APPENDIX

Part Four

Trial Exhibit 1441, pages 71-190

-rise_to *to_object*
Similar to the **-to** option, but applies to only rising delays at the constrained pin. You must specify one of **-to**, **-rise_to**, or **-fall_to**.

-fall_to *to_object*
Similar to the **-to** option, but applies to only falling delays at the constrained pin. You must specify one of **-to**, **-rise_to**, or **-fall_to**.

-setup
Indicates that only the setup data check is to be removed. If neither **-setup** nor **-hold** is specified, both setup and hold checks are removed.

-hold
Indicates that only the hold data check is to be removed. If neither **-setup** nor **-hold** is specified, both setup and hold checks are removed.

-clock *clock_object*
Indicates that the data check for the specified clock at the related pin is to be removed. This option applies only if a previous **set_data_check** command used the **-clock** option to specify the same clock; otherwise, this option is ignored.

71

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790542	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
remove_disable_clock_gating_check Restores clock gating checks previously disabled by **set_disable_clock_gating_check**, for specified cells and pins.	Command: remove_disable_clock_gating_check <db:objects> standard SDC command

SYNTAX
 	
string **remove_disable_clock_gating_check** *object_list*
list *object_list*
	option:
	  --license list required licenses
	  --help display command help

ARGUMENTS
 	
object_list
  Specifies a list of cells and pins for whom previously-disabled clock gating checks are to be restored.
	description:
	  This command is the same as standard SDC command.

72

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790543	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
remove_disable_timing Enables the previously disabled timing arcs.	Command: remove_disable_timing Remove set_disable_timing constraints.
SYNTAX string **remove_disable_timing** [-from *from_pin_name*] [-to *to_pin_name*] *object_list* string*from_pin_name* string*to_pin_name* list *object_list*	option: -all remove all disable timing (require) --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --system_default use system default values --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -from *from_pin_name* -to *to_pin_name* Specifies that only the arcs between these two pins on the specified cell or library cell be disabled. object_list Specifies a list of cells, pins, library cells, library cell pins, or ports. The *object_list* must contain only cells or library cells (if **-from** and **-to** are specified).	description: Remove SDC set_disable_timing constraints. Right now, provide one required option -all to remove all disable timing constraints.

73

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790547	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056750
remove_driving_cell Removes port driving cell information.	**remove_driving_cell** Removes driving cell constraints that were set using the *set_driving_cell* Tcl command. Currently, this command removes all driving cell constraints on the specified pins or ports.
SYNTAX string **remove_driving_cell** [-rise] [-fall] [-min] [-max] [-clock *clock_name*] [-clock_fall] *port_list* string*clock_name* list *port_list*	**Syntax** remove_driving_cell *port_list* \ [-rise] \ [-fall] \ [-min] \ [-max] \ [-clock_fall] \ [-clock *clocks*]
ARGUMENTS -rise Removes rise driving cell information. -fall Removes fall driving cell information. -min Removes min driving cell information. -max Removes max driving cell information. -clock *clock_name* Removes the driving cell set relative to the specified clock. -clock_fall Removes the driving cell relative to the falling edge of the clock. The default is the rising edge. port_list Provides a list of input or output ports.	where the arguments have the following meaning: *port_list* — Names of the ports for which to remove the driving cell constraints. [-rise] — Not supported yet. [-fall] — Not supported yet. [-min] — Not supported yet. [-max] — Not supported yet. [-clock_fall] — Not supported yet. [-clock *clocks*] — Not supported yet.

74

remove_from_collection

Removes objects from a collection, resulting in a new collection. The base collection remains unchanged.

SYNTAX

collection **remove_from_collection**
base_collection
x1collection*base_collection*
list *object_spec*

ARGUMENTS

base_collection
Specifies the base collection to be copied to the result collection. Objects matching *object_spec* are removed from the result collection.

object_spec
Specifies a list of named objects or collections to remove. The object class of each element in this list must be the same as in the base collection. If the name matches an existing collection, the collection is used. Otherwise, the objects are searched for in the database using the object class of the base collection.

remove_from_collection

Creates a new collection, starting from a base collection and removing objects that are part of a subtraction set. Neither the base collection not the subtraction set are modified.

Syntax

remove_from_collection *base_collection subtract_collection*

where the arguments have the following meaning:

base_collection	All objects from the base collection that are not in the subtract collection are returned.
subtract_collection	Objects that are not returned.

75

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790559	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056756

remove_input_delay

Removes input delay information from ports or pins.

SYNTAX

```
string remove_input_delay [-clock clock_name] [-clock_fall] [-level_sensitive] [-
rise] [-fall] [-max] [-min] port_pin_list
list  clock_name
list  port_pin_list
```

ARGUMENTS

```
-clock clock_name
        Relative clock; '' for no clock. Use this option to remove only input delay
        relative to one clock.

-clock_fall
        Delay is relative to falling edge of clock.

-level_sensitive
        Delay is from level-sensitive latch.

-rise
        Removes rising input delay.

-fall
        Removes falling input delay.

-max
        Removes maximum input delay.

-min
        Removes minimum input delay.

port_pin_list
        Specifies a list of ports and pins.
```

remove_input_delay

Removes input delay on the ports or the pins that was set using the *set_input_delay* Tcl command. Currently, this command removes all input delay information from the specified pins or ports.

Syntax

```
remove_input_delay port_pin_list \
   [-rise] \
   [-fall] \
   [-min] \
   [-max] \
   [-clock_fall] \
   [-clock clocks] \
   [-level_sensitive]
```

where the arguments have the following meaning:

port_pin_list	Names of the ports and pins for which to remove the input delays.
[-rise]	Not supported yet.
[-fall]	Not supported yet.
[-min]	Not supported yet.
[-max]	Not supported yet.
[-clock_fall]	Not supported yet.
[-clock *clocks*]	Not supported yet.
[-level_sensitive]	Not supported yet.

76

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790562	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056764-65
remove_lib Removes one or more libraries from memory.	**remove_library** First, optionally, detaches unused libraries from projects in memory and then removes libraries that are not attached to any project in memory. A library is considered used by a project if at least one of its library cells is referenced by at least one cell instance in the project, or if the project uses any of its timing data such as SDC, min/max, noise, leakage, or CSS information. A library is considered attached to a project if it is listed in the link path of the project, as set by the *set_link_path* Tcl command or *link_path* Tcl variable. A used library is always attached, but an attached library may or may not be used. You control whether this command removes unused or detached libraries. Use caution when removing attached libraries. A currently unused library may be needed later for design optimizations. You can remove any library or only remove scaling libraries. Note that, before removing any library, the *remove_library* Tcl command first loads all pending delay load libraries. In case of a loading error for any delay loaded library operation, the *remove_library* Tcl command fails and issues the following error message: `ERROR: No libraries were removed, because of load errors. Check log...` All library load errors must be resolved before any libraries can be removed. This approach ensures that all libraries are accounted for before removing any. Otherwise, a wrong library, for example a library needed for another scenario, may get removed.

SYNTAX
 	
`string` **`remove_lib`** `-all` *`libraries`*
`list` *`libraries`*
	Syntax
	
	`remove_library [`*`object_list`*`] \`
	`  [-all_unused] \`
	`  [-unused_scaling_lib_groups] \`
	`  [-ignore_link_path_reference]`

77

ARGUMENTS

-all
Removes all libraries.

libraries
Provides a list of libraries to remove.

where the arguments have the following meaning:

[*object_list*]	List of libraries to remove from memory provided they are not attached to any project.
[-all_unused]	Remove all libraries from memory that are not attached to any project.
[-unused_scaling_lib_groups]	Only remove all scaling library groups that are currently not used. Typically, you use this argument after you deleted scenarios to remove scaling libraries that were only used in those deleted scenarios.
[-ignore_link_path_reference]	Detach the specified libraries from projects in which they are not used, that is, remove them from the link path of the project.

78

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790565	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
remove_max_area Removes the **max_area** attribute from the current design.	Command: remove_max_area standard SDC command

SYNTAX
 	
int **remove_max_area**
	option:
	  --license list required licenses
	  --help display command help

ARGUMENTS None.	description: This command is the same as standard SDC command.

79

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790566	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056767
remove_max_capacitance Removes maximum capacitance limits from ports or designs.	**remove_max_capacitance** Removes the *max_capacitance* constraint from the specified objects, which can be ports, pins, clocks, or designs. This command is the standard SDC command.
SYNTAX string **remove_max_capacitance** *object_list* list *object_list*	**Syntax** remove_max_capacitance *objects*
ARGUMENTS object_list Provides a list of ports or designs from which to remove maximum capacitance limits.	where the argument has the following meaning: *objects* — List of objects from which to remove a *max_capacitance* constraint. The objects can be ports, pins, clocks, or designs.

80

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790567	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056768
remove_max_fanout Removes maximum fanout limits from ports or designs.	**remove_max_fanout** Removes the *max_fanout* constraint from the specified objects, which can be ports, pins, clocks, or designs. This command is a standard SDC command.
SYNTAX string **remove_max_fanout** *object_list* list *object_list*	**Syntax** remove_max_fanout *objects*
ARGUMENTS object_list Lists the ports or designs from which to remove maximum fanout limits.	where the argument has the following meaning: *objects* — List of objects from which to remove a *max_fanout* constraint. The objects can be ports, pins, clocks, or designs.

81

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790569	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056769
remove_max_transition Removes maximum transition limits from ports, clocks or designs.	**remove_max_transition** Removes the *max_transition* constraint from the specified objects, which can be ports, pins, clocks, or designs. This command is the standard SDC command.
SYNTAX string **remove_max_transition** *object_list* list *object_list*	**Syntax** remove_max_transition *objects*
ARGUMENTS object_list Lists the ports, clocks or designs from which to remove maximum transition limits.	where the argument has the following meaning: *objects* — List of objects from which to remove a *max_transition* constraint. The objects can be ports, pins, clocks, or designs.

82

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790573	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
remove_multi_scenario_design Removes all multi scenario objects from memory and removes from disk all images generated by multi scenario analysis.	Command: remove_multi_scenario_design Don't support.
SYNTAX boolean **remove_multi_scenario_design**	option: --license list required licenses --help display command help
	description: Don't support.

83

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790579-80	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056771

remove_output_delay

Removes output delay from output ports or pins.

SYNTAX

```
string remove_output_delay
[-clock clock_name]
[-clock_fall]
[-level_sensitive]
[-rise]
[-fall]
[-max]
[-min]
port_pin_list

stringclock_name
list  port_pin_list
```

ARGUMENTS

-clock *clock_name*
Relative clock; {""} for input delay relative to no clock.

-clock_fall
Removes the delay relative to falling edge of clock. If you specify *clock_name* without **-clock_fall**, the delay relative to rising edge of the clock is removed.

-level_sensitive
Removes level-sensitive output delay.

-rise
Removes rising output delay.

-fall
Removes falling output delay.

-max
Removes maximum output delay.

-min
Removes minimum output delay.

port_pin_list
Specifies a list of ports and pins. Each element in the list is either a collection of ports or pins, or a pattern which matches ports or pins on the current design.

remove_output_delay

Removes output delay on ports or pins that was set using the *set_output_delay* Tcl command. Currently, this command removes all output delay information from the specified pins or ports.

Syntax

```
remove_output_delay port_pin_list \
   [-rise] \
   [-fall] \
   [-min] \
   [-max] \
   [-clock_fall] \
   [-clock clocks] \
   [-level_sensitive]
```

where the arguments have the following meaning:

port_pin_list	Names of the ports and pins on which to remove the output delays.
[-rise]	Not supported yet.
[-fall]	Not supported yet.
[-min]	Not supported yet.
[-max]	Not supported yet.
[-clock_fall]	Not supported yet.
[-clock *clocks*]	Not supported yet.
[-level_sensitive]	Not supported yet.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 005677584	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0055846
remove_propagated_clock Removes a propagated clock specification.	**remove_propagated_clock** Removes from objects the propagated clock attribute that was set using the *set_propagated_clock* Tcl command. The objects can be a combination of clocks, pins, and ports.
SYNTAX string **remove_propagated_clock** *object_list* list *object_list*	**Syntax** remove_propagated_clock *object_list* where *object_list* is a collection of clocks, pins, or ports.
ARGUMENTS object_list Lists clocks, ports, or pins.	

85

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790587	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056777
remove_rail_voltage Removes power rail voltage that was set by the **set_rail_voltage** command on cells.	**remove_rail_voltage** Removes all rail voltage settings for all or selected scenarios. When you disable automatic updating of the supply voltages by the power network analyzer, that is, you set *ta* parameter *enable_pna_rail_voltage* to *false*, and you manually set supply voltages using the *set_rail_voltage* Tcl command, then these settings are saved in the design database and can only be modified by another *set_rail_voltage* Tcl command or removed by the *remove_rail_voltage* Tcl command.
SYNTAX int **remove_rail_voltage** *cell_list* list *cell_list*	**Syntax** remove_rail_voltage \ -all \ [-scenario *scenario_name*]
ARGUMENTS cell_list Specifies a list of cells from which to remove rail voltages.	where the arguments have the following meaning: -all — Remove rail voltage overrides on all cells. For now, this argument is mandatory. In the future, you will have the option to specify a list of cells. [-scenario *scenario_name*] — The scenario for which to remove the rail voltage overrides. This argument is required for a MCMM design.

86

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790589	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056781
remove_scenario Removes a scenario in multi scenario analysis.	**remove_scenario** Removes scenario(s) that were created using the *create_scenario* Tcl command. A scenario is a set of external and process conditions under which the design needs to be analyzed.
SYNTAX **remove_scenario** *scenario list*	**Syntax** remove_scenario { *scenario_name* [*scenario_name*]... }
ARGUMENTS scenario list A list of unique strings used to identify each scenario.	where *scenario_name* is the name of the scenario you want to remove.

87

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790598-99	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

remove_si_noise_analysis

Removes the effect of the **set_si_noise_analysis** command.

SYNTAX

```
int remove_si_noise_analysis
[-ignore_arrival inets]
[-victims vnets]
[-aggressors anets]
[-above]
[-below]
[-low]
[-high]
[-all]

list    inets
list  vnets
list  anets
```

ARGUMENTS

-ignore_arrival *inets*
Removes the effect of using the **set_si_noise_analysis** command with its **-ignore_arrival** option. You cannot use this option with the **-victims** or **-aggressors** option.

-victims *vnets*
Removes the effect of using the **set_si_noise_analysis** command with its **-victims** option. When this option **-victims** is used to remove the effect set by the command **set_si_noise_analysis** with the option **-aggressors**, it does not remove any exclusion on the nets. However, when you use the **-victims** option with **-aggressors** option, the command restores the pair-wise relationship. You cannot use this **-victims** option with the **-ignore_arrival** option.

-aggressors *anets*
Removes the effect of the **set_si_noise_analysis** command with its **-aggressors** option. When this option **-aggressors** is used to remove the effect set by the command **set_si_noise_analysis** with the option **-victims**, it does not remove any exclusion on the nets. However, when you use the **-aggressors** option with the **-victims** option, the command restores the pair-wise relationship. You cannot use this **-aggressors** option with the **-ignore_arrival** option.

```
Command: remove_si_noise_analysis
  remove the effect of set_si_noise_analysis command

option:
  -above                        above
  -below                        below
  -high                         high
  -low                          low
  -all                          remove all effect
  -victim collection            victime nets
  -aggressor collection         aggressor nets
  -ignore_arrival collection    ignore arrival window
  --get_option arg<1>           get option value
  --set_option ...              set option value
  --get_default arg<1>          get default value
  --set_default ...             set default value
  --system_default              use system default values
  --list_options                list current option values
  --load_options ...            load current option values
  --license                     list required licenses
  --help                        display command help

description:
  -all            : remove all effect of
                    set_si_noise_analysis
  -victim         : remove -victim effect specified by
                    set_si_noise_analysis
  -aggressor      : remove -aggressor effect specified by
                    set_si_noise_analysis
  -ignore_arrival : remove -ignore_arrival effect specified
                    by set_si_noise_analysis
```


88

-above
Removes the effect of **set_si_noise_analysis -exclude -above** command.

-below
Removes the effect of **set_si_noise_analysis -exclude -below** command.

-low
Removes the effect of **set_si_noise_analysis -exclude -low** command.

-high
Removes the effect of **set_si_noise_analysis -exclude -high** command.

-all
Removes all the effects of **set_si_noise_analysis** command on all the nets.

89

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790603	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056785
remove_user_attribute Removes a user attribute from an object.	**remove_user_attribute** Removes one or more user-defined attributes that were previously assigned to the object using the *set_user_attribute* Tcl command.
SYNTAX string **remove_user_attribute** [-quiet] [-class *class_name*] *object_spec attr_name* string*class_name* list *object_spec* string*attr_name*	**Syntax** remove_user_attribute *objects attribute_name* \ [-class *class_name*] \ [-quiet]
ARGUMENTS -quiet Does not report any messages. -class *class_name* If *object_spec* is a name, this is its class. Allowable values are design, port, cell, pin, net, lib, lib_cell, or lib_pin. object_spec Shows objects from which to remove the attribute. Each element in the list is either a collection or a pattern which combines with the *class_name* to find the objects. attr_name Provides the name of the attribute.	where the arguments have the following meaning: *objects* — Objects from which to remove a user-defined attribute. *attribute_name* — Name of attribute to remove. [-class *class_name*] — Only remove the attribute on an object if that object is of the specified class. For a list of all classes, see Aprisa Classes. [-quiet] — Not supported yet.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790637	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056795
report_aocvm Displays information about AOCVM components and AOCVM coefficients in the current design.	**report_aocvm** Reports on the derating factors generated from the advanced on-chip variation (OCV) analysis.
SYNTAX int **report_aocvm** *object_list* list *object_list*	**Syntax** report_aocvm \ [-from *pins-and-ports*] \ [-to *pins-and-ports*]
ARGUMENTS object_list Specifies a path for which AOCVM derate components and metrics are to be reported.	where the arguments have the following meaning: [-from *pins-and-ports*] — Select all timing paths starting at one of the pins or ports from the specified collection. [-to *pins-and-ports*] — Select all timing paths ending at one of the pins or ports from the specified collection.

91

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790640	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056798

report_attribute

Reports the attributes on one or more objects.

SYNTAX

```
string report_attribute [-class class_name] [-nosplit] [-application] object_spec
string  class_name
list    object_spec
```

ARGUMENTS

-class *class_name*
: If object_spec is a name, this is its class. Allowable values are *design*, *port*, *cell*, *pin*, *net*, *lib*, *lib_cell*, or *lib_pin*.

-nosplit
: Does not split lines if column overflows.

-application
: Lists application attributes as well as user-defined attributes.

object_spec
: List of objects to report. Each element in the list is either a collection or a pattern which combines with the *class_name* to find the objects.

report_attribute

Reports attributes and their values on a specified set of objects.

The objects in the provided set may be from different classes. You can further narrow your selection in the set by specifying the class of objects on which you want to report. The report includes both user defined attributes and Aprisa built-in attributes such as wire length and capacitance of a net.

Syntax

```
report_attribute objects \
   [-class class_name] \
   [-application] \
   [-pattern string] \
   [-sort]
```

where the arguments have the following meaning:

objects	Objects for which to report its attributes.
[-class *class_name*]	Class name of the objects for which you want the attributes.
[-application]	Report the application attributes as well as the user-defined attributes.
[-pattern *string*]	Report on the attributes that match the specified name pattern.
[-sort]	Sort the names of the reported attributes.

92

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790647	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
report_case_analysis Reports case analysis entries on ports and pins.	Command: report_case_analysis report case analysis info and affected flip flop
SYNTAX string **report_case_analysis** [-all] [-nosplit]	option: -all — report all case analysis and constant propagation -seq — report all impacted sequential pin -nosplit — no line split -from collection — report from a pin or list of pins --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --system_default — use system default values --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
ARGUMENTS -all Reports the pins upon which you have set case analysis values and reports the built-in constant pins of the design that are considered for start points of logic constant propagation. Logic constant propagation is performed by default from the constant pins of the design, unless the **disable_case_analysis** variable is set to true. -nosplit Prevents line splitting and facilitates writing software to extract information from the report output. If you do not use this option, most of the design information is listed in fixed-width columns. If the information for a given field exceeds the column width, the next field begins on a new line starting in the correct column.	

93

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790653-54	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056807

report_clock

Reports clock-related information.

SYNTAX

```
string report_cloc
[-attributes]
[-skew]
[-groups]
[-nosplit]
[clock_names]

list  clock_names
```

ARGUMENTS

-attributes
Shows clock attributes and provides a list of all the clocks in the current design. The information for each clock includes source type, signal rise and fall times, and attributes. This report is shown by default.

-skew
Reports clock latency (source and network latency) and uncertainty information set on the design by the **set_clock_latency** and **set_clock_uncertainty** commands, respectively.
Clock network latency information includes rise latency and fall latency. Clock source latency information includes rise latency and fall latency for early and late arrivals. Clock uncertainty information includes intraclock setup or hold uncertainty and interclock setup or hold uncertainty. This option also reports any fixed clock transition set by using the **set_clock_transition** command. This option only reports active clocks.

-groups
Shows the current setting of clock groups, including the list of active clocks in the current analysis scope and grouping of exclusive clocks and asynchronous clocks set by using the **set_clock_groups** command.

-nosplit
Specifies not to split lines if a column overflows. Most of the design information is listed in fixed-width columns. If the information for a given field exceeds the column width, the next field begins on a new line, starting in the correct column. This option prevents line-splitting and facilitates writing software to extract information from the report output.

clock_names
Lists the clocks that must be reported. Substitute the list you want for *clock_names*.

report_clock

Reports clock information, such as the clock period, its waveform, clock latency and uncertainty, on all, or the specified set of clocks.

Syntax

```
report_clock [clock_list] \
   [-attributes] \
   [-skew] \
   [-nosplit] \
   [-significant_digits number]
```

where the arguments have the following meaning:

clock_list	Clocks on which to report. By default, all clocks are included.
[-attributes]	Include the values of the clock attributes in the report. This argument is only applicable when the *-skew* argument is used.
[-skew]	Include the clock latency and uncertainty in the report.
[-nosplit]	Do not split lines if the rows do not fit on a letter-sized page. This setting results in a less readable table, but is easier to process by other tools and scripts.
[-significant_digits *number*]	Set the precision by providing the number of digits to report after the decimal. The default value is *2*.

94

report_constraint

Displays constraint-related information about a design.

