributed simulation of each cell-chain, extracting the delay and power from the input to the output of each chain. The distribution of the simulations is controlled across multiple CPUs on a single machine by using the `-thread` argument, and across a computer

network by specifying the client machines to use via `set_client` commands or the `packet_clients` parameter. Every slew and load combination that is defined for every arc of each cell is simulated. Alternatively, a specific set of slews and loads can be specified to be used for all cells.

The **read_ldb** command can be used to recover from network failures during the simulation phase. An ldb (library database) is created under the validation directory in the file <libname>.ldb.gz. If the simulation phase fails, a new run that starts where the previous one stopped can be done by using read_lib <dir>/<libname>.ldb.gz before validate_library.

3. *Timer phase*: During this phase, the machine performs static timing or power analysis using the analyzer specified by the `-timer` argument.

   All the data files created to perform the timing analysis are stored in directory `dir/timer`. All the path reports generated from the timing analysis are stored in `dir/timer/REPORTS`. Files with `nldm` in their name refer to NLDM analysis while `ccs` and `ecsm` refer to CCS and ECSM analysis. The static timing results are summarized into a Liberty-like format in the file `dir/timer/<`*`libname`*`>.<`*`delay_model`*`>.<`*`timer_type`*`>.lib` where *`delay_model`* is one of `nldm`, `ccs`, `ecsm`, or `ssta` and *`timer_type`* is one of `pt`, `ets`, `tc`, or `gt`.

4. *Compare phase*: During this phase, the results from the simulations are compared against the path reports from running the timer phase. The comparison is performed by comparing the Liberty-like libraries created by the SPICE simulation and timer phases.

   The detailed comparison results are stored in the file:

   `dir/<`*`libname`*`>.<`*`delay_model`*`>.<`*`timer_type`*`>_<`*`spice`*`>.cmp.<`*`format`*`>`

   (where *`format`*= `xls` or `txt`).

   A file for graphical display of the results using the supplied `lcplot` utility is written to:

   `dir/<`*`libname`*`>.<`*`delay_model`*`>.<`*`timer_type`*`>_<`*`spice`*`>.gui`

   If the format is `htm` then the comparison results are also stored in:

   `dir/<`*`libname`*`>.<`*`delay_model`*`>.<`*`timer_type`*`>_<`*`spice`*`>_html`

   Open file `index.htm` in this directory to view the results in a web browser.

When validating cells with tri-state arcs the pin capacitance for the tri-state output is subtracted from the load index for those arcs. To disable this subtraction, set the `adjust_tristate_load` variable to 0, so validation is performed using the load index as-is. For more information, see "adjust_tristate_load" on page 119.

If a simulation fails and the `-extsim` argument is enabled, the output deck is written to a tarred and gzipped file named `<`*`supercell`*`>.tgz` in the `dir/extsim` directory. SPICE

decks for passing simulations are saved only when the `extsim_save_passed` variable is set. The results from all the simulations are converted into a Liberty-like format in the directory `dir` in the file `<libname>.<extsim>.lib`.

The report that compares the results from SPICE simulation and the power analyzer are stored in the files `dir/<libname>.<power_model>.<timer_type>_<spice>.cmp.<format>` (example: `./VAL/mylib.nlpm.pt_hspice.cmp.txt`).

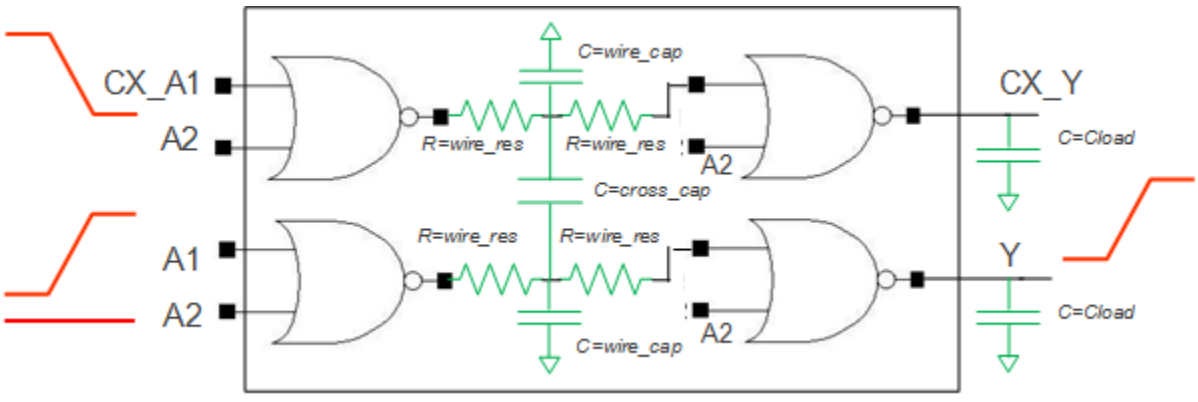
The power analysis reports are stored in the directory `dir/timer/REPORTS`. The leakage results have a `.lkg` postfix, the hidden power results have a `.hdn` postfix and the switching power results have a `.pwr` postfix.

The timing, power and/or constraint validation is performed for every arc for every cell for each logic state defined in the `library`. Use the `set_operating_conditions` command to specify the voltage and temperature to use for the SPICE simulations and the timing analysis runs. If either the voltage and/or temperature specified is different than the voltage or temperature given *in the first library*, then the validation results reflect the impact of voltage and temperature scaling.

If voltage or temperature scaling is performed, warning messages are issued. For example:

```
*Warning* (validate_library) : Voltage scaling will be used as the input library
supply is 1.0 and the operating voltage is 0.9
*Warning* (validate_library) : Temperature scaling will be used as the input
library temperature is 125 and the operating temperature is 55
```

For crosstalk analysis, a circuit composed of two identical parallel paths with the internal node on each path connected via a coupling capacitor (`-cross_cap`) is created. Each path consists of a driver and receiver both using the cell being validated. For example, for the cell `nor2` the following circuit is created.



When `-xtalk` is set to `max`, the active inputs to each path will switch in the opposite direction causing each path to slow down as shown in the above figure. However, when `-xtalk` is set

to `min`, the active inputs to each path will switch in the same direction to speed up both paths. Crosstalk analysis is supported only for combinational cells using `PrimeTime`, `Tempus`, or `ETS`. The delay from Spice is compared against the delay from the timer for the created circuit. The results are stored in the `dir/REPORTS` directory in the file named:

`<libname>.<ccs|ecsm>_xtalk_<max|min>.<timer_type>_<spice>.cmp.txt.`

Performance Considerations

If a pin is defined as a duplicate of another pin using the `define_duplicate_pins` command it is skipped because validating the original pin should be sufficient. By skipping "duplicate" pins the validation runtime can be greatly improved. For example, if you use `define_duplicate_pins cell q0 {q1 q2 q3 q4 q5 q6 q7}`, only arcs ending at `q0` are validated. Arcs ending at `q1` thru `q7` are skipped because they are assumed to be the same as arcs ending at `q0`. This improves runtime for this cell by 8X.

Only *one* `validate_library` command is allowed in a Liberate LV run.

validate_lvf_data

Use this command to perform static checks on the LVF data in a library.

Arguments

`-cells {`*`cell_names`*`}`

Specifies a list of cells to validate. Default: `all cells`

`-exclude {`*`cell_names`*`}`

Specifies a list of cells to be excluded from lvf_data checking.

