Milkyway and Physical Implementation
Variables

Version B-2008.06, June 2008

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

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

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

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

Destination Control Statement

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

Disclaimer

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

Registered Trademarks (®)

Synopsys, AMPS, Astro, Cadabra, CATS, Design Compiler, DesignWare, Formality, HSPICE, iN-Phase, Leda, MAST, ModelTools, NanoSim, OpenVera, PathMill, Physical Compiler, PrimeTime, SiVL, SNUG, SolvNet, TetraMAX, VCS, Vera, and YIELDirector are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, Columbia, Columbia-CE, Cosmos, CosmosLE, CosmosScope, CRITIC, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, DesignPower, Direct Silicon Access, Discovery, Eclypse, Encore, EPIC, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, HSIM, HSIM^{plus}, i-Virtual Stepper, In-Sync, iN-Tandem, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Magellan, Mars, Mars-Rail, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, Planet, Planet-PL, Polaris, Power Compiler, Raphael, Saturn, Scirocco, Scirocco-i, Star-RCXT, Star-SimXT, System Compiler, Taurus, TSUPREM-4, VCS Express, VCSi, VHDL Compiler, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (^{SM})

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

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

Table of Contents

check_mw_lib_mode

Determines whether to check libraries for consistency when they are opened.

TYPE

boolean

DEFAULT

true

DESCRIPTION

Determines whether to check libraries for consistency when they are opened. The default value is *true*. When a library and its reference libraries are initially opened, tool checks for consistency between the main library's and its reference libraries' technology and settings. If discrepancies are found, warning messages are displayed in the window. However, tool continues to open the library and its reference libraries.

SEE ALSO

collection_deletion_effort

TYPE

string

DEFAULT

low

DESCRIPTION

The **collection_deletion_effort** variable controls how saved collections are deleted when objects within them are potentially going out of scope. Allowed values are **low**, **medium**, and **high**, indicating to the tool how much effort to expend to preserve a collection (or part of it) when objects are going out of scope.

When the design that owns the objects is removed or closed, the collection is always deleted, and the **collection_deletion_effort** variable has no effect.

If some of the elements in the design are removed and they appear in collections, some of the collections might be deleted. This depends on the value of **collection_deletion_effort** variable, as described below.

• When the effort is **low**, the collection is deleted if any element in the collection was removed.

• When the effort is **medium**, the behavior is the same as when it is **low** (for the current release).

• When the effort is **high**, individual elements of the collection are removed from the collection. The collection is deleted if, and only if, it becomes empty.
The CPU cost for the **collection_deletion_effort** variable increases from **low** to **high**. In most cases, **low** is a satisfactory choice.

To determine the current value of this variable, type

```
prompt> printvar collection_deletion_effort
or
prompt> echo $collection_deletion_effort
```

EXAMPLES

The following example illustrates the effect of using the **low**, **medium**, and **high** values. In it, a design named *M* has two cells named *i1* and *i2*. The following collections are created:

```
prompt> set s1 [get_cells {i* ]
_sel27
prompt> query_objects $s1
```

```
{"i1", "i2"}
prompt> remove_objects [get_cells i1]
_sel28
prompt> query_objects $s2
Warning: collection _sel1 does not exist
```

If you set the **collection_deletion_effort** variable to **high**, the final value of *s2* will be a collection containing cell *i2*.

SEE ALSO

collections(2)

collection_result_display_limit

Sets the maximum number of objects that can be displayed by any command that displays a collection.

TYPE

int

DEFAULT

100

DESCRIPTION

Use this variable to set the maximum number of objects in a collection that any command can display. The default value is 100 objects.

When you issue a command (for example, **add_to_collection**) at the command prompt, the tool implicitly queries the result and displays the number of objects, just as if you had also issued the **query_objects** command.

You can limit the number of objects the query displays by setting an appropriate integer as the value of the **collection_result_display_limit** variable. A value of **-1** displays all objects. A value of **0** displays the collection handle ID instead of the names of any objects in the collection.

To determine the current value of this variable, type the following command string:

prompt> **printvar collection_result_display_limit**

SEE ALSO

collections(2)
printvar(2)
query_objects(2)

hierarchy_separator

Determines how hierarchical elements of the netlist are delimited in reports and searched for in selections and other commands.