SYNTAX

```
int report_constraint [-all_violators] [-verbose]
          [-path_type format] [-max_delay] [-min_delay]
          [-max_capacitance] [-min_capacitance]
          [-max_transition] [-min_transition]
          [-max_fanout] [-min_fanout]
          [-min_pulse_width] [-min_period]
          [-recovery] [-removal] [-max_skew]
          [-clock_gating_setup] [-clock_gating_hold]
          [-clock_separation]
          [-connection_class]
          [-ignore_register_feedback feedback_slack_cutoff]
          [-significant_digits digits] [-nosplit]


string format
int digits
float slack_cutoff
float feedback_slack_cutoff
```


report_constraint

Reports the status of the design with respect to the specified design constraints. This report includes details of the design constraints that are violated and where they are violated. You can specify the types of constraints for which you want a report.

For MCMM designs, by providing a prefix for the output file names, you can create reports for all scenarios at once.

When you enable parametric on-chip variation modeling (pocvm), then instead of reporting expected (average) timing results, you can get a report on worst results for a given confidence interval. You specify the interval as a multiple of sigmas.

Syntax

```
report_constraint [-all_violators] \
   [-pins pin_list] \
   [-verbose] \
   [-reason] \
   [-path_type end | slack_only] \
   [-max_delay] \
   [-min_delay] \
   [-max_capacitance] \
   [-min_capacitance] \
   [-max_transition] \
   [-min_transition] \
   [-max_fanout] \
   [-min_fanout] \
   [-max_fanout_count] \
   [-delay_noise] \
   [-min_pulse_width] \
   [-min_period] \
   [-recovery] \
   [-removal] \
   [-max_skew] \
   [-clock_gating_setup] \
   [-clock_gating_hold] \
   [-clock_separation] \
   [-include_clock_net] \
   [-remove_clock_reconvergence_pessimism value] \
   [-ignore_register_feedback value] \
   [-significant_digits number] \
   [-nosplit] \
   [-html] \
   [-summary] \
   [-noenvironment] \
   [-no_hierarchical_pins] \
   [-no_buffer_inverter_on_clock] \
   [-scenario scenario] \
   [-prefix filename_prefix] \
   [-sigma n]
```

ARGUMENTS

-all_violators
Indicates that a summary is to be displayed showing the worst violation per endpoint of each violated design rule constraint in the current design. The **-verbose** option provides detailed information on each constraint violation. Multiple violations for a given constraint are listed from the greatest to the least violator.

-verbose
Indicates that more detail is to be shown about constraint calculations.

-path_type *format*
Specifies the format for the path report. Allowed values are *slack_only* (the default), and *end*. This option has an effect only if the **-verbose** option is not used. If *slack_only* is specified, the report displays only endpoint slacks. If *end* is specified, the report has a column format that shows one line for each path, with only the endpoint path total, required-time, and slack.

-max_delay
Indicates that only max_delay and setup information is to be displayed. The default constraint report displays all timing and design rule constraints.

-min_delay
Indicates that only min_delay and hold information is to be displayed. The default constraint report displays all timing and design rule constraints.

-max_capacitance
Indicates that only max_capacitance constraint information is to be displayed. **-max_capacitance** is a design rule used to limit total capacitance on a net. The **-max_capacitance** option displays the max_capacitance cost (the sum of all max_capacitance violations). To see details about the worst violator, use the **-verbose** option in addition to the **-max_capacitance** option. To see details about all max_capacitance violations, use the **-all_violators** and **-verbose** options in addition to the **-max_capacitance** option. The default constraint report displays all timing and design rule constraints.

-min_capacitance
Indicates that only min_capacitance constraint information is to be displayed. The **-min_capacitance** option is a design rule used to limit total capacitance on a net. The default constraint report displays all timing and design rule constraints.

-max_transition
Indicates that only max_transition constraint information is to be displayed. **-max_transition** is a design rule used to limit transition time on a ports and pins. The default constraint report displays all timing and design rule constraints. If the library uses the cmos2 delay model, max_edge_rate information is shown instead.

-min_transition
Indicates that only min_transition constraint information is to be displayed. **-min_transition** is a design rule used to set a minimum transition time on a ports and pins. The default constraint report displays all timing and design rule constraints. If the library uses the cmos2 delay model, max_edge_rate information is shown instead.

-max_fanout
Indicates that only max_fanout constraint information is to be displayed. **-max_fanout** is a design rule used to limit fanout_load on a net. The default constraint report displays all timing and design rule constraints.

-min_fanout
Indicates that only min_fanout constraint information is to be displayed. **-min_fanout** is a design rule used to set a minimum fanout_load on a net. The default constraint report displays all timing and design rule constraints.

-min_pulse_width
Indicates that only min_pulse_width constraint information is to be displayed. **-min_pulse_width** is a design rule used to set a minimum pulse width high or low at a clock pin or at pins in the clock network. The default constraint report displays all timing and design rule constraints.

-min_period
Indicates that only min_period constraint information is to be displayed. **-min_period** is a design rule used to set a minimum period on a clock signal. The default constraint report displays all timing and design rule constraints.

where the arguments have the following meaning:

[-all_violators]	Report all violations of the specified rules. By default, only a summary of violations for each rule is provided.
[-pins *pin_list*]	Report only violations on the specified pins.
[-verbose]	Report in detail on the violations.
[-reason]	Report the reason why optimization was not able to fix the violations.
[-max_delay]	Include violations of the maximum allowed delay and setup constraints. By default, these violations are not included.
[-min_delay]	Include violations of the minimum required delay and hold constraints. By default, these violations are not included.
[-max_capacitance]	Include violations of the maximum capacitance constraints.
[-min_capacitance]	Include violations of the minimum capacitance constraints.
[-max_transition]	Include violations of the maximum transition constraints.
[-min_transition]	Include violations of the minimum transition constraints.
[-max_fanout]	Include violations of the maximum fanout load constraints.
[-min_fanout]	Include violations of the minimum fanout load constraints.
[-max_fanout_count]	Include violations of the maximum fanout count constraints.
[-delay_noise]	Include delay noise violations.
[-min_pulse_width]	Include violations of minimum pulse width constraints.
[-min_period]	Not supported yet.
[-recovery]	Not supported yet.
[-removal]	Not supported yet.
[-max_skew]	Not supported yet.
[clock_gating_setup]	Not supported yet.
[clock_gating_hold]	Not supported yet.
clock_separation]	Not supported yet.
[-include_clock_net]	Include violations on clock nets.
[-remove_clock_reconvergence_pessimism *value*]	Only check for common path pessimism when the slack is less than the specified value.

96

between the control pin transition to the inactive state, and the active edge of the synchronous clock signal. This time is from the control signal going inactive to the clock edge that latches data in. The asynchronous control signal must remain constant during this time, or an incorrect value may appear at the outputs. The default constraint report displays all timing and design rule constraints.

-removal
Indicates that only removal constraint information is to be displayed. **-removal** is a timing constraint used to describe the minimum allowable time between the clock pin inactive edge, while the asynchronous pin is active, to the inactive edge of the same asynchronous control pin. The default constraint report displays all timing and design rule constraints.

-max_skew
Indicates that only max_skew constraint information is to be displayed. **-max_skew** is a timing constraint that checks the maximum separation time allowed between two clock signals. The default constraint report displays all timing and design rule constraint.

-clock_gating_setup
Indicates that only clock_gating_setup constraint information is to be displayed. **-clock_gating_setup** is a timing constraint used to set a minimum setup time between a clock and a signal controlling the gating of that clock. The default constraint report displays all timing and design rule constraints.

-clock_gating_hold
Indicates that only clock_gating_hold constraint information is to be displayed. **-clock_gating_hold** is a timing constraint used to set a minimum hold time between a clock and a signal controlling the gating of that clock. The default constraint report displays all timing and design rule constraints.

-clock_separation
Indicates that only clock_separation constraint information is to be displayed. **-clock_separation** is a timing constraint that checks the minimum separation time allowed between two clock signals. The default constraint report displays all timing and design rule constraint.

-connection_class
Indicates to display only connection_class constraint information. The connection_class constraint is displayed only if there is a connection_class violation.

-ignore_register_feedback *feedback_slack_cutoff*
Indicates to ignore any timing path that starts and ends at the same register and holds a value. This option applies to min delay as well as max delay reports. Only paths with slack less than the specified feedback_slack_cutoff are ignored. This option is applied as a filter to the paths after they are generated. Therefore, the number of paths generated may be less than the
Allowed values are 0-13; the default is determined by the **report_default_significant_digits** variable, whose default value is 2. Use this option if you want to override the default.

-nosplit
Most of the design information is listed in fixed-width columns. If the information for a given field exceeds the width of the column, the next field begins on a new line, starting in the correct column. The **-nosplit** option prevents line splitting and facilitates writing software to extract information from the report output.

[-ignore_register_feedback *value*]	Ignore paths starting and ending at the same clocked element if the total path delay exceeds the specified value.
[-significant_digits *number*]	Controls the precision by number of digits to use after the decimal point.
[-nosplit]	Do not split rows over more than one line if they do not fit a letter-sized page.
[-html]	Generate a report in HTML format. By default, an ASCII report is generated.
[-summary]	Include a timing summary.
[-noenvironment]	Do not report values of environment variables.
[-no_hierarchical_pins]	Do not report on hierarchical pins.
[-no_buffer_inverter_on_clock]	Do not report on buffers and inverters in the clock path.
[-scenario *scenario*]	Report timing results and design constraint violations for the specified scenario in a MCMM design.
[-prefix *filename_prefix*]	If no scenario is set, only one report is created and is named *filename_prefix*. If a working scenario is specified with the *set_working_scenario* Tcl command, only one report is created. Its name starts with *filename_prefix*, followed by a dot and the name of that scenario. If you are in MCMM mode and have several scenarios set with the *current_session* Tcl command, a report is generated for each of the scenarios in the session. The names of the reports start with *filename_prefix*, followed by a dot, and the name of that scenario. Note that the *-prefix* argument redirects the report to a file. Hence, it has precedence over the > redirection operator.
[-sigma *n*]	Report the *n*-sigma delay. This argument is only relevant if the *-verbose* argument is specified and the *ta* parameter *timing_pocvm_enable_analysis* is *true*.

97

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790683-84

report_crpr

Reports the clock reconvergence pessimism calculated between specified register clock pins or ports.

SYNTAX

```
int report_crpr -from from_latch_clock_pin
-to to_latch_clock_pin
[-from_clock from_clock]
[-to_clock to_clock]
[-setup | -hold]
[-significant_digits digits]

stringfrom_latch_clock_pin
stringto_latch_clock_pin
string  from_clock
string  to_clock
integerdigits
```

ARGUMENTS

-from *from_latch_clock_pin*
Specifies the clock pin of the launching sequential device or clock gating check to be reported. A constrained input port may also be used.

-to *to_latch_clock_pin*
Specifies the clock pin of the capturing sequential device or clock gating check to be reported. A constrained output port may also be used.

-from_clock *from_clock*
Specifies the clock that fans out to the launching sequential device.

-setup
Indicates that the CRP used for a setup data path is to be reported. The **-setup** and **-hold** options are mutually exclusive. If neither option is specified, **-setup** is assumed.

-hold
Indicates that the CRP used for a hold data path is to be reported. The **-setup** and **-hold** options are mutually exclusive. If neither option is specified, **-setup** is assumed.

-significant_digits *digits*
Specifies the number of digits to the right of the decimal point that are to be reported. Allowed values are 0-13. If this option is not specified the number of significant digits is determined by the **report_default_significant_digits** variable which has a default value of 2.

PI's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056822

report_crpr

Generates a report on the clock reconvergence pessimism (CRP) on a set of user-specified timing paths. You can specify paths by the source pin, sink pain, launching clock, capturing clock or any combination of these.

Clock reconvergence occurs when the clock path to the launching clock and the clock path to the capturing clock share a common subpath. The reconvergence pessimism is the difference of the max timing and min timing of that shared subpath.

For each of the selected paths, the report includes the path, the clock node that is common to both launch and capturing clock, and the CRP value of the path for both setup and hold constraints.

Syntax

```
report_crpr \
   -from [from_pins] \
   -to [to_pins] \
   [-from_clock [from_clocks]] \
   [-to_clock [to_clocks]]
```

where the arguments have the following meaning:

`-from [from_pins]`	Select only paths that start from one of the specified pins.
`-to [to_pins]`	Select only paths that end in one of the specified pins.
`[-from_clock [from_clocks]]`	Select only paths whose launching clock is one of the specified clocks.
`[-to_clock [to_clocks]]`	Select only paths whose capturing clock is one of the specified clocks.

98

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790688-89

report_delay_calculation

Displays the actual calculation of a cell or net timing arc delay value.

SYNTAX

```
int report_delay_calculation [-min | -max]
         [-from_rise_transition value]
         [-from_fall_transition value]
         -from from_pin -to to_pin | -of_objects objects
         [-nosplit]
         [-thresholds]
         [-crosstalk]

string      from_pin
string      to_pin
float       value
collectionobjects
```

ARGUMENTS

-min
: Indicates that minimum delay calculation is to be shown. The design must be in min/max mode.

-max
: Indicates that maximum delay calculation is to be shown. This is the default if neither **-min** nor **-max** is specified.

-from_rise_transition *value*
: Specifies a value to be used by the delay calculation for the from rise transition.

-from_fall_transition *value*
: Specifies a value to be used by the delay calculation for the from fall transition.

-from *from_pin* -to *to_pin*
: Specifies the start and end points of a timing arc within a design. For a cell timing arc, the pins must represent the input and output pins of a common leaf cell, which have a timing arc specified between them in the library. For a net timing arc, the pins must be a driver and a load on a common net. Port names are allowed in place of pin names for net arcs. You must use either the **-from/-to** combination or the **-of_objects** argument, but not both.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056832-34

report_delay_calculation

Reports the results of the delay calculation for a set of timing arcs. You specify the arcs by providing the ports or pins where the timing arcs start, the pins or ports where the arc ends, or both. You can specify the transition time of the signal at the start of the timing arc.

The delay calculation is always reported for one specific scenario, even when analyzing multiple scenarios.Therefore the *-scenario* argument is mandatory for MCMM designs.

NOTE: Currently, not all the arguments have been fully implemented.

Syntax

```
report_delay_calculation \
   [-min] \
   [-max] \
   [-nosplit] \
   [-threshold] \
   [-from_rise_transition rise] \
   [-from_fall_transition fall] \
   [-crosstalk] \
   [-from start_timing_arc] \
   [-to end_timing_arc] \
   [-of_objects timing_arcs] \
   [-crosstalk] \
   [-scenario scenario]
```

where the arguments have the following meaning:

[-min]	Report on the minimum delay calculation. NOTE: Currently, this argument must be used with the *-threshold* argument because the command is not yet fully implemented.
[-max]	Report on the maximum delay calculation. NOTE: Currently, this argument must be used with the *-threshold* argument because the command is not yet fully implemented.
[-nosplit]	Do not split lines in the report if the columns do not fit the width of the page. Using this argument, an ASCII file is created which is harder to read, but easier to process using script languages.
[-thresholds]	Report the voltage threshold levels that are used to calculate rise and fall delays and slews.
[-from_rise_transition *rise*]	Value to use for the rise transition time of the signal at the start of the arc. The default value is *0.0*.

99

-of_objects *objects*
Specifies a collection of timing arcs (created with the **get_timing_arcs** command) on which to report. Arcs in the list are reported in order of from and to pins. You must use either the **-from/-to** combination or the **-of_objects** argument, but not both.

-nosplit
Prevents line-splitting and facilitates writing software to extract information from the report output. Most of the design information is listed in fixed-width columns. If the information in a given field exceeds the column width, the next field begins on a new line, starting in the correct column.

-thresholds
Reports the characterization thresholds that are used for delay calculation.

-crosstalk
Reports the crosstalk information for a net arc. The arc is specified by **-from_pin** and **-to_pin**. It is not permitted with **-of_objects** and user choosen transition time **-from_rise_transition** and **-from_fall_transition**. The crosstalk information is reported from the last **update_timing**.

[-from_fall_transition *fall*]	Value to use for the fall transition time of the signal at the start of the arc. The default value is *0.0*.
[-crosstalk]	Report on the impact of crosstalk on the arc delay. NOTE: Currently, this argument is not yet implemented.
[-from *start_timing_arc*]	Start point of the timing arcs on which to report.
[-to *end_timing_arc*]	Specifies the end point of the timing arcs on which to report.
[-of_objects *timing_arcs*]	Specifies the objects, typically cells, whose arcs on which to report.
[-crosstalk]	Report the impact of crosstalk on the arc delay.
[-scenario *scenario*]	Scenario for which to report the delay calculations. This argument is mandatory for MCMM designs.

100

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790703	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056842
report_disable_timing Reports disabled timing arcs in the current design.	**report_disable_timing** Reports on all disabled timing arcs in the current design. Timing arcs can be disabled because they are logically impossible, that is, they require a signal to be both true and false, or because they are not possible given the current constant signals like in a case analysis, or the user specifies to ignore a path, or it is a path that the timer breaks to resolve a circular dependency (loop breaking path). For every arc, the report lists the reason why the arc is disabled. The reasons can be one of the following: • Case analysis (c): Arc does not apply for the current case analysis. • Conditional arc (C): Conditional arc which condition is not met. • Default conditional arc (d): Arc to be used when no other conditional arcs are active. • Loop breaking (l): Arc was disabled by the timer to break a timing loop. • False net-arc (f); Arc can logically never be active. • User-defined (u); Arc disabled by the user. • Propagation of constant values (p): Arc disabled based on propagated constant values.
SYNTAX string **report_disable_timing** [-nosplit] [*cells_or_ports*] collection *cells_or_ports*	**Syntax** report_disable_timing [-nosplit]
ARGUMENTS -nosplit Prevents line splitting and facilitates writing software to extract information from the report output. If you do not use this option, most of the design information is listed in fixed-width columns. If the information for a given field exceeds the column width, the next field begins on a new line starting in the correct column. cells_or_ports Limits disabled arc reporting to the specified list of cells or ports. Provide the list or collection of cells or ports as an argument to the command.	where **[-nosplit]** prevents line splitting if the rows do not fit on a single page.

101

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790707

report_driver_model

Displays the driver model for a library cell timing arc used to drive annotated parasitics.

SYNTAX

```
int report_driver_model
-lib_cell lib_cell
-from_pin from_pin
-to_pin to_pin
-rise_slew rise_slew
-fall_slew fall_slew
-capacitance capacitance

string lib_cell
string from_pin
string to_pin
float  rise_slew
float  fall_slew
double capacitance
```

ARGUMENTS

-lib_cell *lib_cell*
Specifies the name of the library cell for which the driver model is to be computed. The name should be in the form *library_name/cell_name*.

-from_pin *from_pin*
Specifies the start point of a timing arc through the *lib_cell*.

-to_pin *to_pin*
Specifies the endpoint of a timing arc through the *lib_cell*.

-rise_slew *rise_slew*
Specifies the rise time in library units on the *from_pin*.

-fall_slew *fall_slew*
Specifies the fall time in library units on the *from_pin*.

-capacitance *capacitance*
Specifies the load in library units on the *to_pin*.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056845-46

report_driver_model

Evaluates a timing arc inside a library cell for a specific input slope and output load capacitance and reports the equivalent driver model, that is, a ramp voltage source and a hold resistor. For MCMM designs, you can specify the scenario for which to show the result. The report includes:

- Total delay from input to output pin
- Slope of transition at the output pin
- The incremental arc delay due to load capacitance
- The incremental output slope due to load capacitance
- The hold resistance of the driver model
- The transition time of the ramp voltage of the driver model

Syntax

```
report_driver_model \
   -lib_cell collection \
   -from_pin string \
   -to_pin string \
   [-rise_slew rise_slew] \
   [-fall_slew fall_slew] \
   [-capacitance load_capacitance] \
   [-scenario scenario]
```

where the arguments have the following meaning:

-lib_cell *collection*	Library cell of which you want to evaluate a timing arc
-from_pin *string*	Start point of the timing arc to evaluate.
-to_pin *string*	End point of the timing arc to evaluate.
[-rise_slew *rise_slew*]	Rise slew at pin specified with the *-from_pin* argument. The default value is *0.1* ns.
[-fall_slew *fall_slew*]	Fall slew at pin specified with the *-from_pin* argument. The default value is *0.1* ns.
[-capacitance *load_capacitance*]	Load capacitance at pin specified with the *-to_pin* argument. The default value is *0.1* pF.
[-scenario *scenario*]	Report the delay values for the specified scenario.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790732	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
report_min_pulse_width Displays minimum pulse width check information about specified pins or ports.	Command: report_min_pulse_width [db:port_pin_list] Report minimum pulse width check information in current design.
SYNTAX int **report_min_pulse_width** [-all_violators] [-significant_digits *digits*] [-nosplit] [-path_type *format*] [-input_pins] [*port_pin_list*] list *port_pin_list*	-verbose show more details -path_type path_type(summary) clock path path display type path_type = summary | full | full_clock_expanded -significant_digits integer(3) number of digits after decimal point -nosplit prevents line splitting if column overflows --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --system_default use system default values --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -all_violators Indicates that only violating minimum pulse width checks are to be reported. -significant_digits *digits* Specifies the number of reported digits to the right of the decimal point. Allowed values are 0-13; the default is determined by the **report_default_significant_digits** variable, whose default value is 2. Use this option if you want to override the default. -nosplit Most of the design information is listed in fixed-width columns. If the information for a given field exceeds the width of the column, the next field begins on a new line, starting in the correct column. **-nosplit** prevents line splitting and facilitates writing software to extract information from the report output.	description: Report minimum pulse width check information in current design.

103

-path_type *format*
Specifies the format of the path report and how the clock path is displayed. Allowed values are: *summary* (the default), which generates a report with a column format that shows one line for each path and shows only the required pulse width, actual pulse width and slack; *short*, which displays only start and end points in the clock path; *full_clock*, which displays full clock paths; and *full_clock_expanded*, which displays full clock paths including all master clocks of a generated clock.

-input_pins
Indicates that input pins are to be shown in the path report. The default isto show only output pins.

port_pin_list
Specifies a list of pins or ports to report. By default, the report contains all pins and ports in the current design.

104

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790746-78

report_noise

Reports noise analysis information.