`-expand_buses` Expand bus groups and bundles into separate pins.

`-range {list}` Check whether the LVF values are within the data range given in library units. For example, 0.0 1.0. Default: {} (`no range`).

`-reltol <{`*`list`*`}>`

Specifies a relative tolerance limit for comparing the LVF data to nominal data. Default: `0.001 1.0 or (0.1% 100%)`

`-report <`*`filename`*`>` Specifies the filename to be used for the LVF data report. Default: `stderr`

`-type {delay constraint}`

List of data types to check: delay, constraint, setup, hold, removal, recovery, nonseq_setup and nonseq_hold. Default: {delay constraint}.

`-verbose` Report all LVF data entries regardless of pass or fail.

`-warn_zero` Generates warning if any LVF data value is zero. By default, an error is reported if a value is zero.

This command can be used to check the current library that has been read using **read_library**. The following checks are performed.

- Each delay and constraint arc value have corresponding LVF data values.
- The ratio of the LVF value to the equivalent nominal value should be greater than or equal to the first value in the reltol list (default 0.001) and should be less than or equal to the last value in the reltol list (default 1.0). If the ratio fails either test, the LVF value is flagged as a warning and a "< W" note appears near the end of the reported line.

- Each LVF value should be with the range of absolute values specified by the -range list. If a range is specified and a value is out of range, it is flagged as an error and a "< E" note appears near the end of the reported line.
- Each LVF value is checked to ensure it is not zero. If the `-warn_zero` option is used, the zero value is flagged as a warning, otherwise it is flagged as an error.

The `-cells` option restricts checking to only the named cells. If `-cells` is used with `-exclude`, all cells except the named cells are checked. The `-verbose` option reports every nominal and LVF value plus their ratio regardless if the value passes or fails. All failing lines are tagged with the "<" character. The `-expand_buses` option checks and reports each bus bit while the default value of the option only checks the values that occur directly under the bus. The `-type` option can be used to restrict the checking to only delay arcs, constraint arcs, or certain types of constraint arcs (setup, hold, recovery, removal, nonseq_setup, nonseq_hold).

Sample Report

```
*Info* (validate_lvf_data) : >>> Start checking cell TC2INVXC. Checking OCV values
are within 1.0-50.0% of nominal values.

TC2INVXC:a->yb: "ocv_sigma_rise_transition early" Vs "rise_transition"
        +--------+--------------+--------------+-----------+
        |  index |      nom     |      ocv     |   diff%   |
        +--------+--------------+--------------+-----------+
        | (1, 1) |     0.013155 |     0.000224 |     1.71% |
        | (1, 2) |     0.022754 |     0.000901 |     3.96% |
        | (2, 1) |     0.022854 |     0.000116 |     0.51% |<W <1.0%
        | (2, 2) |     0.029454 |     0.000087 |     0.29% |<W <1.0%
        +--------+--------------+--------------+-----------+

TC2INVXC:a->yb: "ocv_sigma_rise_transition late" Vs "rise_transition"
        +--------+--------------+--------------+-----------+
        |  index |      nom     |      ocv     |   diff%   |
        +--------+--------------+--------------+-----------+
        | (1, 1) |     0.013155 |     0.000185 |     1.41% |
        | (1, 2) |     0.022754 |     0.000843 |     3.70% |
        | (2, 1) |     0.022854 |     0.000105 |     0.46% |<W <1.0%
        | (2, 2) |     0.029454 |     0.000043 |     0.15% |<W <1.0%
        +--------+--------------+--------------+-----------+
```

validate_monotonicity

Use this command to check the following data in the current library to ensure the tables are monotonically increasing with respect to output load:

cell_rise	retaining_rise
cell_fall	retaining_fall
rise_transition	retain_rise_slew
fall_transition	retain_fall_slew
mpw	

Arguments

`-abstol <value | {list}>`

Specifies an absolute tolerance limit for each validation. The argument accepts a single value or a paired list of type and value. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: for delay `0e-12`; power `0e-15`; trans `0e-11`; constraint `0e-11`; leakage `0e-12`

`-cells {cell_names}`

Specifies a list of cells to check. This argument supports the use of a wildcard. Default: all cells are checked

`-exclude {cell_names}`

Specifies a list of cells to be excluded from monotonicity checking.

`-index1_range <range>`

Specifies a range of indices that designate the values to be validated. The range is either two values separated by a "`-`", (e.g. "`1-3`" to compare the first three indices) or a single value (e.g. "`2`" to compare the second index only.) Default: use all index 1 indices

`-index2_range <range>`

Specifies a range of indices that designate the values to be validated. The range is either two values separated by a "`-`", (e.g. "`1-3`" to compare the first three indices) or a single value (e.g. "`2`" to compare the second index only.) Default: use all index 2 indices

`-reltol <`*`value`* `|` `{`*`list`*`}>`

Specifies a relative tolerance limit for each validation. The argument accepts a single value or a paired list of type and value. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: for delay `0.0`; power `0.0`; delay_variation `0.0`; trans `0.0`; constraint `0.0`; leakage `0.0`

`-report <`*`filename`*`>` Specifies the filename to be used for the report detailing non-monotonicity. Default: `stderr`

The warnings or errors indicate the bad table entry, the values involved, and the arc type including the `when` condition.

`-skip {three_state_disable | three_state_enable}`

Specifies a list of timing types that are not to be checked for monotonicity. Default: none of the three-state timing arcs are skipped

`three_state_disable`

Does not check three_state_disable timing arcs for monotonicity.

`three_state_enable`

Does not check three_state_enable timing arcs for monotonicity.

For example:

```
validate_monotonicity -skip {three_state_disable
   three_state_enable}
```

`-slew` Specifies that the monotonicity checks are also performed with respect to input slew.

Example

```
read_library mono.lib
validate_monotonicity -slew -report mono.txt
```

The errors and warnings, for example, might look like this.

```
*Warning* (validate_monotonicity): Non-monotonic (by load) rise_transition values:
(3, 4) 0.35 < 0.37 for DFFX1:CLK->Q
*Error* (validate_monotonicity): Non-monotonic (by load) cell_fall values: (2, 5)
0.254 < 0.257 for DFFX1:CLK->Q
```

validate_scaling

Use this command to check if a set of libraries is suitable for voltage and temperature scaling.

Arguments

`-db {`*`list`*`}`	Specifies a list of compiled library databases for PrimeTime. Default: "" (always compile the input library)
`-dir <`*`directory`*`>`	Specifies the directory to store the created files. Default is `./SCALE`
`-timer <primetime | ets | tempus>`	Specifies the timing analyzer to use. Default: `tempus`
`{`*`libraries`*`}`	(Required positional argument) Specifies a list of libraries to check for scaling. Default: ""

validate_sdf

Use this command to check SDF annotation for a specified library, timer, and logic simulator.

Arguments

`-cells {`*`cell_names`*`}`

Specifies a list of cells for SDF annotation. This argument supports the use of a wildcard. Default: all cells are used for SDF annotation.

`-db {`*`library_name.db(s)`*`}`

Specifies one or more Synopsys compiled library database files to use for SDF validation. Default is to recompile the input libraries.

This avoids having to recompile the library for each run of `validate_library` when the timer is PrimeTime.