TYPE

string

DEFAULT

/

DESCRIPTION

Determines the way in which hierarchical elements of the netlist are delimited in reports and how they are searched for in selections and in other commands. The default value is the forward slash (/). The choice of a separator is limited to the following characters:

- vertical bar (|)
- caret (^)
- the at symbol (@)
- dot (.)
- forward slash (/), the default

In most situations, you can accept the default forward slash (/). However, sometimes the hierarchy character is embedded within other names; and in such cases the search engine might produce unintended results.

The **hierarchy_separator** variable enables you to remove any ambiguity in your search. For example, consider the contents of the following design:

- Hierarchical cell A contains hierarchical cells B and B/C.
- Cell B/C contains cell D.
- Cell B contains cell C.

If you try to search for the string A/B/C/D, the results probably are not going to

be correct. If you set the **hierarchy_separator** variable to have a value of vertical bar (|), however, you give the search engine explicit instructions to search for A | B | C | D.

SEE ALSO

selection(2)

max_model_depth

Enables the tool to "see" netlist objects beyond the current CEL view.

TYPE

string

DEFAULT

0

DESCRIPTION

If you have HTV models, or otherwise depend on accessing pins, nets, or cells in CEL views in the tool that are hierarchically below the CEL that you opened, you must set the **max_model_depth** variable to reflect the depth of the files in the tool. This will be a value greater than **0**, typically **1**.

Note that setting **max_model_depth** to a nonzero value causes the tool to load its symbol table with cells, nets, and pins from other CEL views. This will consume some time and memory. After you have finished all Tcl commands that require the symbol table (for example, the **get_pins** command), you can remove it by using the **remove_symbol_table** command.

Note that each CEL view may contain multiple levels of hierarchically-preserved names. If all of the hierarchically preserved names are in the current CEL view (which is the case for most Astro runs not using HTVs), then you do not have to set this variable.

To determine the current value of this variable, type

prompt> **printvar max_model_depth**

SEE ALSO

remove_symbol_table(2)

mw_lib_lock_mode

Determines what kind of mode should be used to lock library.

TYPE

string

DEFAULT

new

DESCRIPTION

Determines the mode in which a library is locked when it is in use. The default value is *new*. The choice of a lock mode is limited to the following strings:

- new, the default
- old
- both
- neither

The *old* lock mode is used to provide some protection for the library against pre 2003.09 binaries, which only lock the "lib" file. If a library is not a legacy library, it’s recommended to switch off the *old* mode.

SEE ALSO

si_enable_analysis

Enables or disables PrimeTime-SI, which provides crosstalk analysis.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When true, enables PrimeTime-SI, so that the crosstalk-aware timing calculation mode is used by **update_timing** and **report_timing**. By default, PrimeTime-Si is disabled; this variable is set to false.

If you set this variable to true and enable PrimeTime-SI, you must also do the following:

1. Obtain a PrimeTime-SI license. You cannot use PrimeTime-SI without a license.

2. Use **read_parasitics -keep_capacitive_coupling** to read in the coupling parasitics for your design. PrimeTime-SI is useful only if the design has coupling parasitics data.

For complete information about PrimeTime-SI, see the *PrimeTime Signal Integrity User Guide*.

To determine the current value of this variable, type **printvar si_enable_analysis**.

SEE ALSO

read_parasitics (2), **report_timing** (2), **update_timing** (2).

skip_reference_library_read_lock

Determines whether read lock on a reference library should be skipped.

TYPE

boolean

DEFAULT

true

DESCRIPTION

Bypasses the lock on a reference library when you are certain the reference library is read only. No one else can modify the library. If the variable is set to be *true*, read lock on reference library will be skipped. If the variable is set to be *false*, read lock on ref lib will be performed as usually. The default value is *true*.

SEE ALSO

timing_disable_recovery_removal_checks

Disables or enables the timing analysis of recovery and removal checks in the design.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

If you set the value of this variable as **true** (the default value), the tool does not perform recovery and removal timing analysis.