SYNTAX

```
int report_noise
[-above]
[-below]
[-low]
[-high]
[-nworst_pins pin_count]
[-significant_digits digits]
[-slack_type slack_type]
[-slack_lesser_than slack_limit]
[-all_violators]
[-data_pins]
[-clock_pins]
[-async_pins]
[-verbose]
[-nosplit]
[object_list]

list  object_list
```

ARGUMENTS

-above

Performs the reporting only above the rails. If this option is combined with **-low**, it reports for the noise bumps above the low rail. If it is combined with **-high**, it reports the noise bumps above the high rail. Otherwise, it reports the noise bumps above the high rail and above the low rail.

-below

Performs the reporting only below the rails. If this option is combined with **-low**, it reports for the noise bumps below the low rail. If it is combined with **-high**, it reports the noise bumps below the high rail. Otherwise, it reports the noise bumps below the high rail and below the low rail.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056874-75

report_noise

Reports on the functional noise analysis. The report contains information on the size and width of noise bumps on victim nets, caused by crosstalk, and it reports on the noise slack, that is, the difference between the calculated noise bump and a bump that would cause a functional failure. The latter is derived from the noise sensitivity of the input pin driven by the victim nets.

The noise analysis considers four cases. The victim net can be either low or high and the noise bump can be positive or negative.

- A regular glitch analysis is performed on all pins and the noise resholds for all these nets are set by the *ta* parameters *si_noise_margin_above_high* and *si_noise_margin_below_high*. You can now override these values for selected pins using the *set_noise_margin* Tcl command.
- An overshoot and undershoot analysis is only performed on selected pins for which a noise margin was specified using *set_noise_margin* Tcl command with the *-below_low* or *-above_high* arguments.

Syntax

```
report_noise
   [-threshold value]] \
   [-threshold_low value]\
   [-threshold_high value]\
   [-nets nets] \
   [-victim_only] \
```

where the arguments have the following meaning:

[-threshold *value*]	Do not report on any positive noise peak smaller than the specified value when the victim signal is low or any negative noise peak smaller than the specified value when the victim signal is high. The default value of this threshold is *0.0* mV.
[-threshold_low *value*]	Do not report on any positive noise peak smaller than the specified value when the victim signal is low The default value of this threshold is *0.0* mV.

-low
Performs the reporting only for the low rail. If this option is combined with **-above**, it reports the noise bumps above the low rail. If it is combined with **-below**, it reports the noise bumps below the low rail. Otherwise, it reports the noise bumps for both below and above the low rail.

-high
Performs the reporting only for the high rail. If this option is combined with **-above**, it reports the noise bumps above the high rail. If it is combined with **-below**, it reports the noise bumps below the high rail. Otherwise, it reports the noise bumps for both below and above the high rail.

-nworst_pins *pin_count*
Specifies the number of load pins to be reported. Any number greater than 1 is accepted; the default value is 1.

-significant_digits *digits*
Specifies the number of digits after the decimal point to be displayed for time values in the generated report. Allowed values are 0-13; the default is determined by the **report_default_significant_digits** variable, whose default value is 2. Use this option if you want to override the default. This option controls only the number of digits displayed, not the precision used internally for analysis. For analysis, PrimeTime uses the full precision of the platform's fixed-precision, floating-point arithmetic capability.

-slack_type *slack_type*
Specifies the type of slack to be used. Valid values are area, height, and area_percent. A *slack_type* of area reports slack as the voltage margin multiplied by the noise bump width. The voltage margin is defined by the noise bump height and noise immunity curves or DC noise margin. This setting is the default. A *slack_type* of height reports noise slack as the voltage margin. A *slack_type* of area_percent reports noise slack as the percentage of the noise constraint area. The noise constraint area is computed by multiplying the noise height constraint by the noise bump width.

-slack_lesser_than *slack_limit*
Indicates that only those pins with a slack less (more negative) than *slack_limit* are to be shown.

-all_violators
Indicates that only violating pins (negative slack) are to be shown. This option cannot be used with the **-slack_lesser_than** option. If this option is used with the **-nworst_pins** option, the number of violating pins will be limited by that value.

-data_pins
Indicates that the reporting is done only on pins that are data pins of sequential cells. The effect is similar to preselect the data pins using all_registers -data_pins and pass the resulting collection to the report_noise command.

-clock_pins
Indicates that the reporting is done only on pins that are clock pins of sequential cells. The effect is similar to preselect the clock pins using all_registers -clock_pins and pass the resulting collection to the report_noise command.

-async_pins
Indicates that the reporting is done only on the asynchronous pins of sequential cells. The effect is similar to preselect the asynchronous pins using all_registers -async_pins and pass the resulting collection to the report_noise command.

[-threshold_high *value*]	Do not report on any negative noise peak smaller than the specified value when the victim signal is high.The default value of this threshold is *0.0* mV.
[-nets *nets*]	Only report noise information for the specified nets.
[-victim_only]	Report only victim peak value.

106

-verbose
Shows more details about the calculation of total noise on each load pin, including the individual contribution of each aggressor as well as noise bumps propagated from previous stages of the design.

-nosplit
If the information in a given field exceeds the column width, the next field begins on a new line, starting in the correct column. The **-nosplit** option prevents line-splitting and facilitates writing software to extract information from the report output.

object_list
Specifies the load pins for which the noise reporting is performed. If no pin is specified, reporting is performed on the entire design.

107

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790759

report_port

Displays port information within the design.

SYNTAX

```
string report_port [-verbose]
[-design_rule]
[-drive]
[-input_delay]
[-output_delay]
[-wire_load]
[-nosplit]
[port_names]

list  port_names
```

ARGUMENTS

-verbose
: Indicates that the port report includes all port information. By default, only a summary section is displayed that lists all ports and their direction.

-design_rule
: Reports only port design rule information, including maxCap, manLoad, and maxFanout.

-drive
: Reports only drive resistance, input transition time, and driving cell information for only input and inout ports.

-input_delay
: Reports only the port input delay information you set.

-output_delay
: Reports only the port output delay information you set.

-wire_load
: Reports only the port wire load information.

-nosplit
: Prevents line splitting if column overflows. Most design information is listed in fixed-width columns. If the information for a given field exceeds the column width, the next field begins on a new line, starting in the correct column. This option prevents line-splitting and facilitates writing software to extract information from the report output.

port_names
: Displays information on these ports in the current design. Each element in this list is either a collection of ports or a pattern matching the port names.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056910

report_port

Reports electrical information of boundary ports such as maximum capacitance, minimum capacitance, and reports timing information such as maximum transition time and load information for these ports.

Syntax

```
report_port [port_list] \
   [-design_rule] \
   [-nosplit]
```

where the arguments have the following meaning:

[port_list]	List of ports for which to report on. By default, a report is generated for all ports.
[-design_rule]	Report maximum capacitance, maximum load, and maximum fanout.
[-nosplit]	Do not split rows if they do not fit the width of a page.

108

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790762-63	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056913-14
report_power Generate power reports.	**report_power** Reports the breakdown of total power consumption. It reports static and dynamic power consumption on the entire design if no objects are specified. The report includes a break down of total power consumption by type of objects, such as standard cells, memory cells, flip-flops, latches, gates, and clock cells. You can also request a power consumption report for a specified list of cells. The power analysis is based on activity factors that are typically provided through a VCD or FSDB file that you load using the *extract_switching_activity* Tcl command. If, for a net no switching activity is provided, Aprisa assumes a default activity of *0.1*. You can change that value using the *ta* parameter *power_default_toggle_rate*. Note that the default toggle rate of all clock nets is fixed to *2.0* and cannot be changed. Aprisa also provides four *ta* params to control the active ratio of clock-gating cells: *power_cg_default_active_ratio_1st*, *power_cg_default_active_ratio_2nd*, *power_cg_default_active_ratio_3rd*, and *power_cg_default_active_ratio_extra*.
SYNTAX int **report_power** [-net_power] [-cell_power] [-leaf] [-include_boundary_nets] [-include_estimated_clock_network] [-sort_by *sort_method*] [-nworst *number*] [-power_greater_than *threshold*] [-hierarchy] [-levels *level*] [-clocks *clock_list*] [-groups *group_list*] [*object_list*] [-verbose] [-nosplit] int *number* int *level* string *sort_method* float *threshold* list *clock_list* list *group_list* list *object_list*	**Syntax** report_power [*cells*] \ [-sort_by name | internal_power | switching_power | \ leakage_power | dc_power] \ [-scenario *scenario*] \ [-verbose]

109

ARGUMENTS

-net_power
Indicates to generate net-based power report.

-cell_power
Indicates to generate cell-based power report.

-leaf
Indicates that the power report should traverse the hierarchy and report nets or cells at all lower-levels (as if the design's hierarchy were flat). The default is to report objects at only the current level of hierarchy.

-include_boundary_nets
Indicates that the switching power of primary input nets is to be counted when generating the power report; the default is to exclude boundary input nets.

-include_estimated_clock_network
Indicates that the clock network power estimated by estimate_clock_network_power is to be included in the power report.

-sort_by *sort_method*
Specifies the sorting mode for the net or cell order in the power report.

-nworst *number*
Indicates that the report is to be filtered so that it displays only the top-most *number* of nets or cells sorted by a certain type of sort_method.

-power_greater_than *threshold*
Indicates that only nets or cells with total power value equal to or greater than *threshold* are to be reported.

-hierarchy
Indicates to generate hierarchy-based power report.

-levels *level*
Specifies the number of levels of hierarchies to be displayed in the hierarchy-based power report.

-clocks *clock_list*
Select only nets and/or cells that belong to the clock domains specified by clock_list and generate report for the selected objects.

-groups *group_list*
Select only nets and/or cells that belong to the power groups specified by group_list and generate report for the selected objects.

object_list
Specifies a list of cells and/or nets to be displayed in the cell-based and/or net-based power report.

-verbose
Indicates to report some verbose information, mainly in the header of the report, such as operating conditions, wire load models used to calculate power, and power unit information, etc.

-nosplit
Indicates to prevent line-splitting or section-breaking. By default, if the information for a given field exceeds its fixed column width, the next field begins on a new line, starting in the correct column (line-splitting). Also by default, if the information for one line exceeds the 80 charactor limit, the report is broken into 2 sections, each containing part of the information (section-breaking).

where the arguments have the following meaning:

[*cells*]	List of cells on which to report power. If you do not specify any cells, you receive a total power consumption report on the design.
[-sort_by name | internal_power | switching_power | leakage_power | dc_power]	Order in which the power information needs to be reported. The default value is *internal_power*.
[-scenario *scenario*]	Name of the scenario. This argument is required for MCMM designs.
[-verbose]	Provide a detailed breakdown of power consumption by cells. This argument is only applicable if you specified a list of cells on which to report.

110

report_switching_activity

Reports statistics on the switching activity and signal probability annotation on the current design or instance.

SYNTAX

```
int report_switching_activity
        [-list_not_annotated]
        [-cells cell_list]
        [-average_activity]
        [-base_clock clk]
        [-hierarchy]
        [-coverage]
        [-sort_by [hier | toggle]]
        [-toggle_limit limit]
        [-list_low_activity]
        [-list_by_source source]
        [-list_not_annotated]
        [-exclude exclusion_group]
        [-include_only inclusion_group]


int report_switching_activity -old
        [-rtl | -gate]
        [-list_not_annotated]
        [-cells cell_list]


list cell_list
string clk
int limit
string source
string exclusion_group
string inclusion_group
```


report_switching_activity

Reports signal switching activities (SA) of source pins specified through a collection of input objects. These input objects can include pins, ports, nets, cells or both.

For each pin, it reports the source of its activity factor, which is one of the following:

- *unformat*—Activity set to global default value as no other activity information is available.
- *VCD*—Activity was read from a VCD file.
- *FSDB*—Activity was read from a FSDB file.
- *SAIF*—Activity read from an SAIF file.
- *derived*—Activity was calculated through propagation from other pins.
- *clock*—Activity set to 2.
- default—Activity is set to the value of the *ta* parameter *power_default_toggle_rate because* its activity is not found in a VCD,FSDB or SAIF file, and cannot be determined through propagation

The *report_switching_activity* Tcl command returns a summary report if no arguments are specified.

Syntax

```
report_switching_activity objects \
   [-hierarchical] \
   [-sort_by name | switching_activity] \
   -scenario scenario_name \
   [-verbose]
```


111

ARGUMENTS

-rtl|-gate
These options are only supported when the **-old** option is used.
Indicates whether switching activity annotations are to be reported for objects annotated by an RTL backward SAIF file or a gate-level backward SAIF file. An RTL backward SAIF file is generated using RTL simulation, and contains the switching activity of synthesis invariant objects. These are objects that are not expected to change during synthesis, and include the design ports, and the outputs of sequential and tri-state cells. Calling the **report_switching_activity** command with **-rtl** option will report the switching activity annotation on design ports, sequential cell outputs and tri-state outputs. Use the **-rtl** option after reading an RTL backward SAIF. A gate-level backward SAIF file is generated using gate-level simulation, and contains the switching activity of all design nets. Calling the **report_switching_activity** command with **-gate** option will report the switching activity annotation on the design nets and leaf cell internal power arcs and leakage states. Use the **-gate** option after reading a gate-level backward SAIF file. When neither **-rtl** or **-gate** options are used, the default **-gate** is assumed. If both the options are given, the default **-gate** is assumed.

-list_not_annotated
When the command is used with this flag, the report lists the design nets that have no user switching activity annotation. This report does not include constant value nets (logic one/zero nets). Note that such nets are annotated with default switching activity if they are not user annotated.

-cells *cells_list*
Indicates that switching activity annotation is to be reported only for the specified cells_list.

-average_activity
Produces a report which averages toggle rates over the nets in the design. When combined with the *-hierarchy* flag, average switching activity is computed for each sub block in the design.
It is possible using the filter options *-exclude* or *-include_only* to get a report of average toggle rates over a subset of the nets in the design. This can be useful, for instance, to get the average toggle rate for annotated nets, or even the average toggle rate for annotated nets driven by sequential cells.
This option is not available with the *-old* option.

-base_clock *clk*
When the *-average_activity* report is requested, average activities over nets are reported. The averaged toggle rates reported are by default with respect to a fixed time unit. When the *-base_clock* option is used, the reported averaged toggle rates are with respect to the period of the clock *clk*.
This option is not available with the *-old* option.

-hierarchy
When the command is used with this flag, the report generated will include information about sub blocks in the design. The flag cannot be used with the list generating options for the command.
This option is not available with the *-old* option.

-coverage
Causes the command to produce a summary report about the nets and the design that have switching activity information, but have few toggles. Coverage is defined as the percentage of nets with more than *limit* toggles. See the option *toggle_limit*. This report can be used to verify that switching activities from simulation or propagation are properly exercising the design. Combined with the *-hierarchy* flag, the report can be used to make sure that each block in the design is properly exercised.
This option is not available with the *-old* option.

where the arguments have the following meaning:

objects	Objects such as ports, nets, pins or cells through which source pins are identified.
[-hierarchical]	Get objects at any level of the hierarchy whose local name matches.
[-sort_by name | switching_activity]	Sort the report by object name or switching activity. The default value is *switching_activity*.
-scenario *scenario_name*	Name of scenario for which to report the switching activity. This argument is required for multi-scenario designs.
[-verbose]	Include the toggle rate, clock period, and voltage in the report.

-sort_by *[hier | net_toggle_rate | name]*
When used with the any of the hierarchical reports (that is, using the *-hierarchy* flag with the default report or with *-average_activity* or *-coverage*), the hierarchical blocks in the report are sorted either by hierarchy (depth first), or by averaged toggle rate, or by name.
When used with any of the list reports (that is, using *-list_low_activity* or *-list_by_source* or *-list_not_annotated*), then the list will be sorted by toggle rate, or by name. For the list reports, the option *-sort_by hier* is not accepted.
This option is not available with the *-old* option.

-toggle_limit *limit*
Sets the lower limit for what is considered to be few toggles. This limit is used by the *-coverage* report and by the *-list_low_activity* list report. The default limit is 1 toggle.
This option is not available with the *-old* option.

-list_low_activity
Reports a list of nets in the design with few toggles, where few is defined using the *-toggle_limit* option. The list report can be used to help debug which nets are not being exercised by the simulation testbench.
This option is not available with the *-old* option.

-list_by_source *source*
Reports a list of nets that have switching activity information from a given source.
Possible sources are any of the following:
[annotated, file, propagated, default, set_switching_activity, set_case_analysis, create_clock, no_switching_activity]
Here *annotated* means any annotated (not propagated) source. The *file* source means any source with from an activity file (vcd or SAIF file).
This option is not available with the *-old* option.

-list_not_annotated
Reports a list of nets that have no annotated switching activity. Used for debugging switching activity annotation problems.
This option is not available with the *-old* option.

-exclude *exclusion_group*
This filter option filters out nets from consideration before generating any of the reports the command can generate. Filtering occurs before toggle rate averaging, coverage percentage computations, and list generation. Possible exclusion groups are any of the following: *[sequential, combinational, primary_inputs, black_boxes, three_state, low_activity]*.
A net is defined as sequential if it is driven by a sequential cell. Likewise with combinational, primary_inputs, etc.
Other possible exclusion groups are defined by the source of the activity information:
[annotated, vcd, saif, propagated, default, set_switching_activity, set_case_analysis, create_clock, no_switching_activity]
Here *annotated* means any annotated (not propagated) source. The *file* source means any source from an activity file (usually vcd or SAIF file).
To exclude multiple groups, the argument *exclusion_group* can be a list with multiple groups in the list.
This option is not available with the *-old* option.

-include_only *inclusion_group*
This filter option filters out nets from consideration before generating any of the reports the command can generate. Filtering occurs before toggle rate averaging, coverage percentage computations, and list generation. Only nets that are members of the inclusion group are considered in the report generation.

113

Possible inclusion groups are any of the following: *[sequential, combinational, primary_inputs, black_boxes, three_state, low_activity]*.
A net is defined as sequential if it is driven by a sequential cell. Likewise with combinational, primary_inputs, etc.
Other possible exclusion groups are defined by the source of the activity information:
[annotated, vcd, saif, propagated, default, set_switching_activity, set_case_analysis, create_clock, no_switching_activity]
Here *annotated* means any annotated (not propagated) source. The *file* source means any source from an activity file (usually vcd or SAIF file).
To include multiple groups, the argument *inclusion_group* can be a list with multiple groups in the list. In this case, only nets that are in one of the listed groups will be included in the report.
This option is not available with the *-old* option.

114

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790822	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056965-73

report_timing

Reports timing paths.

report_timing

Reports timing information on a selected set of paths. You can select timing paths by providing a start or end pin, providing a port or a pin along the timing path, selecting the worst timing paths, or providing the path type.

If the *ta* parameter *si_enable_analysis* is enabled, this command also reports crosstalk delta delay information.

If the *ta* parameter *timing_aocvm_enable_analysis* is enabled, the analysis takes into account advanced on-chip variation modeling.

If the *ta* parameter *timing_pocvm_enable_analysis* is enabled, the analysis takes into account parametric on-chip variation modeling.

Delta delay is only reported on input pins. Add the *-input_pins* argument to see the delta delay values.

For MCMM designs, by providing a prefix for the output file names, you can create reports for all scenarios at once.

For designs on which the PASS Timing ECO flow was run, nets that were altered by the PASS Timing ECO flow are marked with a double quote prefix.

SYNTAX

```
string report_timing
[-from from_list
   | -rise_from rise_from_list
   | -fall_from fall_from_list]
[-to to_list
   | -rise_to rise_to_list
   | -fall_to fall_to_list]
[-exclude exclude_list
   | -rise_exclude rise_exclude_list
   | -fall_exclude fall_exclude_list]
[-through through_list]
[-rise_through rise_through_list]
[-fall_through fall_through_list]
[-delay_type delay_type]
[-nworst paths_per_endpoint]
[-max_paths count]
[-path_type format]
[-true]
[-true_threshold path_delay]
[-justify]
[-false]
[-input_pins]
[-unique_pins]
[-start_end_pair]
[-nets]
[-slack_greater_than slack_limit]
[-slack_lesser_than slack_limit]
[-ignore_register_feedback feedback_slack_cutoff]
[-report_ignored_register_feedback]
```

Syntax

```
report_timing [-from pins-and-ports] \
   [-rise_from pins-and-ports] \
   [-fall_from pins-and-ports] \
   [-to pins-and-ports] \
   [-rise_to pins-and-ports] \
   [-fall_to pins-and-ports] \
   [-through pins-and-ports] \
   [-rise_through pins-and-ports] \
   [-fall_through pins-and-ports] \
   [-delay_type max | min | min_max | max_rise | \
      max_fall | min_rise | min_fall] \
   [-nworst number] \
   [-max_paths number] \
   [-path_type short | full | full_clock | \
      full_clock_expanded | slack_only | end | summary] \
   [-reason] \
   [-true] \
   [-true_threshold threshold] \
   [-justify] \
   [-false] \
   [-input_pins] \
   [-unique_pins] \
   [-start_end_pair] \
   [-arrival_time_count count] \
   [-nets] \
   [-slack_greater_than threshold] \
   [-slack_lesser_than threshold] \
   [-ignore_register_feedback value] \
```


115

```
[-group group_name]
[-significant_digits digits]
[-nosplit]
[-transition_time]
[-capacitance]
[-crosstalk_delta]
[-trace_latch_borrow]
[-derate]
[-dont_merge_duplicates]
[-pre_commands pre_command_string]
[-post_commands post_command_string]
[-exceptions]
[-aocvm]
[-recalculate]
[collection1]

list    from_list
list    rise_from_list
list    fall_from_list
list    to_list
list    rise_to_list
list    fall_to_list
list    exclude_list
list    rise_exclude_list
list    fall_exclude_list
list    through_list
list    rise_through_list
list    fall_through_list
stringdelay_type
int   paths_per_endpoint
int   paths_per_startpoint
int   count
stringformat
float path_delay
list  group_name
int   digits
string  pre_command_string
string  post_command_string
collection collection1
```

```
[-report_ignored_register_feedback] \
[-group { name [name]... }] \
[-significant_digits digits] \
[-nosplit] \
[-transition_time] \
[-coordinate] \
[-capacitance] \
[-crosstalk_delta] \
[-trace_latch_borrow] \
[-derate] \
[-html] \
[-noenvironment] \
[-scenario scenario] \
[-internal_path] \
[-no_hierarchical_pins] \
[-no_buffer_inverter_on_clock] \
[-summary] \
[-brief_summary] \
[-histogram] \
[-step_of_histogram] \
[-unconstrained_path] \
[-prefix filename_prefix] \
[-aocvm] \
[-voltage] \
[-variation] \
[-sort_by_slack]
```

ARGUMENTS

-from *from_list*
Specifies that only paths from the named pins, ports, nets, cell instances or startpoints clocked by named clocks are to be reported. If *from_list* is not specified, the default behavior reports the longest path to an output port if the design has no timing constraints. Otherwise, the default behavior is to report the path with the worst slack within each path group if the design has timing constraints.