`-dir <`*`directory`*`>`	Directory name to store intermediate files used in the comparison. Default: `SDF`
`-exclude`	Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from SDF annotation.

`-keep_files` Specifies that output files remaining from previous runs should be kept and not automatically deleted. Default: files are deleted when the `validate_sdf` command runs.

This argument can provide a small performance improvement when re-running, but be careful not to confuse old reports with the current run. (Old reports can remain if the `validate_sdf` command fails to run, which can happen if another tool, such as the timer, is not available).

`-logsim <vcs | vcsi | modelsim | ncverilog>`

Specifies the logic simulator to use. Default: `vcsi`

`vcs`	Specifies the VCS simulator.
`vcsi`	Specifies the VCSI simulator.
`modelsim`	Specifies the ModelSim simulator.
`ncverilog`	Specifies the NC-Verilog simulator.

`-logsim_options <"`*`simulator_options`*`">`

Logic simulator options, such as compiler directives. Default: *""*

`-no_edge` Excludes the posedge or negedge constructs from the generated SDF. Default: includes posedge and negedge constructs.

`-primitives <"`*`filename`*`">`

Filename for pre-defined logic simulator primitives. Default: *""*

`-sdf_version <2.1 | 3.0>`

Specifies the SDF version to generate. Default: `"2.1"`

`-timer <primetime | ets>`

Timing analyzer to use. Default: primetime

`<`*`verilog_or_vital_file`*`>`

(Required positional argument) Filename of the library-level Verilog/Vital netlist.

`{`*`library_name(s)`*`}` (Required positional argument) Library files to validate.

The `validate_sdf` command generates a top-level Verilog netlist that instantiates every cell in the library. It then executes the timing analyzer, using the given library and this netlist, and performs SDF-generation.

After the SDF has been generated, the logic simulator is called using the top-level Verilog netlist, the library level netlist and any additional logic simulator primitives.

All the files generated to run the timer and all the log files created by the SDF-generation process are stored in a sub-directory given by the `-dir` argument. Within this directory, two sub-directories are created *dir*/<*timer*> and *dir*/<*logisim*>. The generated SDF is written to *dir*/<*timer*>/<*libname*>.<verilog | vital>.<*sdf_version*>.sdf. The output from the logic simulator run detailing any SDF annotations errors is written to *dir*/<*logsim*>/<*logsim*>.<verilog>.*sdf_version*.log or *dir*/<*logsim*>/<*logsim*>.<vital>.<*sdf_version*>.log.

Example

```
# Test 2.1 and 3.0 SDF annotation for VCS, NC-Verilog and ModelSim
foreach logsim {vcs ncverilog modelsim} {
    foreach sdf_version {"2.1" "3.0"}
        set sdfv "21"
        if {$sdf_version == "3.0"} {set sdfv "30"}
        validate_sdf -dir SDF_$logsim -sdf_version $sdf_version\
            -logsim $logsim test_$sdfv.v test.lib
    }
}
```

5

Liberate LV Variables

This chapter describes the Liberate LV specific variables that impact library validation.

Note: Liberate LV specific variables are set using the `set_var` command.

a...	
adjust_tristate_load	
e...	
extsim_cmd	extsim_save_driver
extsim_cmd_option	extsim_save_passed
extsim_deck_header	extsim_tar_cmd
extsim_deck_style	extsim_timestep
extsim_interactive	extsim_tran_append
extsim_option	
h...	
heartbeat_initial_timeout	heartbeat_timeout
l...	
lic_max_timeout	logic_and
lic_queue_timeout	logic_or
m...	
msg_level	
p...	
packet_arc_notification_interval	packet_client_timeout
packet_arc_notification_limit	packet_log_filename
packet_arc_notification_list	packet_mode

packet_clients	packet_rsh_mode
packet_client_resubmit_count	predriver_waveform
r...	
rcp_cmd	rsh_cmd
s...	
set_var_failure_action	spice_delimiter
t...	
timer_command	timer_initial_timeout
u...	
user_arcs_only	

adjust_tristate_load

Use this variable to control whether pin capacitance is added to the load indices on tri-state pins.

Arguments

`<0 | 1 | 2 | 21 | 22>`

Controls whether and how pin capacitance is added to the load indices on tri-state pins. (Default: `1`)

`0` Turns off these pin capacitance adjustments, i.e. the library and template do not add or subtract the tri-state pin capacitance.

`1` Liberate LV adds the pin capacitance of the tri-state pin to each of the load indices when outputting the library. The rise index_2 adds the rise_capacitance and the fall index_2 adds the fall_capacitance. In addition when using the `write_template` command to create a Liberate LV Tcl command file, the tri-state pin rise/fall_capacitance is subtracted from the load indices specified in the input library to create the appropriate `define_template` commands for tri-state pins.

`2` Enables functionality similar to 1 with the following addition: instead of adding the rise_capacitance or fall_capacitance, the pin attribute capacitance is added to the load indices for the index_2 values. When using the `write_template` command, the pin capacitance is subtracted from the load indices specified in the input library to create the appropriate `define_template` commands for tri-state pins. The value of the attribute capacitance can be modified using the `set_pin_capacitance` command.

`21` Enables behavior that is the same as a setting of `1`, but power arc loads are not adjusted.

`22`	Enables behavior that is the same as a setting of `2`, but power arcs are not adjusted.

This variable can be used after `char_library`, but must be used before any models are generated.

Example

```
# Disable adjusting tri-state pin load indices
set_var adjust_tristate_load 0
```

extsim_cmd

Use this variable to override the default commands used by Liberate LV to call external SPICE simulators.

Argument

"*string*"	Command string to be used to call the external SPICE simulator. Default: for Spectre, `"spectre"`

This argument can be used to override the default command used by Liberate LV to call the external simulator. The default Liberate LV commands to call external simulators are:

- For Spectre:

  ```
  $extsim_cmd $extsim_cmd_option sim.sp >& sim.lis
  ```

Example

To override the default command for calling the Spectre simulator you might use commands such as

```
set_var extsim_cmd "spectre2"
set_var extsim_cmd_option "+log mylogfile"
```

where the file `spectre2` contains

```
#!/bin/sh
exec spectre $*
```

extsim_cmd_option

Use this variable to override the default command options used by Liberate LV to call the external simulator.

Arguments

`"string"`	Options to be passed to the `extsim_cmd` variable. Default: for Spectre, `extsim_cmd ""`

extsim_deck_header

Use this variable to overwrite what is written in the header of the external simulator SPICE decks.

Arguments

`"options"`	String of SPICE commands written to external SPICE simulation decks.

extsim_deck_style

Use this variable to request the separation of the netlist and the models from the SPICE decks.

Arguments

`<merge | separate>`	Controls whether to separate the netlist and models from the SPICE decks. Default: `merge`	
	`merge`	Keeps the netlist and models in-line.
	`separate`	Separates the netlist and models into a separate file.

The separated and saved netlist and models are loaded using the `.include` SPICE command.

extsim_interactive

Use this variable to allow the external simulator, when it is enabled with `validate_library -extsim`, to operate in an interactive mode.

Arguments

`<0 | 1>`	Set to enable external simulator interactive mode. Default: `0`	
	`0`	Does not run the external simulator in interactive mode.
	`1`	Runs the external simulator in interactive mode.

Running the external simulator in interactive mode reduces the external simulator start-up time and can be useful in network environments that are not configured for efficient cell validation runs.