If you set the value as **false**, the tool does perform recovery and removal checks. For a description of the recovery and removal checks, see the man page for the **report_constraint** command.

To determine the current value of this variable, type

prompt> **printvar timing_disable_recovery_removal_checks**
or
prompt> **echo $timing_disable_recovery_removal_checks**

SEE ALSO

report_constraint(2)

timing_enable_gated_clock_checks

Enables gated clock checks to be included in the design.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_enable_gated_clock_checks** variable enables gated clock checks to be included in the design. The default value is **false**.

To determine the current value of this variable, type

prompt> **printvar timing_enable_gated_clock_checks**

SEE ALSO

timing_enable_ideal_network_delay

Enables ideal network delay for use with I/O ports, when clocks are propagated.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_enable_ideal_network_delay** variable enables the tool to use ideal network delay with I/O ports, when clocks are propagated. The default value is **true**.

To determine the current value of this variable, type

prompt> **printvar timing_enable_ideal_network_delay**

SEE ALSO

timing_enable_inter_clock

TYPE

DEFAULT

true

DESCRIPTION

The **timing_enable_inter_clock** variable enables timing for interclock paths. Otherwise the paths are treated as false paths.

To determine the current value of this variable, type

prompt> **printvar timing_enable_inter_clock**

SEE ALSO

timing_enable_mixed_clock_signal

Enables timing for mixed clock and signal paths.

TYPE

(enter data type here)

DEFAULT

false

DESCRIPTION

The **timing_enable_mixed_clock_signal** variable enables timing for mixed clock and signal paths, if you set the value as **true**.

It is necessary to disable this option by setting the value of this variable as **false**, *before* performing clock tree synthesis.

To determine the current value of this variable, type

prompt> **printvar timing_enable_mixed_clock_signal**

SEE ALSO

timing_enable_multiple_clocks_per_reg

Enables the tool to analyze, in the presence of multiple clocks per register.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

This variable enables the Astro timer to analyze, in the presence of multiple clocks per register. When a sequential device is driven by different clocks, you can choose either to restrict the analysis to one clock at a time, or you can allow the tool to analyze both clocks at the same time. If you set the value of this variable as **true**, the tool

• Analyzes all clocks, in a single analysis run.

• Reports all possible interactions between different clocks, subject to any restrictions you specify, such as false paths or case analysis.

The total time it takes for the tool to analyze all clocks is therefore reduced, because it combines multiple runs into a single run.

If you set this variable as **false**, and your design has multiple clocks per register, you should specify a clock, using the **set_data_check -clock** command, for the tool to use.

To determine the current value of this variable, type

prompt> **printvar timing_enable_multiple_clocks_per_reg**.

SEE ALSO

create_clock(2)
create_generated_clock(2)
printvar(2)
set_data_check(2)

timing_enable_preset_clear_arcs

Controls whether the tool enables or disables preset and clear arcs.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When the value of this variable is set as **true**, the tool permanently enables asynchronous preset and clear timing arcs, so that you use them to analyze timing paths. When the value is set as **false** (the default), the tool disables all preset and clear timing arcs.

The tool performs any minimum pulse width checks that are defined on asynchronous preset and clear pins, regardless of the value of this variable.

To determine the current value of this variable, type

prompt> **printvar timing_enable_preset_clear_arcs**

SEE ALSO

printvar(2)
report_timing(2)

timing_enable_scan_enable

Enables timing for the scan enable timing signal. Otherwise the path is disabled.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_enable_scan_enable** variable enables timing for the scan enable timing signal. Otherwise the path is disabled. The default value of this variable is **true**.

To determine the current value of this variable, type

prompt> **printvar timing_enable_scan_enable**

SEE ALSO

timing_enable_trace_mode

Enables the trace mode, which allows you to trace on certain nets.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_enable_trace_mode** variable enables the trace mode, which allows you to trace on certain nets in your design.

To determine the current value of this variable, type

prompt> **printvar timing_enable_trace_mode**

SEE ALSO

timing_enable_useful_skew

Enables storage in the database of useful skew budget values during timing.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_enable_useful_skew** variable enables the tool to store, in the database, useful skew budget values, during timing. The default value is **false**.