-rise_from *rise_from_list*
Same as the **-from** option, except that the path must rise from the objects specified. If a clock object is specified, this option selects startpoints clocked by the named clock, but only the paths launched by rising edge of the clock at the clock source, taking into account any logical inversions along the clock path.

-fall_from *fall_from_list*
Same as the **-from** option, except that the path must fall from the objects specified. If a clock object is specified, this option selects startpoints clocked by the named clock, but only the paths launched by falling edge of the clock at the clock source, taking into account any logical inversions along the clock path.

where the arguments have the following meaning:

[-from *pins-and-ports*]	Select all timing paths starting at one of the pins or ports from the specified collection.
[-rise_from *pins-and-ports*]	Select timing paths for a rising signal starting at one of the pins or ports from the specified collection.
[-fall_from *pins-and-ports*]	Select timing paths for a falling signal starting at one of the pins or ports from the specified collection.
[-to *pins-and-ports*]	Select all timing paths ending at one of the pins or ports from the specified collection.
[-rise_to *pins-and-ports*]	Select timing paths for a rising signal ending at one of the pins or ports from the specified collection.
[-fall_to *pins-and-ports*]	Select timing paths for a falling signal ending at one of the pins or ports from the specified collection.

116

-to *to_list*
Specifies that only paths to the named pins, ports, nets, cell instances or endpoints clocked by named clocks are to be reported. If *to_list* is not specified, the default behavior reports the longest path to an output port if the design has no timing constraints. Otherwise, the default behavior is to report the path with the worst slack within each path group if the design has timing constraints.

-rise_to *rise_to_list*
Same as the **-to** option, but applies only to paths rising at the endpoint. If a clock object is specified, this option selects endpoints clocked by the named clock, but only the paths captured by rising edge of the clock at clock source, taking into account any logical inversions along the clock path.

-fall_to *fall_to_list*
Same as the **-to** option, but applies only to paths falling at the endpoint. If a clock object is specified, this option selects endpoints clocked by the named clock, but only the paths launched by falling edge of the clock at the clock source, taking into account any logical inversions along the clock path.

-exclude *exclude_list*
Specifies that only paths not including the named pins, ports, nets, cell instances in the data paths are to be reported. Reporting will exclude all data paths from/through/to the named pins, ports, nets and cell instances. If a cell instance is specified, all pins of the cell are excluded. **-exclude** has higher precedence than **-from/-through/-to**. **-exclude** does not work with -true option. **-exclude** is exclusive with **-rise_exclude** or **-fall_exclude**. **-exclude** does not apply to borrowing path from **-trace_latch_borrow** option or clock path from **-path full_clock/full_clock_expanded** options.

-rise_exclude *rise_exclude_list*
Same as the **-exclude** option, but applies only to paths rising at the named pins, ports, nets, cell instances.

-fall_exclude *fall_exclude_list*
Same as the **-exclude** option, but applies only to paths falling at the named pins, ports, nets, cell instances.

-through *through_list*
Specifies that only paths through the named pins, ports, cell instances or nets are to be reported. If *through_list* is not specified, the default behavior reports the longest path to an output port if the design has no timing constraints. Otherwise, the default behavior reports the path with the worst slack within each path group if the design has timing constraints. If you specify **-through** only once, PrimeTime reports only the paths that travel through one or more of the objects in the list. You can specify **-through** more than once in one command invocation. For a discussion of the use of multiple **-through**, **rise_through**, and **fall_through** options, see the DESCRIPTION section.

-rise_through *through_list*
This option is similar to the **-through** option, but applies only to paths with a rising transition at the specified objects. You can specify **-rise_through** more than once in a single command invocation. For a discussion of multiple **-through**, **-rise_through**, and **-fall_through** options, see the DESCRIPTION section.

-fall_through *through_list*
This option is similar to the **-through** option, but applies only to paths with a falling transition at the specified objects. You can specify **-fall_through** more than once in a single command invocation. For a discussion of multiple **-through**, **rise_through**, and **fall_through** options, see the DESCRIPTION section.

-delay_type *delay_type*
Specifies the type of path delay to be reported. Valid values are **max** (the default), **min**, **min_max**, **max_rise**, **max_fall**, **min_rise**, or **min_fall**. The "rise" or "fall" in the *delay_type* refers to a rising or falling transition at the path endpoint.

-nworst *paths_per_endpoint*
Specifies the number of paths to be reported per endpoint per path group. Allowed values are 1 to 2000000; the default is 1.

[-through *pins-and-ports*]	Select all timing paths passing through one of the pins or ports from the specified collection.
[-rise_through *pins-and-ports*]	Select all timing paths with a rising signal passing through one of the pins or ports from the specified collection.
[-fall_through *pins-and-ports*]	Select all timing paths with a falling signal passing through one of the pins or ports from the specified collection.
[-delay_type max | min | min_max | max_rise | max_fall | min_rise | min_fall]	Select paths based on the type of delay they have. The following types of delay are supported: *max*—Paths with a maximum delay *min*—Paths with a minimum delay *min_max*—Paths with a minimum or maximum delay *max_rise*—Currently, same as *max* *max_fall*—Currently, same as *max* *min_rise*—Currently, same as *min* *min_fall*—Currently, same as *min*
[-nworst *number*]	Number of paths reported per end point. The default value is *1*.
[-max_paths *number*]	Number of paths reported per path group. The default value is *1*.
[-path_type short | full | full_clock | full_clock_expanded | end | summary]	Controls how paths are reported. The following options are supported: *short*—Only show the launch pins and the capture pins and skip all intermediate pins for the -nworst paths of each path group. *full*—Show all pins along the -nworst paths of each path group from launch pin to capture pin. This is the default report. *full_clock*—Report the -nworst paths for each path group and include the timing of the clock from the clock root to the clock of the state element that launches the signal and the clock to the state element that captures the signal. *full_clock_expanded*—Same as full_clock, but also include the clock path from the original clock to the generated clock. *end*—Only show the slack value at the capture pin of the -nworst paths of each path group. *summary*—Report the slack at the capture pin of the worst path across all path groups.
[-reason]	Report the reason why the optimizer could not optimize a particular cell or net. Possible reasons are:

117

-max_paths *count*
Specifies the number of paths to be reported per path group. Allowed values are 1 to 2000000; the default value is equal to the -nworst setting.

-path_type *format*
Specifies the format of the path report and how the timing path is displayed. Allowed values are **short**, which displays only start and end points "in the timing path"; **full** (the default), which displays the full path; **full_clock**, which displays full clock paths in addition to the full timing path; **end**, which generates a report with a column format that shows one line for each path and shows only the endpoint path total, required-time, slack and CRP (clock reconvergence pessimism value) when the variable timing_remove_clock_reconvergence_pessimism is set to TRUE; and **summary**, which displays only the path without the accompanying required-time and slack calculation; **full_clock_expanded**, which displays full clock paths between a primary clock and a related generated clock in addition to the full_clock timing path.

-true
Indicates that the longest (least-slack) true paths in the design are to be reported. This option can require long runtimes for certain designs that have many false paths. The variables **true_delay_prove_true_backtrack_limit** and **true_delay_prove_false_backtrack_limit** are used to limit the amount of backtracking during the operation of **report_timing -true**. The command **set_case_analysis** is used to specify a partial input vector to be considered for **-true** analysis. The **-true** option cannot be combined with **-max_paths**(1), **-nworst**(1), **-delay_type** (path type other than *max*) , **-unique**, **-rise/fall_through** and **-rise/fall_from** options.

-true_threshold *path_delay*
Used with the **-true** option. Specifies a threshold path delay value, in library time units, used by the **-true** option to speed up the searching. If this option is specified, **report_timing -true** returns the first path it finds greater than or equal to *path_delay* rather than continuing to search for a longer one.

-justify
Indicates to find and report an input vector that sensitizes the reported paths, or to report the path as false if no input vector is found. The **set_case_analysis** command is used to specify a partial input vector to be considered for **-justify** analysis.

-false
Indicates that only false paths are to be reported. These are the paths where no sensitizing input vector is found. The **set_case_analysis** command is used to specify a partial input vector to be considered for **-false** analysis.

-input_pins
Indicates that input pins are to be shown in the path report. The default is to show only output pins.

-unique_pins
Indicates that only paths through a unique set of pins are to be reported. This option can require longer runtimes when used in combination with the -*nworst* option with a large number of paths targeted for reporting.

-start_end_pair
Indicates that paths are reported for each pair of startpoint and endpoint based on connectivity. This option can lead to long runtime and can lead to generating a huge number of paths depending on the design. By default this option will only search for paths which are violating. This default value can be changed by having an explicit **-slack_lesser_than** option. The options that do not work with this option are **-nworst**, **-max_paths**,**-unique_pins**, **-true**, **-false**, **-justify**, **-slack_greater_than**,**-ignore_register_feedback**, **-report_ignored_register_feedback**. Unlike with other options of report_timing, this option causes the paths reported to no longer be sorted based on slack, instead, paths are arranged based on the endpoint with those sharing the same endpoint appearing next to one another. The maximum number of paths reported is limited to 2000000. In order to avoid the potential of returning duplicate paths, this option works as though the variable timing_report_always_use_valid_start_end_points was set to true.

-nets
Indicates that nets are to be shown in the path report. The default is not to show nets.

L—Do-not-touch cell
d—Do-not-touch net
S—Preserved cell
s—Preserved net
F—Fixed cell
m—Multi-driven net
t—Tristate net
M—Multi-mode clock net

[-significant_digits *digits*]	Display the numbers with the specified number of digits after the decimal point.
[-nosplit]	Do not split lines if they do not fit the width of a page.
[-transition_time]	Report transition time.
[-coordinate]	Include the location for reported pins and cells.
[-capacitance]	Report total capacitance.
[-crosstalk_delta]	Not supported yet.
[-trace_latch_borrow]	Not supported yet.
[-derate]	Not supported yet.
[-html]	Report in HTML format.
[-noenvironment]	Do not report environment variables.
[-scenario *scenario*]	Name of work scenario for which to report timing information.
[-internal_path]	Do no report on timing paths that start or end at I/O ports.
[-no_hierarchical_pins]	Do not report hierarchical pins.
[-no_buffer_inverter_on_clock]	Do not report buffer and inverter on clock paths.
[-summary]	Include timing summary information. This argument also honors the *-nworst* argument.
[-brief_summary]	Include a brief timing summary information.
[-histogram]	Include an end pin slack histogram.
[-step_of_histogram]	Width of buckets to use when generating a histogram.
[-true]	Not supported yet.
[-true_threshold *threshold*]	Not supported yet.
[-justify]	Not supported yet.
[-false]	Not supported yet.

118

-slack_greater_than *slack_limit*
Indicates that only those paths with a slack greater (more positive) than *slack_limit* are to be shown. This option is applied as a filter to the paths after they are generated. Therefore, the number of paths generated may be less than the number specified with the -nworst and -max_paths options. This option can be combined with **-slack_lesser_than** to show only those paths inside or outside a given slack range.

-slack_lesser_than *slack_limit*
Indicates that only those paths with a slack less (more negative) than *slack_limit* are to be shown. This option can be combined with **-slack_greater_than** to select only those paths inside or outside a given slack range.

-ignore_register_feedback *feedback_slack_cutoff*
Indicates that non-inverting timing loops should be ignored if they start and end at the same register pin that holds a value. To be ignored, the data-to-output arc and the output-to-data path must either both be inverting or both be non-inverting. This option applies to min delay as well as max delay reports. Paths are ignored only if they have a slack less than the specified *feedback_slack_cutoff*. This option is applied as a filter to the paths after they are generated. Therefore, the number of paths generated may be less than the number specified with the -nworst and -max_paths options.

-report_ignored_register_feedback
Indicates that paths are to be reported if they are ignored when the **-ignore_register_feedback** option is specified.

-group *group_name*
Specifies the path groups from which timing paths are selected for reporting based on other specified options for reports.

-transition_time
Indicates that transition time (slew) is to be shown in the path report. The default is not to show transition time. For each driver pin or load pin the transition time is displayed in a column preceding incremental path delay.

-capacitance
Indicates that total (lump) capacitance is to be shown in the path report. The default is not to show capacitance. For each driver pin the total capacitance driven by the driver is displayed in a column preceding both incremental path delay and transition time (with **-transition_time**). When **-nets** is specified, the capacitance is printed on the lines with nets instead of the lines with driver pins.

-crosstalk_delta
Indicates that annotated delta delay and delta transition time is reported. The deltas are computed during crosstalk signal integrity analysis, or they can be annotated manually using **set_annotated_delay -delta_only** and **set_annotated_transition -delta_only**. Note that the **-crosstalk_delta** only reports the calculated or annotated deltas, it does not initiate crosstalk analysis. Only deltas on input pins are shown. Delta transition time is shown only with **-transition_time**. The **-crosstalk_delta** automatically sets **-input_pins**.

-derate
Indicates that derate factors are to be shown in the timing report. The default is to show no derate factors. Specifying this option automatically sets both **-input_pins** and **-path_type full_clock_expanded**. For each output pin of a cell in the report that cells derate factor used is displayed in a column preceding the incremental path delay. For each input pin of a cell in the report its preceding nets derate factors is displayed in a column preceding the incremental path delay. In addition a summary report will follow the timing report indicating what portion of the slack is a result of the application of derate factors.

-significant_digits *digits*
Specifies the number of digits after the decimal point to be displayed for time values in the generated report. Allowed values are 0-13; the default is determined by the **report_default_significant_digits** variable, whose default value is *2*. Use this option if you want to override the default. This option controls only the number of digits displayed, not the precision used internally for analysis. For analysis, PrimeTime uses the full precision of the platform's fixed-precision, floating-point arithmetic capability.

[-input_pins]	Show input pins.
[-unique_pins]	Not supported yet.
[-start_end_pair]	Not supported yet.
[-arrival_time_count *count*]	Report the average arrival time counts found on a pin. It is calculated by dividing the sum of all arrival time counts across all pins by the total number of pins. This number gives a good indication of the complexity of the design with respect to timing. The run time, typically, is proportional to the design size and proportional to this average arrival time count.
[-nets]	Show the net names in the path report. By default, only pins are shown.
[-slack_greater_than *threshold*]	Only report path whose slack is greater than the specified threshold. The default value is *-1e+20*.
[-slack_lesser_than *threshold*]	Only report path whose slack is smaller than the specified threshold. The default value is *1e+20*.
[-ignore_register_feedback *value*]	Not supported yet.
[-report_ignored_register_feedback]	Not supported yet.
[-group { *name* [*name*]... }]	Report on the specified groups.
[-unconstrained_path]	Report unconstrained paths if no constrained path is available.
[-prefix *filename_prefix*]	If no scenario is set, only one report is created and is named *filename_prefix*. If you have a working scenario specified with the *set_working_scenario* Tcl command, only one report is created. Its name starts with *filename_prefix*, followed by a dot and the name of that scenario. If you are in MCMM mode, and have several scenarios set with the *current_session* Tcl command, a report is generated for each of the scenarios in the session. The names of the reports start with *filename_prefix*, followed by a dot, and the name of that scenario. Note that the *-prefix* argument redirects the report to a file. Hence, it has precedence over the > redirection operator.
[-aocvm]	For the reported timing paths, list the derating factors applied to the various gate and net delays due to advanced on-chip variation (OCV) modeling. Advanced OCV models are specified using the *read_aocvm* and the *set_timing_derate -aocvm_guardband* Tcl commands.
[-voltage]	Report the voltage applied to the cells. You typically request this report if you applied supply voltages manually using the *set_rail_voltage* Tcl command, or allowed automatic updates from the power network analyser by setting the *ta* parameter *timing_back_annotate_rail_voltage* to *true*.

119

-nosplit

Most of the design information is listed in fixed-width columns. If the information in a given field exceeds the column width, the next field begins on a new line, starting in the correct column. The **-nosplit** option prevents line-splitting and facilitates writing software to extract information from the report output.

-trace_latch_borrow

This option controls the type of report generated for a path that starts at a transparent latch. If the path startpoint borrows from the previous stage, using this option causes the report to show the entire set of borrowing paths that lead up to the borrowing latch, starting with a nonborrowing path or a noninverting sequential loop. By default, the report shows only the last path in the sequence of borrowing stages. Each stage is reported separately, showing the time borrowed and lent and the endpoints of the stage. The cumulative amount of borrowed time along a sequence of stages is not included in the report. The options **-input_pins, -nets, -transition_times, -capacitance**, and **-significant_digits** apply to every stage in the sequence of borrowing paths, but the remaining options (for example, **-from** and **-true**) apply only to the last stage reported.

-dont_merge_duplicates

This option is available only if the user invokes PrimeTime with the -multi_scenario option. It turns OFF a main capability in merged reporting that is ON by default. The option affects the manner in which paths from multiple scenarios are merged. By default, when the same path is reported from more than one scenario, PrimeTime reports only the single most critical instance of that path in the merged report and shows its associated scenario. By using this option, PrimeTime will not merge duplicate instances of the same path into a single instance, but instead shows all critical instances of the path from all scenarios. Since the number of paths reported is limited by the -nworst, -max_paths and other options of this command, the resulting merged report, when this option is used, may not be evenly spread out across the design, but instead may be focussed on the portion of the design that is critical in each scenario.

-pre_commands *pre_command_string*

This option is available only if the user invokes PrimeTime with the -multi_scenario option. This option allows users to specify a string of commands to be executed in the slave context before the execution of the merged_reporting command. Commands must be grouped using the ";" character. The maximum size of a command is 1000 chars.

-post_commands *post_command_string*

This option is available only if the user invokes PrimeTime with the -multi_scenario option. This option allows users to specify a string of commands to be executed in the slave context after the execution of the merged_reporting commands. Commands are grouped using the ";" character. The maximum size of a command is 1000 chars.

-exceptions

Prints user-entered timing exceptions, namely false paths, multi-cyle paths, and min/max delays, that are satisfied per timing path being reported. The -exceptions options requires one and only one of the following three values: **dominant**, **overridden**, and **all**. Please note that the additional analysis required per path with -exceptions is non-trivial. Therefore, a **report_timing** with -exceptions is expected to execute slower than the exact same command without the -exceptions option. **-exceptions** does not work with **-path_type short/end/summary** option.

-aocvm

This option indicates that the timing paths are to be adjusted using AOCVM information. The order in which the paths are printed matches the order in which the paths would have been printed had this option not been specified. This option automatically sets **-derate** and **-path_type full_clock_expanded**. AOCVM derate factors are shown in the *Derate* column of the timing report.

-recalculate

Indicates that path recalculation should be applied during the search. The worst recalculated paths meeting the path requirements are returned. This option can result in long runtimes due to the path searching required. This option does not work with **-aocvm**, **-justify**, **-true**, **-slack_greater_than** and other multi scenario options, including **-pre_commands**, **-post_commands**, **-dont_merge_duplicates** and **-attributes**.

collection1

Specifies the collection of timing paths to report. This option is mutually exclusive of options which control the selection of paths to report and is only compatible with options which control the formatting of the report.

[-variation]

For all timing results, show both the mean delay and the standard deviation.

[-sort_by_slack]

Sort the timing paths by slack regardless of their path groups. By default, paths are listed per path group.

120

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790846

report_transitive_fanin

Reports logic in fan-in of specified sink objects.

SYNTAX

```
string report_transitive_fanin [-nosplit] -to sink_list
list  sink_list
```

ARGUMENTS

-nosplit
Does not split lines if column overflows.

-to *sink_list*
Specifies a list of sink pins, ports, or nets in the design, whose transitive fan-in is reported. If a net is specified, the effect is the same as listing all driver pins on the net.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056981-2

report_transitive_fanin

Reports the fanin of listed to-pins.

Syntax

```
report_transitive_fanin \
   [-scenario scenario] \
   -to collection \
   [-from collection] \
   [-through collection]... \
   [-trace_arcs timing | enabled | all]
   [-quiet] \
   [-level cellLevel] \
   [-pin_level pinLevel] \
   [-nosplit]
```

where the arguments have the following meaning:

[-scenario *scenario*]	Scenario for which data is requested.
-to *collection*	List of pins for which to report the fanin.
[-from *collection*]	Only report that part of the fanin that originates from a pin from the specified collection.
[-through *collection*]...	Only report that part of the fanin that touches one of the pins from one of the specified collections.
[-trace_arcs timing | enabled | all]	Possible values are: *timing*—Stop at disabled cell arcs and timing arcs that have been cut by case analysis. This is the default value. *enabled*—Stop at disabled timing arcs, but ignore case analysis. *all*—Ignore disabled timing arcs and case analysis. Do tracing purely netlist-based.
[-quiet]	Suppress warning messages.
[-level *cellLevel*]	Include pins of all cells that are at a distance up to *cellLevel* cell arcs from the root pin.
[-pin_level *pinLevel*]	Include all pins that are at a distance up to *pinLevel* pin arcs from the root pin.
[-nosplit]	Do not split lines in columns whose lines go beyond the column width.

121

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790848

report_transitive_fanout

Reports logic in fanout of specified sources.

SYNTAX

```
string report_transitive_fanout [-nosplit] -clock_tree -from source_list
list  source_list
```

ARGUMENTS

-nosplit
Does not split lines if column overflows.

-clock_tree
Reports transitive fanout of all clock sources in the design.

-from *source_list*
Specifies a list of source pins, ports, and nets in the design whose transitive fanout is reported. If a net is specified, the effect is same as listing all the load pins on the net.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0056985-86

report_transitive_fanout

Reports the fanout of listed to-pins.

Syntax

```
report_transitive_fanout \
   [-scenario scenario] \
   { -from collection | -clock_tree clocks} \
   [-to collection] \
   [-through collection]... \
   [-trace_arcs timing | enabled | all] \
   [-quiet] \
   [-level cellLevel] \
   [-pin_level pinLevel] \
   [-include_hierarchical_pins] \
   [-nosplit]
```

where the arguments have the following meaning:

[-scenario *scenario*]	Scenario for which data is requested.
-from *collection*	List of pins for which to report the fanout. Either this argument or the *-clock_tree* argument must be specified.
-clock_tree clocks	Trace fanout of clock sources. Either this argument or the *-from* argument must be specified.
[-to *collection*]	Only report paths ending in a pin from the specified collection.
[-through *collection*]...	Only report that part of the fanin that touches a pin from one of the specified collections.
[-trace_arcs timing | enabled | all]	Possible values are: *timing*—Stop at disabled cell arcs and timing arc that have been cut by case analysis. This is the default value. *enabled*—Stop at disabled timing arcs, but ignore case analysis.