Example

```
set_var extsim_interactive 1
```

extsim_option

Use this variable to specify the list of options to be used by external SPICE validation for delay, power, or timing constraint validation.

Arguments

"*options*"	Options to be used for validation with external SPICE. Default: for Spectre, "`save=none`"

The *options* string is passed as a `.option` line in the external SPICE decks that Liberate LV creates for validation.

Examples

```
# Set the .options for external SPICE, leakage and CCS
set_var extsim_option "runlvl=5"
set_var extsim_leakage_option "gmindc=1e-14 pivtol=1e-15"
set_var extsim_immunity_option "runlvl=4 rmax=24"
```

extsim_save_driver

Use this variable to to enable the saving of simulation decks used by the **set_driver_cell** command to characterize the active driver output waveform simulation decks.

Arguments

`<0 | 1>`	Savea SPICE decks from failing active driver simulations. Default: 1	
	`0`	Does not save any active driver simulation decks. This option is available for backward compatibility
	`1`	Saves the simulation decks used for the final driver waveforms. This option works in combination with the `extsim_deck_dir`, `extsim_save_passed`, and `extsim_save_failed` parameters. If the saving of decks is not enabled while using these parameters, no decks are saved.

This variable must be set before the **char_library** command.

extsim_save_passed

Use this variable to control whether output SPICE decks are saved for successful simulations.

Arguments

`<deck | all>`	Save SPICE decks for passing simulations. Default: `deck`	
	`all`	Saves both the input deck and the output listings. As the number of simulation decks is very large, Cadence recommends using this setting only when characterizing a small number of cells. The data is saved in the directory defined by the `extsim_deck_dir` variable.
	`deck`	Saves only the input SPICE deck.

Output SPICE decks are saved only when the `validate_library -extsim` argument is enabled.

extsim_tar_cmd

Use this variable to control the compression of output SPICE decks.

Arguments

`"`*`string`*`"`	Command used to compress the output SPICE decks. Default: `"tar zcf"`

Set this variable to an empty string (`""`) to disable compression.

Example

```
# Disable SPICE deck compression
set_var extsim_tar_cmd ""
```

extsim_timestep

Use this variable to set the time step for the external SPICE simulator.

Arguments

<value>	Specifies the time-step to use for external SPICE simulation. Default: `1e-12s` (1ps)

Example

```
# Set the time step for external SPICE
set_var extsim_timestep 2e-12
```

extsim_tran_append

Use this variable to add additional options to the `.tran` statement.

Arguments

"*options*"	Additional options to append to `.tran`. Default: `""`

Example

```
# Set conservative mode for Spectre
set_var extsim_tran_append "errpreset=conservative"
```

heartbeat_initial_timeout

Use this variable to specify a time limit to wait for a client process to communicate with the master Liberate LV job.

Arguments

<time>	Specifies the time, in seconds, that the server waits for the first client to communicate back to the server. Default: `3600` (1 hour)

When *time* is exceeded, the Liberate LV server issues a warning that the client has failed to start and then restarts the `heartbeat_initial_timeout` timer. This situation could occur, for example, due to network problems.

Example

```
# Set the heartbeat initial timeout to 2 hours
set_var heartbeat_initial_timeout 7200
```

heartbeat_timeout

Use this variable to specify how long the server machine should wait for a response from a client before being released by the server machine.

Arguments

<*time*>	Specifies the time, in seconds, that a client machine can be inactive before being released by the server machine. Default: `300` (5 minutes)

This variable enables recovery from machine failures during distributed validation. It controls how long the server machine waits for a response from a client before releasing that client. If a client hangs, it is not used for the remainder of the validation run. In addition, the task (a collection of arc simulations) being performed by the failing client is re-submitted to another client. If the re-submission of this task causes another client to hang then this task is skipped. At the end of the validation run, any cells that are not fully simulated are reported.

Example

```
# Set the heartbeat timeout to 10 minutes
set_var heartbeat_timeout 600
```

lic_max_timeout

Use this variable to specify how long Liberate LV waits for licenses.

Arguments

<value>	Specifies the duration of time, in seconds, to wait for the required licenses to be acquired. Default: `86400`

When starting up, Liberate LV attempts to check out all the licenses that are needed. For a server, 1 server license is needed. For a client, Liberate LV needs 1 client license for each thread (see `char_library -thread`). If `ALTOS_QUEUE` is set and if only one license is needed, Liberate LV waits until a license is available and then starts running. If `ALTOS_QUEUE`

is set and more than 1 license is needed, Liberate LV waits until the timeout or until it has all of the licenses it needs for all threads. When the timeout ends, if Liberate LV has at least 1 license, it stops waiting and starts the execution with as many licenses as it has. If Liberate LV has no license at the end of the timeout, it resets the timeout clock and begins waiting again for licenses. After the execution begins, Liberate LV stops looking for additional licenses.

For example, if `ALTOS_QUEUE` is set to 1 along with `char_library -thread 4`, and if there are only 2 (mix-and-matched `Liberate_LV` and `Liberate_LX_Client`) licenses available at the beginning, then Liberate LV remains in a wait-and-check queue for an additional 2 licenses. As soon as the additional 2 client licenses are checked out successfully, Liberate LV starts execution with 4 simulation threads. If, however, there are no additional licenses checked out at the end of the timeout, Liberate LV starts execution with only 2 simulation threads.

If `ALTOS_QUEUE` is set to 0 or is not set, Liberate LV does not wait for licenses. Instead, it checks out as many licenses as it can (not exceeding the number it needs) and begins execution. If no licenses are available, Liberate LV terminates.

The shell environment variable ALTOS_LIC_MAX_TIMEOUT will override the value set by this variable in the Tcl file. For more information, see ALTOS_LIC_MAX_TIMEOUT.

lic_queue_timeout

Arguments

`<value>`	Specifies the amount of time, in seconds, to wait for the required licenses to be acquired. Default: `60` (seconds)

The shell environment variable ALTOS_LIC_CHECK_ALT_TIMEOUT will override the value set by this variable in the Tcl file. For more information, see ALTOS_LIC_CHECK_ALT_TIMEOUT.

logic_and

Use this variable to specify the characters to use for denoting logic AND in function comparisons.

Arguments

"*string*"	Specifies the characters to use for denoting logic AND in function comparisons. Default: `" * "` (*Notice the space on both sides of **)

Example

```
# Set the logic AND to &&.
set_var logic_and "&&"
```

logic_or

Use this variable to specify the characters to use for denoting logic OR in function comparisons.

Arguments

"*string*"	The characters to use for denoting logic OR in library attributes or in function comparisons. Default: `" + "` (*Notice the space on both sides of +*)

Example

```
# Set the logic OR string to |
set_var logic_or "|"
```

msg_level

Use this variable to control the quantity of error and warning messages that are issued.

Arguments

`<0 | 1>`	Controls the verbosity of error and warning messages. Default: `0`	
	`0`	Outputs error messages and useful warning and informational messages.
	`1`	Outputs all messages. **Note:** This setting can output a lot of messages (some of which may not be helpful) making it difficult to determine which messages are important.

packet_arc_notification_interval

<value>	Specifies the minimum time interval between two informational notifications (see `packet_arc_notification_list`). The range of value is between 0 to 72000. Default: `600` (in Seconds = 10 minutes)

Example

Request no more than one informational notification per hour.

```
set_var packet_arc_notification_interval 3600
```

The above example requests no more than one informational notification per hour.