To determine the current value of this variable, type

prompt> **printvar timing_enable_useful_skew**

SEE ALSO

timing_ignore_clock_uncertainty

Causes the tool to ignore the clock uncertainty specified in the Synopsys Design Constraints (SDC) format.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_ignore_clock_uncertainty** variable, if you set the value as **true**, enables the tool to ignore the clock uncertainty defined in the SDC. The default value is **false**.

It is useful to specify this variable after propagating clocks, without having to change the SDC.

To determine the current value of this variable, type

prompt> **printvar timing_ignore_clock_uncertainty**

SEE ALSO

timing_ignore_interconnect

Causes the tool to ignore all parasitic wire capacitance during timing.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_ignore_interconnect** variable controls whether the tool ignores all parasitic wire capacitance during timing, including cell-loading capacitance. The default value is **false**. If you want the tool to ignore all parasitic wire capacitance during timing, set the value of this variable as **true**. A setting of **false** (the default) means the tool does not ignore these capacitances during timing.

This variable is a runtime option only and it is not saved in the database.

To determine the current value of this variable, type

prompt> **timing_ignore_interconnect**

SEE ALSO

timing_ignore_propagated_clock

Causes the tool to ignore, for the whole circuit, the propagated clock setting from the Synopsys Design Constraints (SDC).

TYPE

DEFAULT

false

DESCRIPTION

The **timing_ignore_propagated_clock** variable, if you set the value as **true**, enables the tool to ignore, for the whole circuit, the propagated clock setting specified in SDC. Using this variable, you can ignore the propagated clock from the beginning of your design. Then perform the optimization, and then include this information after clock tree synthesis.

To determine the current value of this variable, type

prompt> **printvar timing_ignore_propagated_clock**

SEE ALSO

timing_include_available_borrow_in_slack

Determines whether or not the tool includes available borrow time in slack.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When the value of this variable is set as **false** (the default), the tool measures the slack of a signal arriving before the latch opening edge, relative to the open edge, and does not include available borrow time. The tool considers a signal arriving during the transparent interval as having a slack of zero. Violations are measured with respect to the closing latch edge.

When the value is set as **true**, the tool measures positive or negative slack of any path terminating at the data pin of a transparent latch, with respect to the closing transition at the latch. Available borrow time, considered a component of slack, is typically the duration of the active clock region minus the setup time required. A maximum time borrow set on a latch could decrease this available borrow time.

To determine the current value of this variable, type

```
prompt> printvar timing_include_available_borrow_in_slack
or
prompt> echo $timing_include_available_borrow_in_slack
```

SEE ALSO

printvar(2)
report_timing(2)
set_max_time_borrow(2)

timing_include_io_path

Enables the inclusion of paths related to boundary ports, during optimization.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_include_io_path** variable enables to tool to include paths related to boundary ports, during optimization.

To determine the current value of this variable, type

prompt> **printvar timing_include_io_path**

SEE ALSO

timing_include_lib_max_cap

Includes the maximum capacitance that the library specifies; otherwise, this value is not included.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_include_lib_max_cap** variable includes the maximum capacitance the library specifies, if you set the value as **true** (the default value). Otherwise, this library information is not included.

To determine the current value of this variable, type

prompt> **printvar timing_include_lib_max_cap**

SEE ALSO

timing_include_lib_max_trans

Enables the calculation of maximum transition, based on library values; otherwise, this value is not included.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_include_lib_max_trans** variable enables the tool to calculate maximum transition, based on library values. Otherwise, this value is not included.

To determine the current value of this variable, type

prompt> **printvar timing_include_lib_max_trans**

SEE ALSO

timing_include_non_propagated_nets

Enables the inclusion of all nets for transition and capacitance fixing.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_include_non_propagated_nets** variable enables the tool to include all net for transition and capacitance fixing. The default value is **false**.

To determine the current value of this variable, type

prompt> **printvar timing_include_non_propagated_nets**

SEE ALSO

timing_include_sync_port_phase_delay

Enables the inclusion of the synchronous port phase delay information from the database, during timing.

TYPE

DEFAULT

true

DESCRIPTION

The **timing_include_sync_port_phase_delay** variable enables the tool to include the synchronous port phase delay information from the database, during timing.