122

	all—Ignore disabled timing arcs and case analysis. Do tracing purely netlist based.
`[-level integer]`	Number of traced arcs.
`[-quiet]`	Suppress warning messages.
`[-level cellLevel]`	Include pins of all cells that are at a distance up t *cellLevel* cell arcs from the root pin.
`[-pin_level pinLevel]`	Include all pins that are at a distance up to *pinLevel* pin arcs from the root pin.
`[-include_hierarchical_pins]`	Include the module ports of instantiated modules in the report.
`[-nosplit]`	Do not split lines in columns whose lines go beyond the column width.

123

reset_path

Resets specified paths to single-cycle behavior.

SYNTAX

```
Boolean reset_path
[-setup] [-hold]
[-rise] [-fall]
[-from from_list
   | -rise_from rise_from_list
   | -fall_from fall_from_list]
[-through through_list]*
[-rise_through rise_through_list]*
[-fall_through fall_through_list]*
[-to to_list
   | -rise_to rise_to_list
   | -fall_to fall_to_list]

list  from_list
list  rise_from_list
list  fall_from_list
list  through_list
list  rise_through_list
list  fall_through_list
list  to_list
list  rise_to_list
list  fall_to_list
```

ARGUMENTS

-setup

Indicates that only setup (maximum delay) evaluation is to be reset to its default, single-cycle behavior. If neither **-setup** nor **-hold** is specified, both setup and hold checking are reset to single-cycle.

-hold

Indicates that only hold (minimum delay) evaluation is to be reset to its default, single-cycle behavior. If neither **-setup** nor **-hold** is specified, both setup and hold checking are reset to single-cycle.

-rise

Indicates that only rising path delays are to be reset to single-cycle behavior. If neither **-rise** nor **-fall** is specified, both rising and falling delays are reset to single-cycle.

reset_path

Removes path exceptions that have been set using the *set_false_path* and *set_multicycle_path* Tcl commands.

Syntax

```
reset_path [-setup] \
   [-hold] \
   [-rise] \
   [-fall] \
   [-from pin-and-ports] \
   [-rise_from pin-and-ports] \
   [-fall_from pin-and-ports] \
   [-to pin-and-ports] \
   [-rise_to pin-and-ports] \
   [-fall_to pin-and-ports] \
   [-through pin-and-ports] \
   [-rise_through pin-and-ports] \
   [-fall_through pin-and-ports]
```

where the arguments have the following meaning:

[-setup]	Only remove path exceptions that were defined for setup checks.
[-hold]	Only remove path exceptions that were defined for hold checks.
[-rise]	Only remove path exceptions that were defined for rising signals.
[-fall]	Only remove path exceptions that were defined for falling signals.

-fall
Indicates that only falling path delays are to be reset to single-cycle behavior. If neither **-rise** nor **-fall** is specified, both rising and falling delays are reset to single-cycle.

-from *from_list*
Specifies a list of timing path startpoint objects. A valid timing startpoint is a clock, a primary input or inout port, a sequential cell, a clock pin of a sequential cell, a data pin of a level-sensitive latch, or a pin that has input delay specified. If a clock is specified, all registers and primary inputs related to that clock are used as path startpoints. If you specify a cell, one path startpoint on that cell is affected. You can use only one of **-from**, **-rise_from**, and **-fall_from**.

-rise_from *rise_from_list*
Same as the **-from** option, except that the path must rise from the objects specified. If a clock object is specified, this option selects startpoints clocked by the named clock, but only the paths launched by rising edge of the clock at the clock source, taking into account any logical inversions along the clock path. You can use only one of **-from**, **-rise_from**, and **-fall_from**.

-fall_from *fall_from_list*
Same as the **-from** option, except that the path must fall from the objects specified.If a clock object is specified, this option selects startpoints clocked by the named clock, but only the paths launched by falling edge of the clock at the clock source, taking into account any logical inversions along the clock path. You can use only one of **-from**, **-rise_from**, and **-fall_from**.

-through *through_list*
Specifies a list of pins, ports, and nets through which the paths must pass that are to be reset. Nets are interpreted to imply the leaf-level driver pins. If you omit **-through**, all timing paths specified using the **-from** and **-to** options are affected. You can specify **-through** more than once in one command invocation. For a discussion of the use of multiple **-through** options, see the DESCRIPTION section.

-rise_through *rise_through_list*
This option is similar to the **-through** option, but applies only to paths with a rising transition at the through points. You can specify **-rise_through** more than once in one command invocation. For a discussion of the use of multiple **-through** options, see the DESCRIPTION section.

-fall_through *fall_through_list*
This option is similar to the **-through** option, but applies only to paths with a rising transition at the through points. You can specify **-fall_through** more than once in one command invocation. For a discussion of the use of multiple **-through** options, see the DESCRIPTION section.

-to *to_list*
Specifies a list of timing path endpoint objects. A valid timing endpoint is a clock, a primary output or inout port, a sequential cell, a data pin of a sequential cell, or a pin that has output delay specified. If a clock is specified, all registers and primary outputs related to that clock are used as path endpoints. If a cell is specified, one path endpoint on that cell is affected. You can use only one of **-to**, **-rise_to**, and **-fall_to**.

-rise_to *rise_to_list*
Same as the **-to** option, but applies only to paths rising at the endpoint. If a clock object is specified, this option selects endpoints clocked by the named clock, but only the paths captured by rising edge of the clock at clock source, taking into account any logical inversions along the clock path. You can use only one of **-to**, **-rise_to**, and **-fall_to**.

-fall_to *fall_to_list*
Same as the **-to** option,but applies only to paths falling at the endpoint. If a clock object is specified, this option selects endpoints clocked by the named clock, but only the paths launched by falling edge of the clock at the clock source, taking into account any logical inversions along the clock path. You can use only one of **-to**, **-rise_to**, and **-fall_to**.

[-from *pin-and-ports*]	Only remove path exceptions on paths that originate at one of the specified pins or ports.
[-rise_from *pin-and-ports*]	Only remove path exceptions on paths that originate with a rising signal at one of the specified pins or ports.
[-fall_from *pin-and-ports*]	Only remove path exceptions on paths that originate with a falling signal at one of the specified pins or ports.
[-to *pin-and-ports*]	Only remove path exceptions on paths that end at one of the specified pins or ports.
[-rise_to *pin-and-ports*]	Only remove path exceptions on paths that end with a rising signal at one of the specified pins or ports.
[-fall_to *pin-and-ports*]	Only remove path exceptions on paths that end with a falling signal at one of the specified pins or ports.
[-through *pin-and-ports*]	Only remove path exceptions on paths that pass through one of the specified pins or ports.
[-rise_through *pin-and-ports*]	Only remove path exceptions on paths that pass through one of the specified pins or ports with a rising signal.
[-fall_through *pin-and-ports*]	Only remove path exceptions on paths that pass through one of the specified pins or ports with a falling signal.
[-trace_arcs timing | enabled | all]	Possible values are: *timing*—Stop at disabled cell arcs and timing arcs that have been cut by case analysis. This is the default value. *enabled*—Stop at disabled timing arcs, but ignore case analysis. *all*—Ignore disabled timing arcs and case analysis. Do tracing purely netlist based.
[-level *integer*]	Number of traced arcs.
[-quiet]	Suppress warning messages.
[-level *cellLevel*]	Include pins of all cells that are at a distance up to *cellLevel* cell arcs from the root pin.
[-pin_level *pinLevel*]	Include all pins that are at a distance up to *pinLevel* pin arcs from the root pin.
[-include_hierarchical_pins]	Include the module ports of instantiated modules in the report.
[-nosplit]	Do not split lines in columns whose lines go beyond the column width.

125

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790872	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057006
reset_timing_derate Resets user specified derate factors set either on a design or on a specified list of instances (cells, nets or library cells).	**reset_timing_derate** Removes timing derating factors set with the *set_timing_derate* Tcl command.
SYNTAX string **reset_timing_derate** [-hierarchical_net_delay] [-scalar] [-variation] *object_list* list *object_list*	**Syntax** reset_timing_derate \ [-aocvm]
ARGUMENTS -hierarchical_net_delay Indicates that net derate factors within hierarchical cells should also be reset. -scalar Indicates that only derate factors for deterministic delays should be reset. -variation Indicates that only derate factors for statistical delays should be reset. object_list Specifies a list of designs, cells, library cells or nets which will be reset.	where the argument has the following meaning: [-aocvm] — Only reset derating factors related to aocvm, that is, factors set by the *set_timing_derate -guardbound* Tcl command.

126

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790887-89

set_annotated_delay

Sets the net or cell delay value between two pins.

SYNTAX

```
string set_annotated_delay -cell | -net
[-rise]
[-fall]
[-min]
[-max]
[-load_delay load_delay_type]
[-from from_pins]
[-to to_pins]
[-cond sdf_expression]
[-increment]
[-delta_only]
[-worst]
-variation variation_object
delay_value

stringload_delay_type
list  from_pins
list  to_pins
stringsdf_expression
float delay_value
```

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057096-97

set_annotated_delay

Specifies a delay between two pins. This delay can be a cell delay from an input pin of a cell to an output pin of the same cell, or a net delay from an output pin of a cell to an input pin of another cell.

This delay replaces or is added to the delay calculated by the timing analyzer.

The delay may only be valid for either a min-timing or max-timing analysis and may be only valid for rising or falling edges of a signal.

Annotated delays typically are read in from an SDF file generated by a third-party timing analysis tool.

Syntax

```
set_annotated_delay delay \
   -from collection \
   -to collection \
   [-net] \
   [-cell] \
   [-rise] \
   [-fall] \
   [-min] \
   [-max] \
   [-cond string] \
   [-load_delay string] \
   [-increment] \
   [-delta_only] \
   [-worst]
```


127

ARGUMENTS

-cell
Specifies that the delay annotated is a cell delay. The **-cell** and **-net** arguments are mutually exclusive; you must specify one, but not both.

-net
Specifies that the delay annotated is a net delay. The **-net** and **-cell** arguments are mutually exclusive; you must specify one, but not both.

-rise
Indicates that the delay is for the data rise transition. If you do not specify either **-rise** or **-fall**, both values are set.

-fall
Indicates that the timing check is for the data fall transition. If you do not specify either **-rise** or **-fall**, both values are set.

-min
Use this option only if the design is in min_max mode (min and max operating conditions). Specifies the minimum delay for both data rise and data fall transitions.

-load_delay *load_delay_type*
Specifies whether load delay is to be included as part of annotated net delays or as part of annotated cell delays. Allowed values are **net** or **cell**. Load delay is the portion of cell delay resulting from the capacitive load of the net the cell is driving. All timing arcs of the same net and of the same cell, must be annotated with the same *load_delay_type*.

-from *from_list*
Specifies a list of leaf cell pins and top level ports that are the startpoints of the timing arcs for which delays are annotated.

-to *to_list*
Specifies a list of leaf cell pins and/or top level ports that are the endpoints of the timing arcs for which delays are annotated.

-cond *sdf_expression*
Use this option only if the library has a condition attached to the specified delay timing arc; otherwise, an error message is generated. Specifies the condition for which the annotated delay is valid. The syntax of the condition must match the condition specified in the library using the construct sdf_cond. The syntax is the same one used in the Standard Delay Format (SDF).

-increment
Specifies that the delay is to be incremented to the current delay of the specified timing arc.

where the arguments have the following meaning:

delay	Delay in pico seconds.
-from *startpins*	Set delay only on arcs that start from a pin in the set.
-to *endpins*	Set delay only on arcs that end in a pin in the set.
[-net]	Delay only models the net delay.
[-cell]	Delay only models the gate delay.
[-rise]	Only use this delay for a rising signal at the input pin.
[-fall]	Only use this delay for a falling signal at the input pin.
[-min]	Only use this delay for min-timing, that is, hold analysis.
[-max]	Only use this delay for max-timing, that is, setup analysis.
[-increment]	Add this delay to whatever the timing analyzer calculates based on cell timing models and net parasitics.
[-delta_only]	Indicates that this is a delta delay, i.e. a change in delay, that can be positive or negative.

128

-delta_only
Specifies that the annotated delay is to be added to the net delay value calculated by PrimeTime. You cannot use this option with **-cell**.

-worst
This option is not yet implemented, so it is ignored.

delay_value
Specifies the delay value between pins on the same cell, in units consistent with the technology library used during optimization. For example, if the technology library specifies delay values in nanoseconds, *delay_value* must be expressed in nanoseconds.

-variation *variation_object*
Specify a variation to annotate on all arcs between the from and to pins. The variation_object must be created using the create_variation command.

129

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790893

set_annotated_transition

Sets the transition time to be annotated on specified pins in the current design.

SYNTAX

```
int set_annotated_transition [-rise][-fall][-min][-max] [-delta_only] slew_value
pin_list
float slew_value
list  pin_list
```

ARGUMENTS

-rise
: Indicates that *slew_value* represents the data rise transition time.

-fall
: Indicates that *slew_value* represents the data fall transition time.

-min
: Indicates that *slew_value* represents the minimum transition time. Use this option only if the design is in min-max mode (min and max operating conditions).

-max
: Indicates that *slew_value* represents the maximum transition time. Use this option only if the design is in min-max mode (min and max operating conditions).

-delta_only
: Indicates that *slew_value* represents a delta transition time to be added to the transition time computed by delay calculation.

slew_value
: Specifies the transition time of the specified pins or ports, in units consistent with the technology library used during optimization. For example, if the technology library specifies delay values in nanoseconds, *slew_value* must be expressed in nanoseconds. If used with the **-delta_only** option, *slew_value* can be a negative number.

pin_list
: Specifies a list of pins or ports to be annotated with the transition time *slew_value*.

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057099

set_annotated_transition

Sets a transition time on pins and ports that overrides the transition time that is otherwise calculated by the timing analyzer based on cell models, net parasitics, or wireload models.

Syntax

```
set_annotated_transition transition_time objects \
   [-rise] \
   [-fall] \
   [-min] \
   [-max] \
   [-delta_only]
```

where the arguments have the following meaning:

transition_time	Time it takes for a signal to transition from high to low or vice versa.
objects	Pins and ports on which this transition is set.
[-rise]	Transition time is only valid for a rising signal.
[-fall]	Transition time is only valid for a falling signal.
[-min]	Only use this transition time during a min-timing analysis, that is, during hold analysis.
[-max]	Only use this transition time during a max-timing analysis, that is, during setup analysis.
[-delta_only]	Indicates that the transition time is a delta time, that is, a variation that might be positive or negative.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790895	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_aocvm_coefficient Specifies AOCVM random or systematic coefficients on library cells for use during an AOCVM analysis.	Command: set_aocvm_coefficient [double:value] <db:library_cell_or_leaf_cell_list> standard SDC command
SYNTAX int **set_aocvm_coefficient** [-random *path_depth_scalar* | -systematic *systematic_component_scalar*] *object_list* float *path_depth_scalar* float *systematic_component_scalar* list *object_list*	option: -random double(1.000) path depth coefficient -systematic double(1.000) systematic component coefficient -sigma_divider force to divide the sigma extracted for pocvm --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --system_default use system default values --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -random *path_depth_scalar* Indicates that *path_depth_scalar* is the random coefficient for the specified library cells in the *object_list*. The *path_depth_scalar* should be a floating-point value greater than zero. -systematic *systematic_component_scalar* Indicates that *systematic_component_scalar* is the systematic coefficient for the specified library cells in the *object_list*. The *systematic_component_scalar* should be a floating-point value greater than zero. object_list Specifies a list of library cells on which the coefficient is set.	description: This command is the same as standard SDC command.

131

set_aocvm_component

Specifies AOCVM random or systematic component on the top-level design for use during an AOCVM analysis.

SYNTAX

```
int set_aocvm_component
        [-early | -late]
        [-cell_delay] [-net_delay]
        [-random path_depth | -systematic path_distance]
        value


int    path_depth
float  path_distance
float  value
list   object_list
```

ARGUMENTS

-early
: Indicates that the component will apply to delay arcs that are early derated.

-late
: Indicates that the component will apply to delay arcs that are late derated.

-cell_delay
: Indicates that the component will apply to cell arc delays only.

-net_delay
: Indicates that the component will apply to net arc delays only.

-random *path_depth*
: Indicates that *value* is the random component to be used at a path depth of *path_depth*. The *path_depth* should be an integer value greater than zero.

-systematic *path_distance*
: Indicates that *value* is the systematic component to be used at a path distance of *path_distance*. The *path_distance* should be a floating-point value greater than zero.

value
: Indicates the fractional quantity of process variation for this component with respect to nominal arc delay to be set on the design. The *value* should be a floating-point number between 0 and 1.

set_aocvm_component

NOTE: This command is deprecated. Use the read_aocvm Tcl command to read the advanced on-chip variation derating information.

Configure the derating factors used during advanced on-chip variation analysis.

You can create a derating table by invoking the *set_aocvm_component* Tcl command multiple times with different values for the depth or distance. Aprisa will do linear interpolation or extrapolation to calculate derating factors for depths or distances for which no derating factor is specified.

Syntax

```
set_aocvm_component \
   [-rise | -fall] \
   [-late | -early] \
   [-cell_delay | -net_delay] \
   -random | -systematic \
   depth_or_distance] \
   derate_factor]
```

where the arguments have the following meaning:

[-rise | fall]	Controls whether the setting holds for a rising or falling event. By default, the setting holds for both.
[-late | early]	Controls whether the setting holds for an early path (signal path for hold analysis, or clock path for setup analysis) or late path (signal path for setup analysis, clock path for hold analysis).
[-cell_delay | -net_delay]	Controls whether the setting holds for cell or net delays. By default, both are derated by this setting.
-random | -systematic	Controls whether a random or systematic variation is being described. For systematic variation, the distance is interpreted as the geographical distance of the path being derated. For random variations, the distance is interpreted as a logical depth, that is, the number of gates in the path.
depth_or_distance	Integer number that is either a geographical distance in database units for a systematic variation or a gate-level distance (that is, number of gates on a timing path) for a random variation.
derate_factor	A real number, which is the delta by which the cell or net delays on a path (or both) are derated, that is, their delays are scaled by a factor (1 + *derate_factor*).

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790944	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_disable_clock_gating_check Disables the clock gating check for specified objects in the current design.	Command: set_disable_clock_gating_check <db:objects> standard SDC command

SYNTAX
 	
string **set_disable_clock_gating_check** *object_list*
list *object_list*
	option:
	  --license list required licenses
	  --help display command help

ARGUMENTS
 	
object_list
  Specifies a list of cells and pins for which the clock gating check is to be disabled.
	description:
	  This command is the same as standard SDC command.

133

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790953	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057204
set_dont_touch Sets the **dont_touch** attribute on cells, nets, designs, and library cells to prevent synthesis from replacing or modifying them during optimization.	**set_dont_touch** Creates a *dont_touch* design constraint for selected cells and nets. These cells or nets cannot be deleted during optimization, though they can be physically moved. If you set a *dont_touch* constraint on a module, then all its cells and nets inherit the constraint. The constraint is only effective once you start simulating or optimizing the design, so subsequent simulator commands, clock definitions, and case analysis settings are not affected. When the simulation and optimization starts, Aprisa generates a *dont_touch* attribute on these cells and nets. To remove this *dont_touch* constraint, you must remove the constraint and you need to remove the *dont_touch* attributes. Use the *remove_dont_touch_network* command to remove the constraints in the database that were set with the *set_dont_touch_network* Tcl command, and use the *set_attribute* Tcl command to remove all *dont_touch* attributes on instances and nets.
SYNTAX string **set_dont_touch** *object_list* [*value*] list *object_list* Boolean*value*	**Syntax** set_dont_touch *nets_or_cells* \ [-cts *object*]
ARGUMENTS object_list Specifies a list of cells, nets, designs, or library cells on which to place the **dont_touch** attribute. value Specifies the value with which to set the **dont_touch** attribute. Allowed values are *true* (the default) or *false*.	where the arguments have the following meaning: *nets_or_cells* — Collection of nets and cells for which to set the *dont_touch* attribute. [-cts *object*] — Mark all cells and nets belonging to the clock subtree rooted by the specified pin, port, or clock with the *dont_touch* attribute.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790955	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_dont_touch_network Sets the dont_touch attribute on clock networks for synthesis.	Command: set_dont_touch_network <db:object_list> Set dont_touch on the cells and nets of downstream of one port/pin, or source point of clock.
SYNTAX string **set_dont_touch_network** *object_list* list *object_list*	option: --license list required licenses --help display command help
ARGUMENTS object_list Specifies a list of clocks, pins, or ports.	description: Set dont_touch on the cells and nets of downstream of one port/pin, or source point of clock.

135

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791020	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_min_library Sets the library to be used for minimum delay analysis The **set_min_library** command is used to relate a minimum conditions library to a maximum conditions library.	Command: set_min_library <string:max_library> standard SDC command
SYNTAX string **set_min_library** [-min_version *min_library*] [-none] *max_library* string*min_library* string*max_library*	option: -min_version string — name of min library -none — dissociate min library --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --system_default — use system default values --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
ARGUMENTS -min_version *min_library* The library for min analysis. This library is not to be used in the **link_path**. -none Dissociate max_library from its min library max_library The library for max analysis. This library should be used in the **link_path**.	

136

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791024	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_mode Selects the active mode of cell mode groups or design mode groups	Command: set_mode <list:mode_list> <db:instance_list>
SYNTAX Boolean **set_mode** [-type cell | design] [*mode_list*] [*instance_list*] list *mode_list* list *instance_list*	option: -type string(cell) not supported yet --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --system_default use system default values --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -type design | cell Indicates the type of mode to be made active. This option has the following mutually-exclusive valid values: **design** and **cell**. The **cell** value specifies that cell modes are to be made active. Cell modes are defined on library cells in the library. The **design** value specifies that design modes are to be made active. Design modes are user specified using **define_design_mode_group** and **map_design_mode**. If the **-type** option is ommitted from the command then this is equivalent to specifying **-type cell** mode_list Specifies a list of modes, each of which is to be made the active mode for its mode group. If -type has a cell value then the mode_list must contain only cell modes. If -type has a design value then the mode list must contain only design modes. instance_list Specifies a list of instances for which the specified cell modes are to be made active. This list must only be included with **-type cell**.	