This variable must be used before the `char_library` command.

packet_arc_notification_limit

<value>	Specifies the maximum number of informational notifications per run. This variable is effective when the `packet_arc_notification_list` has been set. The specified value should be between 0 to 100. Default: `10`

This variable must be used before the `char_library` command.

Example

```
set_var packet_arc_notification_limit 5
```

The above example limits the notifications to no more than 5.

packet_arc_notification_list

<string>	Sets the e-mail addresses or SMS equivalent e-mail addresses that can receive notifications. Multiple e-mails or SMS numbers can be specified by using a comma-separated list. By default, no notifications are sent. You can set this variable to a valid e-mail address to enable notifications to that address. Default: `""` (empty list)

Requirements:

- The main Liberate AMS job must run on a machine that is able to send e-mails.
- Any SMS numbers provided for notifications should be able to receive messages by e-mail. Some carriers block this ability to prevent spam messages.

This variable must be used before the `char_library` command.

Example

```
set_var packet_arc_notification_list "1111111111@mms.att.net,\
2222222222@messaging.sprintpcs.com,3333333333@tmomail.net,abc@def.com"
```

The above example has three SMS numbers (ATT/Sprints PCS/TMobile) and one e-mail address.

packet_clients

Use this variable to enable parallel packets mode and to specify the number of machines to be used for distributed processing.

Arguments

`<0 | integer>`	Enables parallel packets mode and specifies the number of machines to be used for distributed processing. Default: `0` (Packet-mode off)	
	`0`	Sets parallel packet mode off.
	`integer`	Enables parallel packet mode and sets the number of machines to be used.

Note: If your flow uses the `write_vdb` command, you must set parallel packet mode to off.

This variable must be used before `char_library`.

packet_client_resubmit_count

Use this variable to specify the number of times a failed LSF job is resubmitted.

`<number>`	Specifies the number of times a failed LSF job is resubmitted. Default: `0`

Liberate LV also checks the LDB to make sure it contains data from the job. If no data was generated, Liberate LV resubmits the job. This variable must be used before `char_library`.

packet_client_timeout

Use this variable to specify a timeout limit for client machines on the network.

Arguments

<value>	Sets a timeout value in seconds for client machines on the network. Default: `86400` (1 day in seconds)

If a packet client log file has not been updated for more than the number of seconds specified, Liberate LV assumes that packet has died. (This can occur because of a machine crash, or a signal such as `kill -9` that cannot be trapped.) If a packet client is assumed to be dead, the server does not wait and moves on to the next client.

This variable must be used before `char_library`.

packet_log_filename

Use this variable to specify the name of the log file to hold characterization statistics.

Arguments

<file_name>	Specifies the name of the log file. Default: `"log"`

To report characterization statistics, you must set *file_name* to match the log file specified in the `rsh_cmd` variable. Cadence recommends using the default name "`log`" and also setting the `rsh_cmd` variable to use "`/log`" as `stdout` and `stderr` filenames.

Note: `"%L"` is not allowed in the *file_name* string.

packet_mode

Use this variable to control the parallel packet distribution mode.

Arguments

`<cell | arc>`	Controls the parallel packet distribution mode. Default: `arc`	
	`arc`	Uses the arc-based mode. **Note:** Arc-based packet mode is in beta with release 12.1.
	`cell`	Uses the cell-based mode.

This variable must be used before `char_library`.

packet_rsh_mode

Use this variable to instruct the server during the arc packet flow (see packet_mode) to automatically kill client jobs if the server job is interrupted.

Arguments

`<lsf | ns | custom>`	Instructs the server during the arc packet flow (see packet_mode) to automatically kill client jobs if the server job is interrupted. Default: custom

This variable is automatically set for the following batch submission commands (see rsh_cmd and set_rsh_cmd):

- bsub
- nc

You can explicitly set the variable to the correct value if the rsh_cmd is pointing to a wrapper script instead of using the commands mentioned above.

This variable must be set before `char_library`.

predriver_waveform

Use this variable to enable the use of a piece-wise linear (PWL) waveform as the pre-driver.

Arguments

`<0 | 1 | 2>`	Uses a PWL waveform as the input driver based on averaging a linear ramp and the equivalent exponential response from an RC network. Default: `0`
`0`	Use a linear map as the input slew.
`1`	Limits the linear ramp to the supply voltage rails. Also overrides any `set_driver_cell` commands.
`2`	The linear ramp is not limited by the supply rail but continues in a linear fashion. Cadence recommends using this setting for CCS format data.

Using this analytical waveform gives a good approximation for real waveforms over a large variety of different input driver/receiver combinations, including fast slews on short wires and slow slews on long wires.

Be aware that

- This variable is disabled when the `predriver_waveform_ratio` variable is set to 0.
- For library validation, if a normalized waveform exists in the library, it overrides the `predriver_waveform` setting.

Example

```
# Use a PWL pre-driver derived from an RC network
set_var predriver_waveform 2
```

rcp_cmd

Use this variable to control which file-copy command is used for copying files from the host to the client machine when using distributed parallel processing.

Arguments

`<scp | rcp | cp>` Controls which file-copy command is used for copying files from the host to the client machine when using distributed parallel processing. Default: `scp`

- `cp` Uses the `cp` (copy) command.
- `rcp` Uses the `rcp` (remote copy) command.
- `scp` Uses the `scp` (secure copy) command.

The `rcp_cmd` and `rsh_cmd` variables are used to control the interface to remote clients when using distributed parallel processing. Before using parallel processing, make sure that the server machine (the machine from which Liberate LV is run) can perform an `rsh` or `ssh` and a `cp`, `rcp`, or `scp` to each client machine without requiring a password or passphrase.

rsh_cmd

Use this variable to control which shell command is used for accessing a remote client when using distributed parallel processing.

Arguments

`<rsh_cmd_str | ssh_cmd_str>`

Controls which shell command is used for accessing a remote client when using distributed parallel processing. Default: `ssh`

- `rsh_cmd_str` Specifies an `rsh` (remote shell) command.
- `ssh_cmd_str` Specifies an `ssh` (secure shell) command.

The `rcp_cmd` and `rsh_cmd` variables are used to control the interface to remote clients when using distributed parallel processing. Before using parallel processing, make sure that the server machine (the machine from which Liberate LV is run) can perform an `rsh` or `ssh` and a `cp`, `rcp`, or `scp` to each client machine without requiring a password or passphrase.

The `rsh_cmd` command string can reference the current client and the command with which to invoke Liberate LV by using `%M` (machine) and `%C` (command). In addition, command line options that appear after the Liberate LV Tcl file name can be passed into the `rsh_cmd_str` string by using `%O` (options). These substitutions can be useful if your system is using a job queuing system.

Example

```
# Set the shell and copy variables for distributed runs
# Set remote file copy to rcp instead of scp
set_var rcp_cmd rcp
# Set remote shell to rsh instead of ssh

set_var rsh_cmd rsh
```

set_var_failure_action

`<warning | error>`	Notifies the `set_var` command how to consider a failure. Default: `warning`	
	`error`	When a `set_var` fails, an error message is issued and subsequent commands which would result in characterization or library generation (for example, compare_library) are suppressed. Subsequent `set_var` commands are still allowed so they can be checked for correctness.
	`warning`	A warning is issued when `set_var` fails. The failed `set_var` is ignored and execution continues.