To determine the current value of this variable, type

prompt> **printvar timing_include_sync_port_phase_delay**

SEE ALSO

timing_input_port_default_clock

Determines whether a default clock is assumed at input ports for which the user has not defined a clock-specific input external delay.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

This Boolean variable affects the behavior of the tool when it is timing a path from an input port that does not have a clocked input external delay. If you set the value of this variable as **true** (the default value), the tool gives all such input ports one imaginary clock, so that the inputs are constrained. This also causes the clocks along the paths driven by these input ports to become related. If you set the value as **false**, no such imaginary clock is assumed.

To determine the current value of this variable, type

prompt> **printvar timing_input_port_default_clock**.

SEE ALSO

report_timing(2)

timing_remove_clock_reconvergence_pessimism

Enables or disables removal of clock reconvergence pessimism.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When this variable is set as **true**, the tool removes clock reconvergence pessimism from slack calculation and minimum pulse-width checks. This variable replaces the following discontinued options of the **report_timing**, **report_constraint**, and **get_timing_paths** commands:

-report_clock_reconvergence_pessimism
-remove_clock_reconvergence_pessimism

Clock reconvergence pessimism (CRP) is a difference in delay along the common part of the launching and capturing clock paths. The usual causes of CRP are reconvergent paths in the clock network and different min/max delay of cells in the clock network.

The tool calculates the CRP independently for rise and fall clock paths. You can use the variable **timing_clock_reconvergence_pessimism** to control CRP calculation, with respect to transition sense. If the capturing device is a level-sensitive latch, the tool calculates two CRP values, as follows:

crp_open The CRP that corresponds to the opening edge of the latch.

crp_close The CRP that corresponds to the closing edge of the latch.

The required time at the latch is increased by the value of crp_open, thereby reducing the amount of borrowing (if any) at the latch. The maximum time-borrowing allowed at the latch is affected by shifting the closing edge by the value of crp_close. For a more detailed description of a CRP calculation, use the **report_crpr** command.

The tool calculates the CRP in a different way for minimum pulse-width checks. The CRP is given as the minimum of

maximum rise arrival time/minimum rise arrival time
and
maximum fall arrival time/minimum fall arrival time

at the pin where the check is being made.

If the **si_enable analysis** variable is set to **true**, delays in the clock network might

also include delta delays, which result from crosstalk interaction. Such delays are dynamic in nature, which means that they might vary from one clock cycle to the next, causing different delay variations (either a speedup or slowdown) on the same network, but during different clock cycles.

The tool considers only SI delta delays as part of the CRP calculation, if the type of timing check deployed derives its data from the same clock cycle.

Any change in the value of the **timing_remove_clock_reconvergence_pessimism** variable causes full update timing. You cannot perform one report-timing operation that considers CRP and one that does not, without a full update of timing in-between. CRPR is intended for use with reporting commands such as **report_bottleneck**, **report_analysis_coverage**, **report_constraint**, and **report_timing**. Slack attributes, such as max_rise_slack, do not reflect CRPR. CRPR has not been validated with modeling or characterize. CRPR does not support ideal clock latency set on pins or ports. In the case of unsupported features, a more pessimistic value of CRP is used.

To following example illustrates how to turn on CRP removal:

```
prompt> set timing_remove_clock_reconvergence_pessimism true
true
prompt> report_timing
```

SEE ALSO

get_timing_paths(2)
report_analysis_coverage(2)
report_bottleneck(2)
report_constraint(2)
report_crpr(2)
report_min_pulse_width(2)
report_timing(2)
si_enable_analysis(2)
timing_clock_reconvergence_pessimism(3)
timing_crpr_threshold_ps(3)

timing_set_io_clock_latency

Creates 0 clock skew for all I/O ports in the design.

TYPE

DEFAULT

false

DESCRIPTION

The **timing_set_io_clock_latency** variable enables the creation of 0 clock skew for all I/O ports in the design. It does this by automatically choosing a clock arrival time at the port that is equal to the worst of the fanout of the port. The default value of this variable is **false**.

To determine the current value of this variable, type

prompt> **printvar timing_set_io_clock_latency**

SEE ALSO