137

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791041	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057318-19
set_noise_margin Sets noise margin for a library pin, port, or pin.	**set_noise_margin** Specifies noise margins on selected nets for the glitch noise analysis. Glitch analysis accept four different noise margins: • *below_low* is the amount a noise peak might pull down a low signal before it is considered an undershoot violation. • *above_low* is the amount a noise peak might pull up a low signal before it is no longer considered a low signal and thus a glitch violation. • *below_high* is the amount a noise peak might pull down a high signal before it is no longer considered a high signal and thus a glitch violation. • *above_high* is the amount a noise peak might pull up a high signal before it is considered an overshoot violation. *set_noise_margin* allows you to set values for the following: • *below_low* using the *-below_low* argument • *above_high* using the *-above_high* argument • *above_low* and *below_high* when neither the *-below_low* nor the *-above_high* argument is specified. Note that in this case the value is used for both margins. Note that the glitch analysis always does a regular glitch analysis on all pins when signal integrity is enabled. If no above_low and below_high margins are specified, it uses default values set by the ta parameters *si_noise_margin_above_high* and *si_noise_margin_below_high*.
SYNTAX int **set_noise_margin** [-above] [-below] [-low] [-high] *margin_value* *object_list* float *margin_value* list *object_list*	**Syntax** set_noise_margin \ *pin_list* \ [-above_high] \ [-below_low] noise*Margin*

138

ARGUMENTS

-above
Specifies the noise margin for above ground or power rail noise analysis region.

-below
Specifies the noise margin for below ground or power rail noise analysis region.

-low
Specifies the noise margin for ground rail noise.

-high
Specifies the noise margin for power rail noise.

margin_value
Specifies a margin value. The value is the input height in units of voltage. 1.0.

object_list
Specifies a list of lib-pins or ports.

where the arguments have the following meaning:

pin_list	List of pins for which the noise margins are specified and on which to perform overshoot or undershoot analysis.
[-above_high]	The specified margin holds for a high signal pulled further up.
[-below_low]	The specified margin holds for a low signal pulled further down.
noiseMargin	Amount of noise voltage that can be injected before it is considered a noise violation. If no arguments are given, this value is used for both the *above_low* and *below_high* threshold for the selected pins, thus overriding the equivalent *ta* parameter settings. If the *-above_high* argument is used, this margin is used for overshoot analysis, that is, a high signal pulled higher. If the *-below_low* argument is used, this margin is used for undershoot analysis, that is, a low signal pulled lower.

139

set_rail_voltage

Sets power rail voltage on cells.

SYNTAX

```
int set_rail_voltage
[-rail_value rvalue | -rail_list rname_value_list]
[-min]
[-max]
cell_list

float rvalue
list  rname_value_list
list  cell_list
```


set_rail_voltage

Overrides the voltages as set by the operating conditions for a specified list of standard cells. You can limit the override to only be used for min-time or max-time timing analysis.

Typically, rail voltage overrides are automatically created by Aprisa's power network analyzer, but you can use the *set_rail_voltage* Tcl command instead, for example, to import the results calculated from a third-party power network analyzer.

When you execute the *set_rail_voltage* Tcl command, Aprisa calculates the actual supply voltage of all affected cells and stores the result as persistent properties on these cells. The *set_rail_voltage* Tcl commands are typically specified in the SDC files. When you save a design, these *set_rail_voltage* Tcl commands are not saved as part of the SDC constraint of the database, but the result of these commands are stored on the affected cells in the design database and thus are persistent. To remove the rail constraints, use the *remove_rail_voltage* Tcl command.

For the timing analyzer to take into account the manually specified power supply voltages, before you issue the *set_rail_voltage* Tcl command, you must:

- In case of an MCMM design, you must select the scenario for which to set the voltages by invoking the *set_working_scenario* Tcl command with the name of the scenario. You cannot set voltages for multiple scenarios.
- Specify the timing libraries characterized for various voltages. Aprisa will use these libraries to calculate the various timing parameters through interpolation.

  Use the *add_scaling_lib_group* Tcl command to these timing libraries. Then, use the *link_design* Tcl command to rebuild the simulator.
- Disable the automatic backannotation of voltages, calculated during power network analysis.

  Set the *ta* parameter *timing_back_annotate_rail_voltage* to *false*.

Note that, if you set the rail voltages in the top-level design and you merge in a partition, then these rail voltages are propagated in the merged partition.

Syntax

```
set_rail_voltage cells\
   [-max] \
   [-min] \

    -rail_value static_rail_value \
    [-dynamic_rail_value dynamic_rail_value]
```

ARGUMENTS

-rail_value *rvalue*
Sets voltage on single-rail cells. If you use this option, you cannot use the **-rail_list** option: They are mutually exclusive. Replace *rvalue* with a decimal or an integer value.

-rail_list *rname_value_list*
Sets voltages on individual rails of multi-rail cells. Replace *rname_value_list* with a list, which must have even number of elements. (Odd elements are names of rails; even elements are voltage values.) The rail names must match the definitions of rails in the library power_supply section and in the library operating conditions. If you use this option, you cannot use the **-rail_value** option: They are mutually exclusive.

-min
Sets rail voltages for the minimum operating condition. If you do not specify either the **-min** or the **-max** option, the tool assumes you are using both the **-min** and **-max** options.

-max
Sets rail voltages for the maximum operating condition. If you do not specify either the **-min** or the **-max** option, the tool assumes you are using both the **-min** and **-max** options.

cell_list
Specifies a list of cells on which to set rail voltages. Replace *cell_list* with the collection of cells you want.

where the arguments have the following meaning:

cells	List of standard cells, that is, instances of library cells for which to set the rail voltage.
[-max]	The rail voltage applies only to the max PVT condition.
[-min]	The rail voltage applies only to the min PVT condition.
-rail_value *static_rail_value*	Static rail voltage
[-dynamic_rail_value *dynamic_rail_value*]	Dynamic rail voltage.

141

Pl's Ex. 289 (PrimeTime 2012.06) at ATopTech 0140184	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_sense Specifies unateness propagating forward for pins with respect to clock source.	Command: set_sense <db:object_list> standard SDC command
SYNTAX string **set_sense** [-type *type*] [-positive] [-negative] [-stop_propagation] [-logical_stop_propagation] [-pulse *pulse_type*] [-clocks *clock_list*] *object_list*	option: -type sense_type(clock) — specify type of sense (clock or data) to be applied; sense_type = clock | data -stop_propagation — stop propagation of specified signals -positive — only propagate positive unateness from clock source -negative — only propagate negative unateness from clock source -pulse string — pulse type -clocks collection — constraint applied to specified clocks only --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --system_default — use system default values --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
ARGUMENTS -type *type* Specifies whether the type of sense being applied refers to clock networks or data networks. The possible values for the *type* variable are: 'clock', 'data'. Note that 'clock' is assumed to be the default if *-type* option is not given. Note that if *-type data* is given, the **-stop_propagation** and **-clocks** options must be used as well. -clocks *clock_list* Specifies a list of clock objects to be applied with the given unateness that is placed on all pin objects in the *object_list* variable. If the *-clocks* option is not supplied, all clocks passing through the given pin objects are considered. -stop_propagation Stops the propagation of specified clocks in the *clock_list* variable from the specified pins or cell timing arcs in the *object_list* variable. Only clock used as clock is stopped if used with *-type clock*. To stop propagation for both clock as clock and clock as data, you need two commands one with *-type clock* and one with *-type data*. The **-stop_propagation** option cannot be specified with the *-positive*, *-negative* or *-pulse* options.	description: This command is the same as standard SDC command.

142

-positive
Specifies positive unateness applied to all pins in the *object_list* variable with respect to clock source. The *-positive* option cannot be specified with the *-negative* or *-pulse* options.

-negative
Specifies negative unateness applied to all pins in the *object_list* variable with respect to clock source. The *-positive* option cannot be specified with the *-negative* or *-pulse* options.

-pulse *pulse_type*
Specifies the type of pulse clock applied to all pins in the *object_list* variable with respect to clock source. The possible values for the *pulse_type* variable are: 'rise_triggered_high_pulse', 'fall_triggered_high_pulse', 'rise_triggered_low_pulse', or 'fall_triggered_low_pulse'. The *-pulse* option cannot be specified with the *-negative* or *-pulse* options.

object_list
Lists of pins or cell timing arcs with specified unateness to propagate. The timing arcs object can be used with the *-stop_propagation* and *-logical_stop_propagation* options only.

143

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791081-82	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
set_si_delay_analysis Sets coupling information on nets for crosstalk analysis.	Command: set_si_delay_analysis internal development utility
SYNTAX int **set_si_delay_analysis** [-reselect *rnets*] [-ignore_arrival *inets*] [-exclude] [-victims *vnets*] [-aggressors *anets*] [-rise] [-fall] [-min] [-max] list *rnets* list *inets* list *vnets* list *anets*	option: -exclude — exclude nets as victims or aggressors respectively -victim collection — victime nets -aggressor collection — aggressor nets --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --system_default — use system default values --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
ARGUMENTS -reselect *rnets* Specifies a list of nets to be reselected in each iteration, independent of reselection criteria. A net cannot be reselected if it is filtered out; if this is attempted, the XTALK-106 message comes up during the update_timing. You cannot use this option with the **-ignore_arrival**, **-exclude**, **-victims**, or **-aggressors** options. If it applied on a noncoupled net, it is ignored. -ignore_arrival *inets* Specifies a list of nets to be analyzed as infinite window. You cannot use this option with the **-reselect**, **-exclude**, **-victims**, or **-aggressors** options.	description: This command is for AtopTech internal use only.

144

-exclude

Indicates that nets specified as *vnets* or *anets*) are to be excluded from the crosstalk analysis as victim nets or aggressor nets, respectively. You cannot use this option with the **-reselect** or **-ignore_arrival** option. When both **-victims vnets** and **-aggressors anets** are applied, all cross capacitances between **vnets** and **anets** are excluded, when **vnets** are victims and **anets** are aggressors.

-victims *vnets*

Specifies the list of nets on which **-exclude** information is applied as a victim. You cannot use this option with the **-reselect** or **-ignore_arrival** option. If you use the **-victims** option, you must use the **-exclude** option. When used with the **-aggressors** option, **-victims** excludes the cross capacitances between the victim nets (*vnets*) and the aggressor nets (*anets*).

-aggressors *anets*

The list of nets on which **-exclude** option information is applied as an aggressor. You cannot use this option with the **-reselect** or **-ignore_arrival**

option. If you use the **-aggressors** option, you must use the **-exclude** option. When used with the **-victims** option, **-aggressors** excludes the cross capacitances between the victim nets (*vnets*) and the aggressor nets (*anets*).

-rise

Excludes a list of nets for victim rising. If you use the **-rise** option, you must use the **-exclude** option.

-fall

Excludes a list of nets for victim falling. If you use the **-fall** option, you must use the **-exclude** option.

-min

Excludes a list of nets for min path analysis. If you use the **-min** option, you must use the **-exclude** option.

-max

Excludes a list of nets for max path analysis. If you use the **-max** option, you must use the **-exclude** option.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791085-86	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

set_si_noise_analysis

Sets coupling information on nets for noise analysis.

```
Command: set_si_noise_analysis
  exclude or include specified nets for noise analysis.
```

SYNTAX

```
int set_si_noise_analysis
[-ignore_arrival inets]
[-exclude]
[-victims vnets]
[-aggressors anets]
[-above]
[-below]
[-low]
[-high]

list inets
list vnets
list anets
```

```
option:
  -exclude                      exclude nets as victims or
                                aggressors respectively
  -above                        above
  -below                        below
  -high                         high
  -low                          low
  -victim collection            victime nets
  -aggressor collection         aggressor nets
  -ignore_arrival collection    ignore arrival window
  --get_option arg<1>           get option value
  --set_option ...              set option value
  --get_default arg<1>          get default value
  --set_default ...             set default value
  --system_default              use system default values
  --list_options                list current option values
  --load_options ...            load current option values
  --license                     list required licenses
  --help                        display command help
```

ARGUMENTS

-ignore_arrival *inets*

Specifies a list of nets to be set as infinite window. When *inets* are analized as victims, all of their aggressors are set with infinite window. When *inets* are analized as aggressors, they are set as aggressors with arrival time ignored. You cannot use this option with the **-exclude**, **-victims**, **-aggressors** or **-high**, **-low**, **-above**, **-below** options.

-exclude

Indicates that nets specified as *vnets* or *anets*) are to be excluded from the noise analysis as victim nets or aggressor nets, respectively. You cannot use this option with the **-ignore_arrival** option. When both **-victims vnets** and **-aggressors anets** are applied, the noise between **vnets** and **anets** is not analyzed, when **vnets** are victims and **anets** are aggressors.

```
description:
  use -exclude to exclude net from noise analysis
  use -victim or -aggressor to specify victim nets
  or aggressor nets
  use -ignore_arrival to ignore the arrival window
  for specified nets
```

-victims *vnets*
Specifies the list of nets on which **-exclude** information is applied as a victim. You cannot use this option with the **-ignore_arrival** option. If you use the **-victims** option, you must use the **-exclude** option. When used with the **-aggressors** option, **-victims** excludes the noise analysis between the victim nets (*vnets*) and the aggressor nets (*anets*).

-aggressors *anets*
The list of nets on which **-exclude** option information is applied as an aggressor. You cannot use this option with the **-ignore_arrival** option. If you use the **-aggressors** option, you must use the **-exclude** option. When used with the **-victims** option, **-aggressors** excludes the noise analysis between the victim nets (*vnets*) and the aggressor nets (*anets*).

-above
Excludes a list of nets for victim above rail. If you use the **-above** option, you must use the **-exclude** option.

-below
Excludes a list of nets for victim below rail. If you use the **-below** option, you must use the **-exclude** option.

-low
Excludes a list of nets for low rail noise analysis. If you use the **-low** option, you must use the **-exclude** option.

-high
Excludes a list of nets for high rail noise analysis. If you use the **-high** option, you must use the **-exclude** option.

147

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791091-94

set_switching_activity

Sets switching activity annotation on selected nets, pins, ports and cells, etc of the current design.

SYNTAX

```
int set_switching_activity
          [-static_probability value]
          [-toggle_count count]
          [-clock_derate value]
          [-glitch_count count]
          [-state_condition state]
          [-path_sources name_list]
          [-rise_ratio ratio_value]
          [-period period_value]
          [-base_clock clock]
          [-type object_type_list]
          [-no_hierarchy]
          [object_list]
          [-clocks clocks]
          [-quiet]

float value
float count
string state
list name_list
float ratio_value
float period_value
string clock
list object_type_list
list object_list
list clocks
```

Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057498-99

set_switching_activity

Set switching activity (SA) on specified pins, ports, or nets. You can specify the activity in one of the following ways:

- Average number of signal changes per unit of time
- Average number of signal changes per clock period
- Static probability that the signal is high

You can also use this commed to remove all previously set activities.

Syntax

```
set_switching_activity pinsPortsNets \
   { -static_probability probability | \
     -toggle_count count { -period period | -clock string } | \
     -clear }
```

ARGUMENTS

-static_probability *sp_value*
Specifies the value of the **static_probability** switching activity. *sp_value* represents the percentage of the time the signal is at the logic state 1. For example, a *value* of 0.25 indicates that the signal is in the logic state 1 for 25% of the time.

-toggle_count *count*
Specifies the value of the toggle rate switching activity. The count is a floating point number that represent the number of 0-1 and 1-0 glitch free transitions, that the signal makes during a period of time. The period can be specified with the **-period** option. Alternatively, a related clock can be annotated using the **-base_clock** argument, and the specified toggle rate will be relative to the related clock period. If the option -clocks is specified, the related clock is chosen based on which clock domain the object belongs. If belonging to multiple clock domains, the faster clock will be the related clock. If no clock domain is found for the object, choose the fastest clock in the design as the related clock.

-clock_derate *value*
An alternative to -toggle_count if -base_clock or -clocks is specified. The annotated toggle rate for a certain object will be a factor of the toggle rate of the related clock of the object. The related clock is chosen based on which clock domain the object belongs. If belonging to multiple clock domains, the faster clock will be the related clock. If no clock domain is found for the object, choose the fastest clock in the design as the related clock.

-glitch_count *count*
Specifies the value of the glitch rate switching activity. The count is a floating point number that represent the number of 0-1 and 1-0 glitch transitions, that the signal makes during a period of time. The period can be specified with the **-period** option. Alternatively, a related clock can be annotated using the **-base_clock** argument, and the specified glitch rate will be relative to the related clock period. This option cannot be used with -clocks.
Please note that if only one of **-toggle_count** or **-glitch_count** options is specified, then the other value is assumed to be zero.

-state_condition *state*
Specifies the state condition when annotating state-dependent toggle rate and glitch rate on pins or state-dependent static probabilities on cells. State dependent toggle rate and glitch rate can be annotated when the internal power of the library cell pin is characterized with state-dependent power tables. State-dependent static probabilities can be annotated when the cell leakage power is characterized with state-dependent power tables. The state condition specified with this argument must be logically equivalent to a state condition in the internal / leakage power characterization. The state condition should be enclosed between " ". Moreover, if the user wants to set switching activity information for the default state condition then the argument of **-state_condition** option should be "default". This option cannot be used with -clocks.

-path_sources *name_list*
Specifies the path sources when annotating path-dependent toggle rate and glitch rate on pins. This is used when the library cell pin has path-dependent internal power. The path source(s) specified with this argument must be the same as those in the internal power characterization. When listing more than one pin in path sources the pin names should be separated by a space and enclosed between " ". For example if the pins in path sources are A and B, the argument to option **-path_sources** will look like "A B". Please note that if *name_list is given using* ***-path_sources*** argument, then *state* ***should also be provided using -state_condition*** condition. This option cannot be used with -clocks.

where the argument has the following meaning:

pinsPortsNets	Collection of pins, ports and nets on which to set the specified activity.
[-static_probability *probability*]	Static probability that the signal on the specified pins, ports and nets is high.
-toggle_count *count*	Average number of signal changes in a specified time period. Use either the *-period* argument to specify an absolute time or use the *-clock* argument to specify the clock period to use.
[-period *period*]	Time in database time units.
[-clock *string*]	Clock whose period to use for toggle rate.
[-clear]	Remove all activity information on the specified ports, pins, and nets.

149

-rise_ratio *ratio_value*
Specifies the ratio of rise transitions to total transitions for the specified toggle rate and glitch rate when annotating pins that are characterized with both rise and fall internal power. The *ratio_value* argument is a floating point number between 0.0 (all transitions are falling) and 1.0 (all transitions are rising). You need to specify a toggle rate and glitch rate in order to use this option. The default value is 0.5.

clock domain an object belongs to. If belonging to multiple clock domains, the faster clock will be the related clock. If no clock domain is found for the object, choose the fastest clock in the design as the related clock.

-quiet
Specifies quiet mode and for instance suppresses warnings on design objects that could not be annotated on the design.

150

set_user_attribute

Sets a user attribute to a specified value on an object.

SYNTAX

```
string set_user_attribute
[-class class_name]
[-quiet]
object_spec
attr_name
value

stringclass_name
list  object_spec
stringattr_name
stringvalue
```

ARGUMENTS

-class *class_name*
If *object_spec* is a name, this is its class. Allowable values are design, port, cell, pin, net, lib, lib_cell, or lib_pin.

-quiet
Suppresses all report messages.

object_spec
Objects on which to set the attribute. Each element in the list is a collection or a pattern which is combined with the *class_name* to find the objects.

attr_name
Shows the name of the attribute.

value
Shows the value of the attribute.

set_user_attribute

Sets a user-defined attribute on an object. First, you need to define the attribute, the value it can accept, and the class of objects on which it can be placed using the *define_user_attribute* Tcl command.

Syntax

```
set_user_attribute objects att_name att_value [-quiet]
```

where the arguments have the following meaning:

objects	Collection of objects on which this attribute need to be set.
att_name	Name of the attribute that will be set. This attribute must have been created using the *define_user_attribute* Tcl command.
att_value	Value of the attribute. This value must be of the type defined for this attribute.
[-quiet]	Suppress error messages.

151

PI's Ex. 18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057536
`set_variation  # Set variations on timing objects`	**set_variation** Specifies the random variation of delays in library cells for parametric on-chip variation modeling (POCVM) during timing analysis. POCVM can more accurately assess the impact of random delay variation on circuit timing, and thus reduces the pessimism of the traditional on-chip variation timing modeling. Besides the local random variation you set with the *set_variation* Tcl command, you can also use the *read_locv* Tcl command to import a LOCV file that describes distance and stage-dependent variation. You enable POCVM with the *ta* parameter *timing_pocvm_enable_analysis.*

	Syntax
	
	`set_variation \`
	`  -lib_cell lib_cell_list \`
	`  -sigma value \`
	`  [-parameter`
	`  random_variation | random_rise_variation | random_fall_variation]`

`variation_list  (Variation objects to set)` `  [object_list]  (List of timing objects)`	where the arguments have the following meaning `-lib_cell lib_cell_list` — Collection of library cells for which the variation is set. `-sigma value` — The magnitude of the variation. It is the ratio between the standard deviation of the cell delay and the nominal cell delay. This value must be non-negative. `[-parameter random_variation | random_rise_variation | random_fall_variation ]` — Variation is set for rise delay, fall delay, or both: *random_variation*—Variation is set for both rise and fall delays. This is the default value. *random_rise_variation*—Variation is set for the rise delay. *random_fall_variation*—Variation is set for the fall delay.

152

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790974	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057604
sizeof_collection Returns the number of objects in a collection.	**sizeof_collection** Return the number of objects in the specified Aprisa collection. This command has the same functionality as the standard *llength* Tcl command. However, it is much faster and uses less memory because Aprisa collections are handled by the ATopTech database manager directly. For this reason, it is good practice to use Aprisa collections and Aprisa functions to operate on these collections.
SYNTAX int **sizeof_collection** *collection1* collection *collection1*	**Syntax** sizeof_collection *collection* where *collection* is a collection of objects.
sizeof_collection Returns the number of objects in a collection. **SYNTAX** int **sizeof_collection** *collection1* collection *collection1* **ARGUMENTS** collection1 Specifies the collection for which to get the number of objects. If the empty collection (empty string) is used for the *collection1* argument, the command returns 0.	