This variable must be set prior to any other `set_var` command.

spice_delimiter

Use this variable to specify the hierarchy delimiter in the SPICE format netlists loaded into **read_spice**. It can be a single- or multiple-character string. Default: ".".

Arguments

"*string*"	Specifies one or more characters used to indicate hierarchy in the input SPICE netlists. Every character in this variable is treated as a hierarchical delimiter. In the SPICE decks that are written out, the first character in this string will be used as the hierarchical delimiter.

Example

```
# Set the SPICE delimiter to |
set_var spice_delimiter "|"
```

timer_command

Use this variable to specify the command to use to call the timing analyzer.

Arguments

<*string*>	Specifies the command to use to call the timing analyzer. Default: when the `-timer` argument is: ■ `primetime`, the default is `"pt_shell"` ■ `ets`, the default is `"ets"`

The `timer_command` <*string*> variable can be used to change the call to the timing analyzer from the `validate_library` command. For example, a wrapper script could be used to set environment variables before calling the timer to ensure the use of a specific version of the timing tool.

This variable can also be used to deploy a queuing system to call the timer. With this approach, each distinct timing and power model run is sent to the queue. For example, NLDM and CCS timing, and NLPM power and CCSP power can all be submitted in parallel. The `validate_library` command waits for all of the timing analyzer runs to finish before performing the comparison of the timing analyzer results against SPICE simulations.

Example

```
# Set the timer command to use a shell script to call ets
set_var timer_command "my_ets_command"
# Set the timer command to use Sungrid
# Note %O gets expanded to the output log file created by
# validate_library
set_var timer_command "qsub -q q64 -b y -j y -cwd -o %O pt_shell"
```

timer_initial_timeout

Use this variable to specify how long the `validate_library` command waits for the timing analysis job to start on the queue.

Arguments

<time>	Specifies the time, in seconds, that `validate_library` waits for the timer to start on the queue. Default: `3600` (1 hour)

If *time* is exceeded the timer job does not run. This situation could occur because there are not enough timer analyzer licenses available. If the timer job does not run, `validate_library` can be re-run later with the `-start_from "timer"` argument to complete the validation task.

Example

```
set_var timer_initial_timeout 7200 # Wait 2 hours
```

user_arcs_only

Use this variable to have the `validate_library` command validate only arcs that are explicitly defined with `define_arc` commands.

Arguments

`<0 | 1>`	Control whether only user-defined arcs (arcs specified with `define_arc`) are validated. Default: `0`	
	`0`	All the arcs present in the library are passed to the `validate_library` command and all the arcs defined using `define_arc` are validated.
	`1`	Only arcs that are explicitly defined with `define_arc` commands are validated.

Example

```
# Validate user-defined arcs only
set_var user_arcs_only 1
```

6

Library Comparisons

This chapter describes the Liberate LV utilities for comparing libraries.

Comparing two libraries might help you understand, for example, how using new SPICE models would change library characteristics such as leakage power. You might also use these utilities to compare Liberate LV-generated libraries against existing libraries.

Liberate LV provides for both textual and graphical comparisons. The `compare_library` command can be used to generate a text comparison report highlighting outliers that exceed the defined absolute and relative tolerances. Using the `-gui` argument with the `compare_library` command generates a file that is formatted for graphical comparisons. The graphical comparison utility, lcplot, is described in the following section.

lcplot

Use this utility to plot the results of library comparisons.

Arguments

<gui_comparison_file>

Specifies the name of an existing graphical comparison file generated by the `-gui` argument of Liberate LV commands.

Running this utility opens the `lcplot` display tool, providing controls that allow you to plot the comparison file data in various ways. You can use the graphical comparison plots to pin-point library entities that have significant differences, or to confirm expected trends such as slower delays when comparing a library generated at a slow corner versus a fast corner.

Panel Buttons

Fit X	Expands the current graph in the X range to full-scale while keeping the Y range fixed.
Fit Y	Expands the current graph in the Y range to full-scale while keeping the X range fixed.
Fit All	Expands the current graph in both the X and Y ranges to full-scale.
Log - X	Changes the X-axis in an X/Y style graph into a logarithmic scale.
Log - Y	Changes the Y-axis in an X/Y style graph into a logarithmic scale.
Timing	Selects only timing data to display (delay, setup, hold, recovery, removal and trans).
Power	Selects only power data to display.
Leakage	Selects only leakage data to display.
Capacitance	Selects only capacitance data to display (capacitance, fall_capacitance and rise_capacitance).
Style	Selects the style of the graph to display. For more information, see "Graphical Library Comparison Plot Styles" on page 143.
Cell Sel	Opens the Cell Selection form to select cells to display.
Data Sel	Opens the Data Selection form to select data type to display.
Direction	Opens the Direction Selection form to select data-toggle direction (rise, fall).
Close	Closes the lcplot window.

Pull-Down Menus

File – Print	Opens the Print form to print to a file (postscript format) or to a printer, in gray-scale or color.
File – Exit	Closes the lcplot utility.
View – ZoomOut 2	Zooms out by 2X (or, click right in the graphic display window to zoom out by 2X).
View – ZoomIn 2	Zooms in by 2X (or, click left in the graphic display window to select a box to automatically zoom into).

Redraw	Redraws the window to clean up any cursor ghosts.
Help – About	Displays the Version and Copyright notices.

Zooming with the Mouse

Holding down the left mouse button and dragging the rectangle over the plot area zooms into the selected area (when the cursor is in the plot window). A single click of the right mouse key zooms out by 2X (when the cursor is in the plot window). The *fit_x*, *fit_y*, and *fit_all* buttons can be used to fit the data within the window after zooming.

Graphical Library Comparison Plot Styles

There are three styles of plots available: X/Y, Accuracy, and Errorbound.