153

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791121	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057609
sort_collection Sorts a collection based on one or more attributes, resulting in a new, sorted collection. The sort is ascending by default.	**sort_collection** Sorts a list of objects. The sorting is done by comparing the object attributes in the order as listed in the criteria argument. You can sort using both Aprisa attributes and user-defined attributes. By default, the objects are sorted in ascending order. The command also supports dictionary-style sorting, that is, when the names contain numbers, these numbers are sorted numerically.
SYNTAX collection **sort_collection** [-descending] *collection1 criteria* collection *collection1* list *criteria*	**Syntax** sort_collection *objects* {*criteria_list*} \ [-descending] \ [-nocase] \ [-dictionary]
ARGUMENTS -descending Indicates that the collection is to be sorted in reverse order. By default, the sort proceeds in ascending order. collection1 Specifies the collection to be sorted. criteria Specifies a list of one or more application or user-defined attributes to use as sort keys.	where the arguments have the following meaning: *objects* — List of objects to sort. {*criteria_list*} — List of object attributes on which to sort. The attributes are used in the specified order. [-descending] — Sort objects in descending order. [-nocase] — Ignore case for sorting. [-dictionary] — Use dictionary-style comparison

154

swap_cell

Swaps one or more cells with a new design or library cell.

SYNTAX

```
int swap_cell cell_list swap_in

[-dont_preserve_constraints]
[-file file_name]
[-format file_format]
list  cell_list
stringswap_in
stringfile_name
string file_format
```

ARGUMENTS

cell_list
: Specifies a list of cells to be swapped out.

swap_in
: Specifies the name of the design or library cell to be swapped in.

-dont_preserve_constraints
: Indicates that **swap_cell** is not to reapply the current design constraints after the swap.

-file *file_name*
: Specifies the name of a file that contains a design that is to be swapped in.

-format *file_format*
: Specifies the format for *file_name*. Allowed values are *db* (the default), *Verilog*, *EDIF*, and *VHDL*.

swap_cell

Renames and possibly moves a cell in the netlist hierarchy without changing the layout, that is, without changing the cell, its location, or its connectivity.

This operation is mostly used during a metal-only ECO when a placed spare cell, in the logic hierarchy typically located in the top module, is used to make a logic change in some module.

Syntax

```
swap_cell cell \
   -rename name \
```

where the arguments have the following meaning:

cell	Cell, that is, library cell instance or module instance you want to rename.
-rename *name*	New hierarchical name for the cell. If you use the hierarchy separator, this cell effectively moves in the logic hierarchy.

155

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791136	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
update_noise Performs static crosstalk noise analysis for the current design.	Command: update_noise update functional noise

SYNTAX
 	
int **update_noise** [-full]
	option:
	  --license list required licenses
	  --help display command help
	license: AP
	Command: update_rlc_model
	  update all RLC models to ensure data consistency
	option:
	  -rlc_model string only update specified RLC-model
	  --get_option arg<1> get option value
	  --set_option ... set option value
	  --get_default arg<1> get default value
	  --set_default ... set default value
	  --system_default use system default values
	  --list_options list current option values
	  --load_options ... load current option values
	  --license list required licenses
	  --help display command help

ARGUMENTS -full By default, update_noise performs the noise analysis only if the design is not up to date for noise analysis. Using **-full**, forces the update_noise to perform the noise analysis regardless whether the design is out of date or not.	description: This should be the last command in the rlc_model file.

156

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791142	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057743
update_timing Updates timing information on the current design.	**update_timing** Invokes the timing simulator on the design. The simulator can run in regular or incremental mode. A regular analysis ignores all current analysis information. An incremental simulation uses the results of the previous analysis and the changes made to the design since the last analysis. By default, AP knows when it can run in incremental mode and when it needs a full analysis. However, you can force a complete analysis. This command also determines whether a new parasitic extraction is needed, and whether the extraction and timing information of its hard partitions need to be updated as well. For MCMM designs, the timing information is updated for all scenarios, unless you specify a scenario.

SYNTAX
 	
`string update_timing [-full]`
	Syntax
	
	`update_timing [-full] \`
	`  [-scenario scenario_name]`

ARGUMENTS `-full` Indicates that the entire timing analysis is to be performed from the beginning. The default is to perform an incremental analysis, which updates only out-of-date information and runs more quickly.	where the arguments have the following meaning: `[-full]` — Run a complete timing analysis. By default, an incremental timing analysis is performed. `[-scenario scenario_name]` — Name of scenario for which to update the timing information. By default, the timing information is updated for all scenarios. If you specify the *-scenario* argument, you can only update one scenario at a time.

157

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791164	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057808
write_parasitics Writes out annotated parasitics information for the current design.	**write_parasitics** Writes parasitic information to a user-specified file. By default, the parasitic data is written in SPEF format.
SYNTAX Boolean **write_parasitics** -format *file_fmt* *file_name* string *file_fmt* string *file_name*	**Syntax** write_parasitics *file* \ [-use_name_map] \ [-no_coupling_cap] \ [-min] \ [-max] \ [-flat] \ [-format SPEF | DSPF]
ARGUMENTS -format *file_fmt* Specifies the format of the output parasitics file. Currently, the only allowed values are SPEF (Standard Parasitic Exchange Format) and SBPF (Synopsys Binary Parasitics Format). file_name Specifies the name of the output parasitics file.	where the arguments have the following meaning: *file* — Name of the output file with parasitic information. [-use_name_map] — Use a name map when generating an SPEF file. The name map is part of the SPEF standard: it reduces the file size by translating names into numeric IDs. [-no_coupling_cap] — Do not include coupling capacitance [-min] — Write parasitic data for the minimum PVT condition, that is, the condition that results in minimum parasitic resistance and capacitance. [-max] — Write the parasitic data for the maximum PVT condition, that is, the condition that results in maximum parasitic resistance and capacitance. [-flat] — Flatten the netlist, then write parasitic data. [-format SPEF | DSPF] — Format of the file. The default value is *SPEF*.

158

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791175	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057819-20
write_sdc Writes out a script in Synopsys Design Constraints (SDC) format.	**write_sdc** Writes design constraints to a user-specified file using the standard SDC format. SDC constraints typically contain information on the following: • Clocks—Clock definitions, clock latency, clock uncertainty, and clock transition times • Ports—Minimum and maximum arrival times of signals on input ports, and minimum and maximum required arrival times of signals at output ports • Signals—Minimum and maximum transition times, maximum allowed output load, • Paths—Maximum and minimum allowed delay on signal paths, false paths, and multi-cycle paths. These design constraints are used by both construction tools (such as placement/optimization, routing, and clock-tree synthesis) and analysis tools (such as timing analysis and the DRC checker). You can write all SDC constraints or select types of design constraints.
SYNTAX int **write_sdc** *file_name* [-version *sdc_version*] [-compress *compression*] [-include *categories list*] [-nosplit] string*version* string*file_name* string*compression* list *categories list*	**Syntax** write_sdc *file* \ [-port_latency] \ [-port_latency_only] \ [-pin_latency_only] \ [-latency_offset_only] \ [-balanced_source_latency_only] \ [-cancel_out] \ [-extension] \ [-group_path_only] \ [-scenario *scenario_list*]
ARGUMENTS file_name Specifies the name of the file to which the SDC script is to be written. -version *sdc_version* Specifies the version of SDC to write. Allowed values are *1.2*, *1.3*, *1.4*, *1.5*, *1.6* and *latest* (the default). -compress *compression* Specify that the script should be compressed. The only valid value for *compression* is gzip. -include *include_list* Write specified command categories only. The only valid value for *include_list* is **exceptions**. -nosplit The **-nosplit** option prevents line-splitting. This is most useful for doing diff on previous scripts, or for post-processing the script.	where the arguments have the following meaning: *file* — Name of the output file with the SDC constraints. [-port_latency] — Include the clock latency of each module port of the current design. [-port_latency_only] — Only write out the clock latencies of the ports.

159

[-pin_latency_only]	Only write out the clock latencies of the pins.
[-latency_offset_only]	Only write clock latency offsets. The latency of the clock at the various clock pins does not vary much during minor design changes. Instead of recalculating these latencies after every design change, they can be written to a file once and read during successive iterations of the design. Besides cutting down on run-time, designs typically converge faster with this approach.
[-balanced_source_latency_only]	Identify the clocks that have the longest average latency, then add source latency constraints to all other clocks so that their average clock latency matches that of the longest clock. It does so to minimize the timing slack on inter-clock paths.
[-cancel_out]	Set the clock source latency so that it cancels out the average clock network latency. First, the average clock arrival time of all clock sink pins is calculated. Then, the clock source latency is set to the negative value of that average clock network latency. This way, the average clock source latency and the network latency will cancel each other out.
[-extension]	Write all SDC constraints including SDC extensions.
[-group_path_only]	Only write group path constraints.
[-scenario *scenario_list*]	Write seperate SDC files for the listed scenarios. This argument is only applicable in multi-scenario mode. Each SDC file name will have the scenario name as suffix. By default, all constraints of all scenarios in the *current_session* are written.

160

write_sdf

Writes a Standard Delay Format (SDF) back-annotation file.

SYNTAX

```
string write_sdf [-version sdf_version]
[-no_cell_delays]
[-no_timing_checks]
[-no_net_delays]
[-input_port_nets]
[-output_port_nets]
[-significant_digits digits]
[-enabled_arcs_only]
[-no_internal_pins]
[-instance inst_name]
[-context sdf_context]
[-map sdf_map_file_list]
[-annotated]
[-levels level]
[-no_edge]
```


write_sdf

Writes timing data calculated by AP's timing analyzer to a user-specified file using the Standard-Delay Format (SDF, version 3.0) format. This information typically includes:

- Environment and technology conditions for which these results are valid
- Minimum and maximum pin and path delays
- Setup and hold slacks on inputs of memory elements
- Timing constraints
- Skew information

The runtime *ta* parameter, *write_sdf_tmcheck_allow_neg_val*, controls whether negative values are allowed in the TIMINGCHECK section. Note that, this feature is strictly speaking not supported by the SDF 3.0 standard, but it is supported by some third-party tools.

By default, derated delays are reported, but you have the option to get underated delays.

If SI analysis is enabled during timing analysis, the *write_sdf* Tcl command merges both the non-SI net-edge delay and SI-induced delta delay in the interconnect delay value written in the SDF file.

If you only want the non-SI net-edge delay in the interconnect delay value in the SDF file, you must disable SI analysis before using the *update_timing* and *write_sdf* Tcl commands.

You can control the precision with which the results are printed out, and you can filter out disabled arcs and arcs with invalid delays.

Aprisa's SDF writer also supports condition labels and the CONDELSE statement needed for conditional paths.

Note that this command uses the current setting of the *ta* parameter *propagate_clocks*. If this parameter is set to *false*, then, regardless their current values, clock-tree delays and transition times are set to zero in the SDF file.

Syntax

```
write_sdf file \
   [-significant_digits number] \
   [-increment] \
   [-enabled_arcs_only] \
   [-no_derated_delays]
```


161

```
[-compress compression]
[-include include_list]
[-exclude exclude_list]
[-no_negative_delays]
[-no_edge_merging arc_type_list]
file_name

stringsdf_version
int digits
string  inst_name
stringfile_name
stringsdf_context
list sdf_map_file_list
int level
stringcompression
```

ARGUMENTS

-version *sdf_version*
Selects which SDF version to use. Supported SDF versions are 1.0, 2.1, and 3.0. SDF 2.1 is the default.

-no_cell_delays
Indicates that no cell delays are be written in the SDF file. By default, all cell pin-to-pin delays are written to the SDF file. Cell delays include the load delay of the cell. Following the SDF conventions, only cell input pin to cell output pin delays are written. In case one cell output is unbuffered, delays are usually represented in libraries by a delay from an output pin to another output pin. Because this is not allowed by the SDF convention, the SDF delay for an unbuffered output is specified from cell inputs.

-no_timing_checks
Indicates that no cell timing checks are to be written in the SDF file. By default, all cell timing checks (for example, setup, hold, recovery, and removal) are written to the SDF file.

-no_net_delays
Indicates that no net delays are to be written in the SDF file. By default, all net pin-to-pin delays are written to the SDF file.

-input_port_nets
Indicates that the SDF file is to include delays of nets connected to input ports of the current design. By default, these delays are not written to the SDF file because the external connectivity information for ports is not available. If **-instance** is specified, then all net delays across the instance boundary leading to a pin inside the instance are included instead. The pin must be be found on any of levels 1 to *level* of hierarchy if **-levels** *level* is specified.

-output_port_nets
Indicates that the SDF file is to include delays of nets connected to output ports of the current design. By default, these delays are not written to the SDF file because the external connectivity information for ports is not available. If **-instance** is specified, then all net delays across the instance boundary leading from a pin inside the instance are included instead. The pin must be be found on any of levels 1 to *level* of hierarchy if **-levels** *level* is specified.

where the arguments have the following meaning:

file	Name of the SDF file to write.
[-significant_digits *number*]	Number of digits to include after the decimal point. The default value is *3*.
[-increment]	Generates an incremental SDF file.
[-enabled_arcs_only]	Skip disabled arcs and other arcs with invalid delays.
[-no_derated_delays]	Report the non-derated delay values in the SDF file.

NOTE: The following standard arguments to the *write_sdf* Tcl command are not yet supported:

-no_cell_delays, -no_timing_checks, -no_net_delays, -input_port_nets, -output_port_nets, -no_internal_pins, -no_edge, -annotated, -no_negative_delays, -levels, -instance, -context, -version, -compression, -map, -no_edge_merging

162

-significant_digits *digits*
Specifies the number of digits to the right of the decimal point that are to be written in SDF delay triplets. Allowed values are 0-13; the default is 3.

-enabled_arcs_only
Indicates that the SDF file is to contain delays only of enabled timing arcs, and is not to include delays of currently-disabled timing arcs. By default, delays of all timing arcs in the design are written to the SDF file, whether they are disabled or enabled.

-no_internal_pins
Indicates that the SDF file is not to include delay timing arcs from or to internal pins. Timing arcs to or from internal pins are expanded into delays from and to primary input and output of the given cell.

-instance *inst_name*
Specifies that the SDF file is to be written only for the instance named *inst_name*. By default, all pin names are relative to the *inst_name*. However, if boundary net delays are included (**-input_port_nets** or **-output_port_nets**) all pin names are relative to the top design. Note that in general, if **-input_port_nets** or **-output_port_nets** is specified, boundary nets are written leaf-to-leaf and do not start or end on hierarchical pins. If boundary nets are required to start or end on hierarchical pins, refer to the **write_physical_annotations** command.

-context *sdf_context*
Specifies the context for writing bus names in SDF. Valid values are verilog, vhdl, or none (the default). In the verilog context, when pin names are displayed, the last two square bracket characters ("[" and "]") are not escaped. In the vhdl context, the last two parenthesis characters ("(" and ")") in a pin name are not escaped. In the default context none, all bus-delimiting characters are escaped with a backslash character ("always escaped. When used with the **-map** option, **-context** also affects the way names are printed in mapped SDF files. In the verilog context, names are printed in %s[%d] format; in the vhdl context, names are printed in %s(%d) format.
Note: Names are affected only if they are mapped using the bus(name_to_be_changed) function in the mapping file.

-map *sdf_map_file_list*
Specifies a list of mapping files the SDF writer is to use when writing out the SDF file. A mapping file contains a user-specified format for printing SDF cell delays and constraints. When writing out SDF for a cell, the SDF writer takes the user-specified mapping, if present, to print out SDF for the cell. If no user-specified mapping is present for a cell, the SDF writer writes out SDF in the normal way.

-annotated
Indicates that the SDF is to include only timing arcs that have been annotated with the **read_sdf**, **set_annotated_delay**, or **set_annotated_check** commands.

-levels *level*
Specifies the number of levels of hierarchy for which the SDF is written out. Level 1 means only the top design or *inst_name*. Value of N means all levels of hierarchy, 1 to N. By default, all levels of hierarchy are written out. Note that boundary net delays (**-input_port_nets**, **-output_port_nets**) typically have some net arcs from or to pins outside the *inst_name*. The location of such outside pins is not limited by **-levels**. That is, the **-levels** and **-instance** options let you choose which boundary arcs are included, but do not restrict where the arcs lead outside of *inst_name*.

-no_edge
Indicates that the generated SDF is not to include any edges (posedge or negedge) for both combinational and sequential IOPATHs.

163

file_name
Specifies the name of the SDF file to be written.

-compress *compression*
Specifies a format to be used to compress the file. The only valid value for *compression* is gzip. By default, files are not compressed.

-include *include_list*
Specifies a list of constructs to include in the SDF file; these replace one or more constructs from the set of default constructs. Allowed values are one or more of the following:

• SETUPHOLD, which indicates that all SETUP and HOLD constructs are to be replaced by SETUPHOLD constructs. If a pair of setup and hold arcs are found between the same pin edges, timing information for the/both arc/arcs is written in a single SETUPHOLD construct. If a single setup/hold arc is found then the arc will be written in a single SETUPHOLD construct with no timing information for the hold/setup portion. SETUPHOLD supports negative values and can be written only for versions 2.1 and 3.0.

• RECREM, which indicates that all RECOVERY and REMOVAL constructs are to be replaced by RECREM constructs. If a pair of recovery and removal arcs are found between the same pin edges, timing information for both arcs is written in a single RECREM construct.If a single recovery/removal arc is found then the arc will be written in a single RECREM construct with no timing information for the removal/recovery portion. RECREM supports negative values and can be written only for version 3.0.

-exclude *exclude_list*
Specifies a list of timing values of construct types to be either excluded from the SDF file in order to reduce its size, or to be replaced by another construct, as in the case of condelse. Allowed values are one or more of the following:

• constant_nets, which indicates that nets are to be omitted from the SDF file if they propagate a constant.

• constant_delay_arcs, which indicates that delay arcs are to be omitted from the SDF file if they propagate a constant, either from case analysis or logical inputs.

• default_cell_delay_arcs, which indicates that all default cell delay arcs are to be omitted from the SDF file if conditional delay arcs are present. If there are no conditional delay arcs, the default cell delay arcs are written to the SDF file.

• wlm_load_delay, which indicates that net delays and cell delays calculated using WLM are to be ommitted from the SDF.

• checkpins, when library compiler finds both combinational and sequential arcs between pins, a checkpin is created so that all arcs are expanded in the db so that a single arc pinA-pinB is replaced by the combination of a positive unate arc pinA-pinAcheckpin1 with zero delay and an arc pinAcheckpin1-pinB with the same sense and values as the original arc. When this option is set the SDF is written out as if all checkpins were never created.

• no_condelse, indicates that PrimeTime will not use the condelse statement to write out default iopaths. By default PrimeTime will replace default iopaths with the condelse construct. Specifying this option will result the condelse statement being replaced by a default iopath. This option should be used for generating simulator compatible SDF.

-no_negative_delays
Specifies that PrimeTime will generate an sdf file without negative delays. Any delay values which are negative prior to writing the sdf file will be

164

represented as a zero in the sdf file. This option should be used when the sdf file is intended for simulator use. Using this option leads to inaccruate delay estimation in PrimeTime, so the user should use caution with this option.

-no_edge_merging
Specifies a list of arc types which are not to be compressed in the SDF file through edge merging. Allowed values are one or more of the following

165

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791189-92	Pl's Ex. 266 (AP 13.11.rel.4) at ATopTech 0057836-37
write_spice_deck Writes to a SPICE deck the paths or arcs generated by **get_timing_paths** or **get_timing_arcs**.	**write_spice** Generates a netlist in Spice format of a set of paths in a clock tree. You specify the paths to include by specifying the source pins and sink pins. By default, the order of pins of all cells is assumed to be the order of those pins in the LEF library. However, if you provide a list of Spice subcircuits, then the pins are ordered according to the Spice subcircuit definitions. Sources pins can be: • ports • clock meshes • ECK pins of ICG cells
SYNTAX int **write_spice_deck** [-align_aggressors] [-analysis_type *type*] [-c_effective_load] [-full_clock_cone] [-ground_coupling_capacitors] [-header *header_file_name*] [-initial_delay *delay*] [-logic_one_name *v1name*] [-logic_one_voltage *v1*] [-logic_zero_name *v0name*] [-logic_zero_voltage *v0*] [-margin *margin_value*] [-minimum_transition_time *trans*] [-no_clock_tree] [-output *file_name*] [-pre_driver] [-sub_circuit_file *spice_sub_circuit_file*] [-sweep_size *number_of_points*] [-sweep_step *num*] [-time_precision *precision*] [-transient_size *tran_size*] [-transient_step *tran_step*] [-use_probe] [-user_measures *user_measure_list*] [-sample_size *number_of_samples*] *paths_arcs_list*	**Syntax** write_spice \ -from_pins *fromPins* \ -to_pins *toPins* \ [-subcircuit *sub_circuit_file*] \ *output_file*

166

```
stringheader_file_name
float delay
stringv1name
float v1
stringv0name
float v0
float margin_value
float trans
stringpaths
stringfile_name
stringspice_sub_circuit_file
unsignednumber_of_points
float num
unsignedprecision
float tran_size
float tran_step
int     number_of_samples
```

ARGUMENTS

-align_aggressors
Apply only to a **net** timing arc. Indicates that the relative switching time of the active aggressors of the net arc compute by the cross talk delay or noise bump are written out the corresponding PWL statement. It is effective if the net has a coupled RC network annotated. The victim will switch after the initial_delay and the active aggressors will switch relative to that. Spice uses the calculation engine to get the worst case alignment, and uses simillar setup so that alignment stays valid. i.e. the filtered aggressors are not considered as effective during calculation so they are coupling capacitance is grounded.

-analysis_type *type*
Specifies the type of cross talk/noise analysis for the spice deck generated. The possible crosstalk delay types are *max_rise*, *max_fall*, *min_rise* and *min_fall*. The possible noise types are *above_high*, *above_low*, *below_high* and *below_low*. This option has no effect on the timing path. The default value is *max_rise* for a timing arc.

-c_effective_load
Indicates that the effective capacitors computed by the PrimeTime during the delay calculation are connected to some driver pins in the SPICE deck. These driver pins are not driving any victim nets and aggressor nets.

-full_clock_cone
Indicates that the full fan-in cone of the clock tree is to be generated. By default, if the clock is propagated, a single chain of clock tree gates is generated; otherwise, a piecewise linear waveform (PWL) is connected to the clock pin. Note that using this option could generate a very large spice deck, because **write_spice_deck** must determine all voltage levels or waveforms of every input port of these input cones. An error message is issued if -no_clock_tree is also set.

where the arguments have the following meaning:

-from_pins *fromPins*	Start pin for which to generate a netlist in Spice format.
-to_pins *toPins*	End pin for which to generate a netlist in Spice format.
[-subcircuit *sub_circuit_file*]	If specified, the pins of the cells are ordered according to the port positions in the *sub_circuit_file*. Otherwise, they are ordered based on port positions in the LEF file. *sub_circuit_file* must be in spice format.
output_file	Name of the file to create.