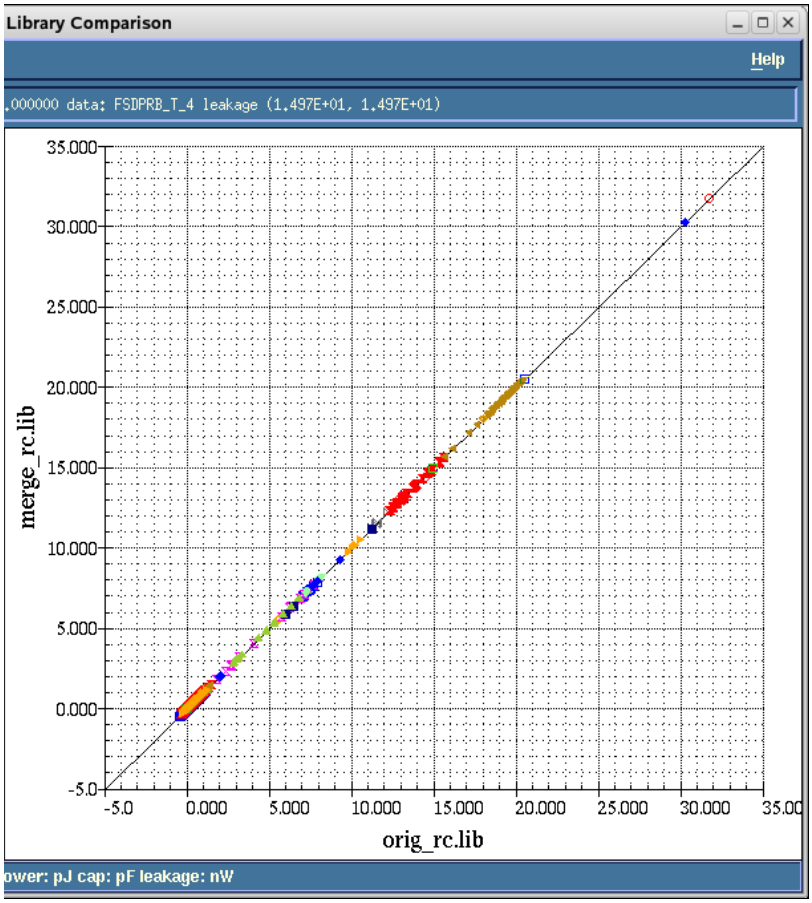
- X/Y style

  The following graphical library comparison shows a scatter-plot where the values from the reference library are given on the X-axis and the values from the comparison library are given on the Y-axis. When the values in the two libraries match, the plotted data points fall exactly on the 45-degree axis.

  By default, this plot type displays all four data types: timing, power, leakage, and capacitance.

  An example is shown below:

Figure 6-1 lcplot GUI - X/Y Library Comparison



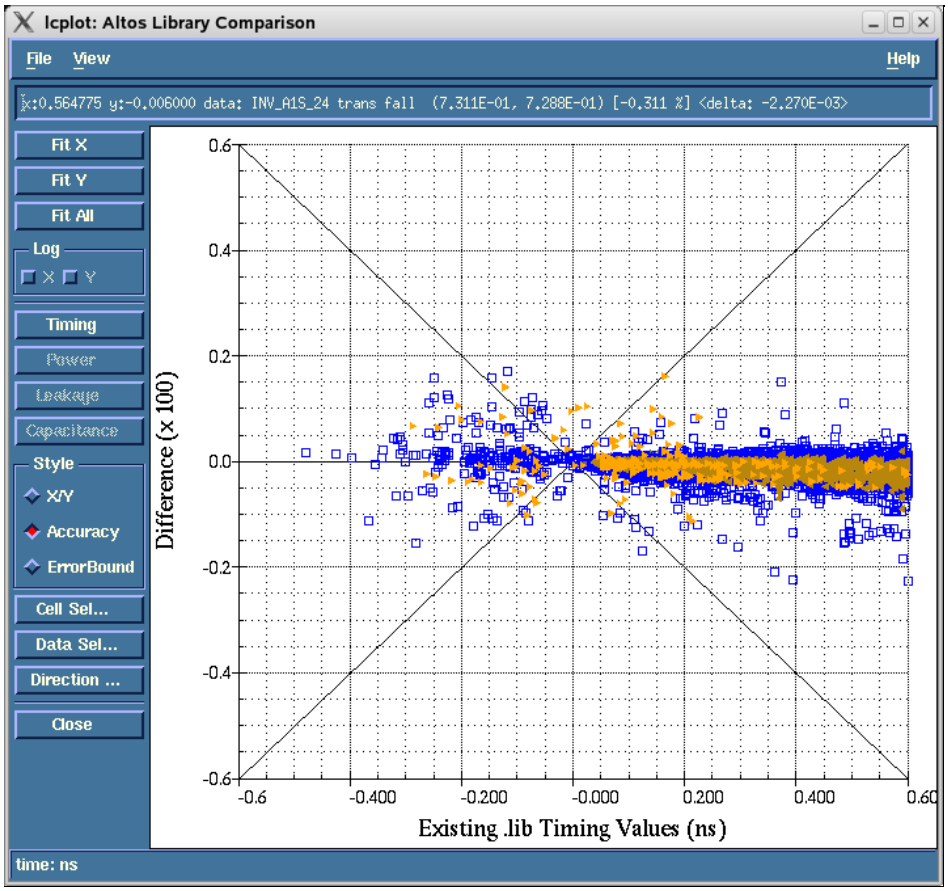
- Accuracy style

  The following graphical library comparison shows an Accuracy plot where the existing value in the reference library is given on the X-axis and the absolute difference in values (scaled up by 100) between the comparison library and the reference library is given on the Y-axis. This plot also includes a +45 degree and a -45 degree axis. These two axes form four triangular regions. The left and right regions represent the data points that fall within 1%. The upper and lower triangular regions represent the data points that are greater than 1% of difference. This plot is useful in determining which data points are greater than 1%. When the mouse is positioned directly over a data point, the actual data from both libraries is displayed in the frame directly above the graph.

  This plot style only applies to timing comparisons.

  An example is shown below:

Figure 6-2 lcplot GUI - Accuracy Library Comparison



- ErrorBound style

  The following graphical library comparison shows an ErrorBound plot where the absolute difference in values between the comparison library and the reference library is given on the X-axis and the difference ratio

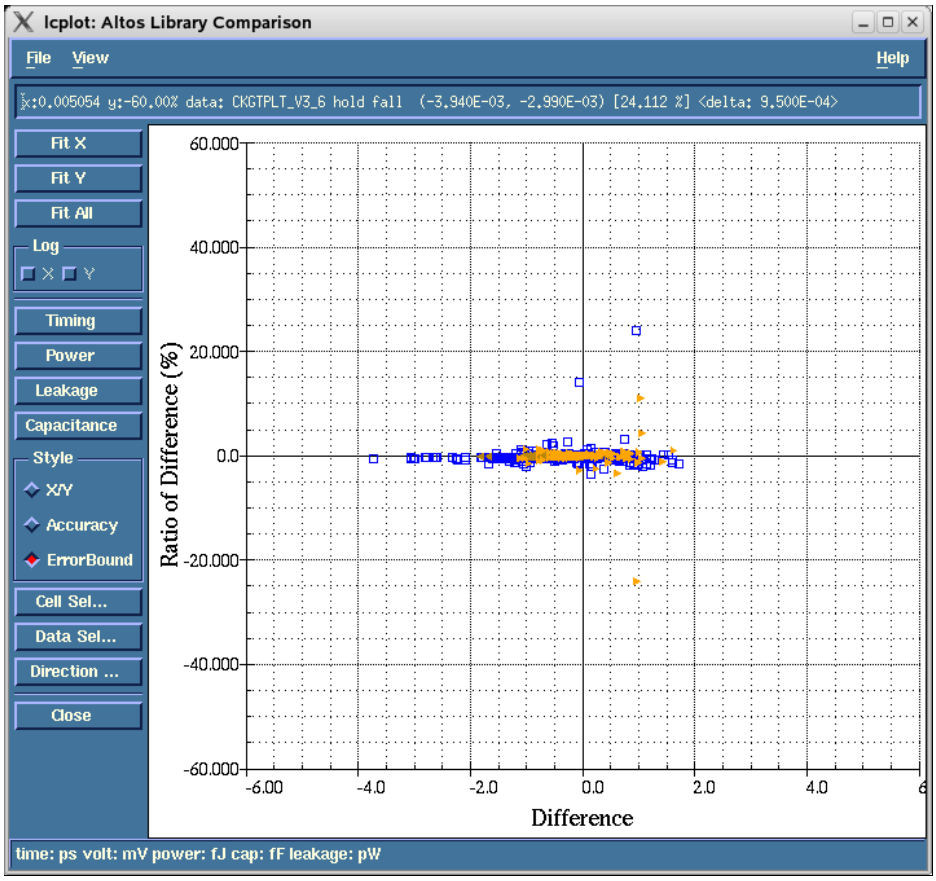
  ```
  ((compVal - origVal)/origVal)*100)
  ```

  of the comparison library to the reference library is given on the Y-axis. If the data point on the graph falls close to the X-axis zero-reference (X=0), then the absolute error is small, so the relative error can be ignored even if it is large. If the data point falls close to the Y-axis zero-reference (Y=0), then the ratio of difference is small and even if the absolute difference is large, this difference can be ignored. When data points do not fall near either of the zero-reference axes, the difference may be significant and should be reviewed. When the mouse is positioned directly over a data point, the actual data from both libraries is displayed in the frame directly above the graph.

  By default, this plot type displays all four data types: timing, power, leakage, and capacitance.

An example is shown below:

Figure 6-3 lcplot GUI - ErrorBound Library Comparison



Data Selection Methods

There are two ways to select data: Cell type selection and data type selection.

- Cell type selection

  To select the cells to compare, click *Cell Sel*. This pops up a selection menu that lists all of the cells in the library. An example is shown below:

Figure 6-4 Cell Selection Menu

To select a cell from the list, click the cell name and then click *Apply*. To select multiple cells, hold down the `Shift` key and select the cell name, followed by *Apply*. To add cells to the selected set, hold down the `Ctrl` key, select the cell names, and click *Apply*. To select all cells, click *All* followed by *Apply*. To unselect all cells, click *None* followed by *Apply*. To quit the cell selection menu, click *Close*.

- Data type selection

  To select the type of library data to compare, click *Data Sel*. This pops up a selection menu that lists all of the available data types. Select a data type such as *delay* from the list and click *Apply*. Multiple data types can be compared at once by holding down the `Shift` key and selecting the data types, followed by *Apply*. The units for each of the values displayed are defined by the reference library. The following data types are available for comparison (assuming they are present in the reference library).

`trans`	Selects transition times data.
`capacitance`	Selects input pin capacitance data.
`power`	Selects switching and hidden power data.
`leakage`	Selects leakage power data.
`fall_capacitance`	Selects pin capacitance for falling transitions data.
`rise_capacitance`	Selects pin capacitance for rising transitions data.
`delay`	Selects transition delays data.
`recovery`	Selects recovery time constraints data.

`hold`	Selects hold time constraints data.
`setup`	Selects setup time constraints data.
`removal`	Selects removal time constraints data.

Figure 6-5 Data Type Selection Menu

A

Deprecated Commands and Variables

This section lists all the variables that are deprecated. These variables are being phased out, and have been replaced by either new commands, new variables, or new behaviors of the tool.

Deprecated Commands

compare_ccsp_nlpm

Use this command to compare the CCSP and NLPM data in a single library and report the differences that exceed the defined tolerances. Comparisons are performed for leakage current to leakage power groups, and dynamic current to internal power groups.

Note: This command should not be used because of the difficulty to accurately compare CCSP, which is a waveform based to NLDM. The command will be removed in a future release. Recommend to use the `validate_library` command with a power tool to validate the power data.

Arguments

`-absolute_average` Reports averages using absolute values.

For example, assuming that one difference is -3ps and another is 5ps, the calculation is this when `-absolute_average`:

is not used. *is* used.

$$\frac{-3+5}{2} = \frac{2}{2} = 1 \qquad \frac{|-3|+5}{2} = \frac{8}{2} = 4$$

`-abstol <`*`value`*`>` Sets the absolute tolerance for the CCSP vs. NLPM error comparison. `-abstol` and `-reltol` define absolute and relative tolerance limits for each comparison. Any comparison that exceeds both these tolerances is considered an outlier and is reported. Default: 0.001 * *`power_unit`* (typically 1nW).

`-cells {`*`cell_names`*`}`

Specifies a list of cell names. Default: all cells

`-exact_match` Compares arcs only when the logic (`when`) conditions are an exact match.

`-exclude` Reverses the meaning of the `-cells` list, so that the specified list of cells are excluded from comparison.

`-format <txt | xls | htm>`

Specifies the format for the output report. Default: `txt`

- `htm` Requests an HTML output format. The default directory name is `./html` and can be changed using the `-group` argument. A one page comparison is generated for each cell group. Open the file `index.html` in a web browser to view the report.
- `txt` Requests a report formatted as standard text.
- `xls` Requests a report in an output format that is suitable for import into Microsoft Excel.

`-group <dirname>` Specifies the name of a directory to store cell comparisons for each cell group. `-group` requests a group-by-group comparison, storing the results in the given directory name. A cell group is determined by the `define_group` command or by the `cell_footprint` attribute. The comparison report for each group is stored in the file `<dir_name>/<group_name>.cmp.txt`. Default: all cells in a single report.

`-gui <filename>` Generates, and specifies a name for, an intermediate file that can be used for graphical comparisons of data with the `lcplot` utility. For more information, see "lcplot" on page 141.

`-lcplot` Uses the `lcplot` utility to display the comparison results graphically. The `-gui` argument is not required because a comparison data file called `<library_lib>.gui` is automatically created.

`-multiple_matches` Reports the results of comparing all arcs that have functional overlap with a reference arc. Default: reports the table that gives the best match.

`-nworst` Specifies how many of the worst delay difference and worst percent difference outliers to include in the summary for each data type (delay or leakage, for example) of each cell. For each cell included in the summary, the worst absolute and relative outlier is reported. Default: 5

`-percent_max_diff` Reports the percent of the maximum difference. Default: reports the maximum of the percent difference.

`-reltol <value>` Sets percentage tolerance for CCSP versus NLPM error comparison. `-reltol` defines a relative tolerance limit for each comparison. Any comparison that exceeds both the `-abstol` and `-reltol` tolerances is considered an outlier and is reported. Default: 0.01 (1%).

`-report <filename>` Specifies the filename to be used for the output comparison file. Default: `<library_name>.cmp.txt`

`-type <lc_2_lp | dc_2_pwr>`

Specifies the type of comparison to perform. Default: both types of comparisons are performed.

`lc_2_lp`	Leakage current to leakage power.
`dc_2_pwr`	Dynamic current to internal power.

`-unmatched` Reports unmatched data entries.

`-verbose`	Generates a report showing every comparison including those that do not exceed a tolerance. The output is written to the `-report` filename. An overall comparison summary is also written to the standard output.
<library_file>	(Required positional argument) Library filename.

Important

Before running this command, be sure to source all variables and commands that pertain to characterizing and modeling power. Typical command and variable statements include:

```
set_var  leakage_add_input_pin
set_var  power_subtract_leakage
set_pin_vdd
set_vdd
```

Also, be sure to source commands and variables associates with power.

validate_library

The `-ocv` option of the `validate_library` command has been deprecated.

Deprecated Variables

bundle_count

Use this variable to specify the number of packets to be used while validating a library.

Arguments

<number>	Sets the number of packets. Default: `0` (Use 1 packet per cell.)

If this variable is not set, Liberate LV uses 1 packet per cell. If *<number>* is set to a non-zero number, Liberate LV divides the number of cells in the library into that number of packets. (See packet_mode for more information.)

This variable must be used before `char_library`.