167

-ground_coupling_capacitors
Indicates that the aggressors of the timing path or timing arc are not written. The associated coupling capacitors are grounded with the factor one.

-header *header_file_name*
Specifies the path to the user header file whose content is copied to the spice deck generated. User can use this file to identify the spice deck, to include the library file(s), or to copy text to spice deck for any other purposes to facilitate the spice run.

-initial_delay *delay*
Specifies the initial delay, in library unit, added to all PWL statements. The default value is the longest clock period, or 1.0 library unit for asynchronous designs. Note that setting *delay* to zero makes generating a ramp difficult and is not recommended.

-logic_one_name *v1name*
Specifies name of the default upper rail voltage source.

-logic_one_voltage *v1*
Specifies the upper rail of the voltage swing of the gate input pins. This

168

is used in the PWL and power rail vdd generated by the command. The default value is main library voltage. This option will be effective only if the variable library_thresholds_use_main_lib is set to TRUE.

-logic_zero_name *v0name*
Specifies name of the default lower rail voltage source.

-logic_zero_voltage *v0*
Specifies the lower rail of voltage swing of the gate input pins. This is used in the PWL and the ground voltage vss generated by the command. The default value is 0 volts. This option will be effective only if the variable library_thresholds_use_main_lib is set to TRUE.

-margin *margin_value*
Specifies the value in time to be reduced from the switching time of the data pin of the lunching sequential cell of the timing path or timing arc.

-minimum_transition_time *trans*
Specifies the minimum transition time in nanoseconds, to be used in all generated PWL if the transition time computed by PrimeTime is smaller than *trans*. The default value is 0.001 ns; transition times less than 0.0001 ns are not recommended.

-no_clock_tree
Indicates no clock path is traced. A clock pulse statement is connected directly to the clock pin of a sequential gate. An error is issued if the -full_clock_cone is set. If the delay type of the timing path is max (max_rise or max_fall) the pulse statement of the lunching clock is computed from the late edges of the clock arrival windows and the maximum slew of the clock pin. The pulse statement of the capturing clock is computed from the early edge of the clock arrival windows and the minimum slew. For the min (min_rise or min_fall) delay type, the early edges are used for the lunching clock and the late edges are used for the capturing clock.

-output *name*
If **-sample_size** option is not used, this option specifies the name of the SPICE deck file to be written for the first timing path. SPICE deck files related to subsequent timing paths are also based on this name. This is required. If the **-sample_size** option is used, then this option specifies the name of the directory to be created for writing the sampled spice deck files.

-pre_driver
The PWL voltage sources are replced by the equivelent synopsys pre-driver. The pre-driver is a smooth wavefrom which is more realastic than the ramp. Use this option only if the library is characterized by the standard synopsys pre-driver.

-sub_circuit_file *spice_sub_circuit_file*
Specifies the path to the file that contains all the SPICE .subckt definitions of all gates in the timing paths. By default, a subcircuit call uses the pin order in the Synopsys .lib file. Use this option if the SPICE subcircuit has a different pin order from that of the .lib file.

-sweep_size *number_of_points*
Used in conjunction with the -align_aggressors option. Indicates the number

169

of sweep point generated for each active aggressors of the net arc. The number of simulation will increase geometrically with the number of the active aggressor

-sweep_step *num*
Used in conjunction with the -align_aggressors option and -sweep_size. Indicates the maximum time interval between sweep points generated for each active aggressors of the net arc. The unit is in nano second. The default is 0.1ns.

-time_precision *precision*
Specifies the number of precision digits for time in the PWL generated. The default value is 6. The range is from 1 to 20.

-transient_size *tran_size*
Specifies the total transient time used in the SPICE .tran statement. The unit is in the largest clock period. The default is 4 clock periods. If there is no clock in the design, 10ns is used.

-transient_step *tran_size*
Specifies the transient step size used in the SPICE .tran statement. The unit is in nano second. The default is 0.001ns.

-use_probe
Use .probe statement to output the node voltage instead of .print statement.

-user_measures *user_measure_list*
Use this option to add you'r own measures instead of the ones generated by the spice deck automatically. The empty *user_measure_list* could be used as a way to remove all auto generated .measure and .print from spice deck.

-sample_size
Specifies the number of spice deck files that have to be created while performing variation-aware timing analysis. This option takes the name of the directory via **-output** option and creates multiple spice deck files that correspond to various samples of the variations defined.

paths_arcs_list
Specifies the collection of timing paths or timing arcs that their circuits are written out.

170

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793655

case_analysis_propagate_through_icg

Determines whether case analysis is propagated through integrated clock gating cells.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *false* (the default), constants propagating throughout the design will stop propagating when an integrated clock gating cell is encountered. Regardless of whether the integrated clock gating cell is enabled or disabled, no logic values will propagate in the fanout of the cell.

When *true*, constants propagated throughout the design will propagate through an integrated clock gating cell provided the cell is enabled. An integrated clock gating cell is enabled when its enable pin (or test enable pin) is set to a hi logic value. If the cell is disabled, then the disable logic value for the cell is propagated in its fanout. e.g. for a latch_posedge ICG, when it is disabled, it will propagate a logic 0 in its fanout.

Since all latch based integrated clock gating cells are sequential in nature, these cells will only be considered for logic propagation if the *case_analysis_sequential_propagation* variable is set to *always*

To activate logic propagation through all integrated clock gating cells, the user must set the following prior to performing an update_timing.

set case_analysis_sequential_propagation always set case_analysis_propagate_through_icg true

To determine the current value of this variable, type **printvar case_analysis_propagate_through_icg** or **echo $case_analysis_propagate_through_icg**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018474

case_analysis_propagate_through_icg

Persistent parameter that controls whether case analysis propagation should be enabled through ICG cells. This parameter is only relevant if the value of the *ta* parameter, *case_analysis_sequential_propagation* is *ALWAYS*.

Syntax:

```
case_analysis_propagate_through_icg true | false
```

where the values have the following meaning:

`true`	Enable case analysis propagation through ICG cells.
`false`	Do not perform case analysis propagation through ICG cells. This is the default value.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793656

case_analysis_sequential_propagation

Determines whether case analysis is propagated across sequential cells.

TYPE

fIstringfP

DEFAULT

never

DESCRIPTION

Determines whether case analysis is propagated across sequential cells. Allowed values are *never* (the default) or *always*. When set to *never*, case analysis is not propagated across the sequential cells. When set to *always*, case analysis is propagated across the sequential cells.

To determine the current value of this variable, type **printvar case_analysis_sequential_propagation** or **echo $case_analysis_sequential_propagation**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018475

case_analysis_sequential_propagation

Persistent parameter that controls whether Aprisa propagates constants specified for a case analysis across sequential cells.

Syntax:

```
case_analysis_sequential_propagation never | always
```

where the values have the following meaning:

`never`	Do not propagate conditions on conditional arcs across sequential cells.
`always`	Propagate conditions on conditional arcs across sequential cells.

The default value is *never*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793661

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

This variable sets the maximum number of objects that can be displayed by any command that displays a collection. The default is 100.

When a command (for example, **add_to_collection**) is issued at the command prompt, its result is implicitly queried, as though **query_objects** had been called. You can limit the number of objects displayed by setting this variable to an appropriate integer. 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, use **printvar collection_result_display_limit**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018393

collection_result_display_limit

Runtime parameter that specifies for commands returning the maximum number of objects in a collection to display on the screen and in the command log.

Syntax:

`collection_result_display_limit` *integer*

where *integer* is the maximum number of objects to display on the screen and to write in the log file.

The default value is *100*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793664

default_oc_per_lib

Enables the use of a default operating condition per individual library.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

Enables the use of a default operating condition per individual library. When the **default_oc_per_lib** variable is set to *true* (the default value), each cell that does not have an explicitly-set operating condition (on the cell itself, on any of its parent cells, or on the design) is assigned the default operating condition of the library to which the cell belongs. When set to *false* all cells that do not have any explicitly-set operating condition are assigned the default operating condition of the main library (the first library in the link_path).

The recommended flow is to explicitly set operating conditions on the design or on each hierarchical block that is powered by the same voltage (also called the voltage island). This variable is mainly for obtaining backward compatibility for the corner case of use of default conditions in releases prior to 2002.09.

To determine the current value of this variable, use **printvar default_oc_per_lib**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018480

default_oc_per_lib

Persistent parameter that controls whether Aprisa uses for each library the default operating condition as set in the library as opposed to using one operating condition for all cells of all libraries.

Syntax:

```
default_oc_per_lib true | false
```

where the values have the following meaning:

true	Use the default operating condition as specified in a library for all instances from cells in that library.
false	Use the same default operating condition for all instances used in the design. The default operating condition is the operating condition from the first library listed in the link path.

The default value is *false*.

Pl's Ex. 290 (PrimeTime G-2012.06) at ATopTech 0140383

delay_calc_waveform_analysis_mode

Controls usage of CCS-based waveform analysis for uncoupled and signal integrity calculation.

TYPE

string

DEFAULT

disabled

DESCRIPTION

You can set this variable to *disabled* , *clock_network*, or *full_design*. When set to *clock_network* or *full_design*, it turns on CCS-based waveform analysis and applies it to the clock network or the entire design. By default, this variable is set to *disabled*. This feature requires CCS data, so libraries need to contain CCS timing data, CCS noise data, or both.

A PrimeTime SI license is required for this feature, including uncoupled calculation. However, you do not need to set the **si_enable_analysis** variable for uncoupled analysis.

For complete information about the CCS-based waveform analysis feature, see the PrimeTime SI User Guide.

The delay_calc_enable_waveform_analysis variable has been deprecated and will be obsolete in a future release.

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

printvar delay_calc_waveform_analysis_mode

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018482-83

delay_calc_waveform_analysis_mode

Persistent parameter that enables a more accurate delay calculation mode that takes into account the shape of the signals. You can enable this mode for clocks or for clocks and signals, and you can select whether this should be done for all delays or only for the delays calculated using the Arnoldi model.

Syntax:

```
delay_calc_waveform_analysis_mode disabled | clock_path | full_design |
          clock_path_ccs_arnoldi | full_design_ccs_arnoldi
```

where the values have the following meaning:

disabled	Do not perform waveform analysis.
clock_path	Apply waveform analysis on all clock paths.
full_design	Apply waveform analysis on the entire design, that is, on all clock nets and all signal nets.
clock_path_ccs_arnoldi	Apply waveform analysis only on nets of clock paths for which the Arnoldi delay calculation is used.
full_design_ccs_arnoldi	Apply waveform analysis only on signal and clock nets of clock paths for which the Arnoldi delay calculation is used.

The default value is *dsiabled.*

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793666

disable_case_analysis

Specifies whether case analysis is disabled.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *false* (the default), constant propagation is performed in the design from pins either that are tied to a logic constant value, or for which a **case_analysis** command is specified. For example, a typical design has several pins set to a constant logic value. By default, this constant value propagates through the logic to which it connects. When the variable **disable_case_analysis** is *true*, case analysis and constant propagation are not performed.

To determine the current value of this variable, use **printvar disable_case_analysis**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018484

disable_case_analysis

Persistent parameter that controls whether the timing analyzer works in case analysis mode. In this mode, you specify specific constant values and signal transitions to selected inputs to force the circuit in a certain state.

Syntax:

```
disable_case_analysis true | false
```

where the values have the following meaning:

`true`	Ignore all case-analysis settings.
`false`	Perform a case analysis using all case-analysis settings.

The default value is *false*.

Pl's Ex. 36 (PrimeTime D-2010.06) at SYNPS-00009997

power_analysis_mode

Sets the power analysis mode

TYPE

fIstringfP

DEFAULT

averaged

GROUP

power_variables

DESCRIPTION

Use this variable to explicitly select the analysis mode for power calculation. Today PrimeTime PX provides three different analysis modes: averaged, time_based and leakage_variation. This variable needs to be set before the first power command, otherwise, the default mode will be assumed. For a particular analysis mode, appropriate activity information must be provided. Allowed values are as follows:

* *averaged* (the default): PrimeTime PX will calculate power based on toggle-rate and state-probability. Only averaged power results will be calculated. In this mode, it can take VCD file, SAIF file as activity input files. Commands **set_switching_activity** and **set_case_analysis** can also be used to set the statistical switching activity on top of the default switching activity. For more information, please check out the man page for *update_power*.

* *time_based*: PrimeTime PX will calculate power based on the events from VCD. Not only averaged power results will be calculated, but also peak powers and time_based power waveforms will be calculated. VCD file must be provided in this mode. Both gate-level VCD and RTL VCD can be specified for this mode. For more information, please check out the man page for *update_power*.

* *leakage_variation*: PrimeTime PX will perform leakage variation analysis. For more information, please check out the man page for **power_enable_leakage_variation_analysis**.

Variable **power_analysis_options** can be changed in one run. However, once changed, all the activity and power data will be removed internally. User needs to provide activity information again before **update_power**.

In addition, command **set_power_analysis_options** can be used to specify the options for power analysis.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta power_analysis_mode 1
#       Type    : int
#       Usage   : 0: PTPX correlated, 1: RedHawk correlated
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793733	Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018803
power_default_static_probability Specifies the default static probability value. power_default_toggle_rate Specifies the default toggle rate value. **TYPE** Float **DEFAULT** 0.5	**power_default_static_probability** Persistent parameter that specifies the default static probability factor. **Syntax:** power_default_static_probability *static_prob_factor* where *static_prob_factor* is a real number ranging between *0.000* and *1.000* indicating the default static probability factor. The default value is *0.500*.

power_default_toggle_rate

Specifies the default toggle rate value.

TYPE

Float

DEFAULT

0.1

DESCRIPTION

The power_default_toggle variable and the power_default_toggle_rate_reference_clock and power_default_static_probability variables are used to determine the switching activity of non-user annotated nets that are driven by primary inputs or black-box cells.

For other unannoated nets, PrimeTime PX will propagate the switching activities of the driving cell inputs based on the cell functionality to derive the switching activity required for power calculations. This mechanism cannot be used for primary inputs and black-box outputs. Instead the following values are used for these type of nets:

- User annotated values are used.

- In some cases, unannotated switching activity values may still be accurately derived, for example, if the net drives a buffer cell and the output of this cell is user annotated, then the user annotated values are used as the default values. Also, if the input is a clock then the clock period and waveform are used to derive the switching activity values.

- If the static probability is not annotated then the value of the power_default_static_probability variable is used for the static probability value.

- If the toggle rate is not user annotated, no mater the static probability is set or not, the following is used for the toggle rate value:

dtr * fclk

where fclk is the frequency of the related clock, and dtr is the value of the power_default_toggle_rate variable.

The related clock is determined by the value of power_default_toggle_rate_reference_clock. Two values (fastest, related) are allowed for the value of power_default_toggle_rate_reference_clock variable. If the value fastest is specified, the related clock would be simply the fastest clock in the design. If the value related is given, the related clock would depend on which clock domain the net belongs to.

The value of power_default_static_probability should be between 0.0 and 1.0, both inclusive. The value of power_default_toggle_rate should be greater or equal to 0.0. Also, if the value of power_default_static_probability is 0.0 or 1.0, then the value of power_default_toggle_rate should be 0.0. If the value of power_default_toggle_rate is 0.0, then the value of power_default_static_probability should be either 0.0 or 1.0.

The default value of power_default_static_probability is 0.5 and the default value of power_default_toggle_rate is 0.1.

power_default_toggle_rate

Persistent parameter that sets the default toggle rate for all signal net, except for clock nets.

Syntax:

`power_default_toggle_rate` *`power_toggle_rate`*

where *`power_toggle_rate`* is a value ranging from *0.00* to *1.00*, indicating the number of times a signal is assumed to change during one clock period. This value is used if, for a net, no other activity information is available.

The default value is *0.100*, indicating that a signal changes every 10 clock periods. The default toggle rate for every clock net is *2.0*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793770-71

rc_degrade_min_slew_when_rd_less_than_rnet

Enables or disables the use of slew degradation in min analysis mode during the RC-009 condition.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When false (the default), PrimeTime does not use slew degradation through RC networks in min analysis mode during the RC-009 condition. When true, PrimeTime uses slew degradation during the RC-009 condition. This variable is effective only if the **rc_adjust_rd_when_less_than_rnet** variable is true.

The "RC-009 condition" means a condition in which PrimeTime checks the library-derived drive resistance, and if it is less than the dynamic RC network impedance to ground by an amount equal to or greater than the value of the **rc_rd_less_than_rnet_threshold** variable, PrimeTime adjusts the drive resistance using an empirical formula to improve accuracy, and issues the RC-009 message. In case this improved accuracy is not sufficient, PrimeTime provides extra pessimism by not using slew degradation in min analysis mode; however, superfluous min delay violations could occur as a side effect. You can keep slew degradation on in min analysis mode after you have qualified the RC-009 methodology for your accuracy requirements, by setting this variable to true.

rc_degrade_min_slew_when_rd_less_than_rnet is one of a set of four variables relevant to the RC-009 condition. The other three are as follows:

- **rc_adjust_rd_when_less_than_rnet** enables or disables the RC-009 condition; the default is true. When this variable is set to false, PrimeTime does not check the drive resistance, and the values of the other related variables do not matter.

- **rc_filter_rd_less_than_rnet** determines whether the RC-009 message is issued only when a network delay is greater than the corresponding driver transition time. The default is true. To receive RC-009 messages every time PrimeTime overrides the drive resistance, set this variable to false. This variable has no effect if **rc_adjust_rd_when_less_than_rnet** is false.

- **rc_rd_less_than_rnet_threshold** specifies the threshold beyond which PrimeTime overrides the library-derived drive resistance with an empirical formula. The default is 0.45 ohms. You can override this default by setting the variable to another value. This variable has no effect if **rc_adjust_rd_when_less_than_rnet** is false.

Note: If **rc_degrade_slew_when_rd_less_than_rnet** is false while **rc_filter_rd_less_than_rnet** is true, the RC-009 message is not issued.

For more information, see the manual page of the RC-009 warning message.

To determine the current value of this variable, type **printvar rc_degrade_min_slew_when_rd_less_than_rnet** or **echo $rc_degrade_min_slew_when_rd_less_than_rnet**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018511

rc_degrade_min_slew_when_rd_less_than_rnet

Persistent parameter that controls how the delay is calculated for interconnects with high resistance.

By default, for high-resistive nets, AP uses a different approach for calculating interconnect delays compared to the Arnoldi method used for regular nets.

This is done to ensure sufficient pessimism. You can force Aprisa to use the same approach as for regular nets, resulting in a more accurate, but not guaranteed worst-case result. You may want to use this parameter to avoid false hold time violations.

Syntax:

```
rc_degrade_min_slew_when_rd_less_than_rnet true | false
```

where the values have the following meaning:

`true`	Revert to the default interconnect delay calculation if the algorithm for high-resistive nets is pessimistic.
`false`	Always use the special algorithm for high-resistive nets.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793774

rc_driver_model_max_error_pct

Specifies the maximum error tolerated in a driver model used in RC effective-capacitance calculations.

TYPE

double

DEFAULT

16.0

DESCRIPTION

When a driving cell is connected to a network with annotated parasitics, an effective capacitance is computed for use with lumped-load based library data. For a candidate value of effective capacitance, PrimeTime builds a driver model that matches the library's delay, slew, and sensitivity to output capacitance. Since the PrimeTime driver model is an abstract approximation of the actual transistor behavior, it is usually impossible to match these criteria exactly. Some error tolerance must be used.

The total allowable error tolerance over all criteria can be specified with **rc_driver_model_max_error_pct**. PrimeTime tries initially to build a driver model with 1% error and gradually relaxes that goal to the value of **rc_driver_model_max_error_pct**.

The value of **rc_driver_model_max_error_pct** is ignored if it is set to a value less than 1%. The value is specified as a percentage, so a desired error tolerance of 10% should be specified as 10.0.

You can use the **report_driver_model** command to examine the driver model error, library data, and matching criteria for a given operating point.

If the RC effective-capacitance calculation fails because the driver model error is greater than **rc_driver_model_max_error_pct**, the warning message RC-004 (single drive) or RC-007 (multidrive) will be issued.

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

pt_shell **printvar rc_driver_model_max_error_pct**

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta rc_driver_model_max_error_pct 1.000 ; # expert
#      Type    : float
#      Usage   : maximum error tolerance in effective
                 capacitance during report_driver_model
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793775

rc_driver_model_mode

Specifies which driver model type to use for RC delay calculation.

TYPE

string

DEFAULT

advanced

DESCRIPTION

PrimeTime supports two types of driver models for RC delay calculation, basic and advanced. The basic model is derived from the conventional delay and slew library schema, while the advanced model is derived from a new schema. The advanced model has many advantages, one of which is the solution to the problem described by the RC-009 warning message. The advanced driver model is part of the Synopsys Composite Current-Source (CCS) model.

When the shell variable **rc_driver_model_mode** is set to *basic*, RC delay calculation will always use driver models derived from the conventional delay and slew schema present in design libraries. When set to *advanced*, RC delay calculation will use the advanced driver model if data for it is present. The **report_delay_calculation** command used on a cell arc will show the message "Advanced driver-modeling used" as appropriate.

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

pt_shell **printvar rc_driver_model_mode**

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018511-12

rc_driver_model_mode

Persistent parameter that controls whether the basic or the advanced CCS model is used for output pins driving nets.

Syntax:

`rc_driver_model_mode basic | advanced | auto`

where the values have the following meaning:

`basic`	Use the simplified CCS model on selected net drivers.
`advanced`	Use the advanced CCS model on all net drivers.
`auto`	Automatically select between the basic and the advanced model, depending on the criticality.

The default value is *auto*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793788

rc_receiver_model_mode

Specifies which receiver model type to use for RC delay calculation.

TYPE

string

DEFAULT

advanced

DESCRIPTION

PrimeTime supports two types of receiver models for RC delay calculation, basic and advanced. The basic model is a single capacitance dependent only on the rise, fall, min, or max arc condition. The advanced model is a voltage-dependent capacitance additionally dependent on input-slew and output capacitance. The advanced model has many advantages, one of which is that the accuracy of **both** delays *and* slews is improved. Another advantage is that nonlinearities such as the Miller effect are addressed. The advanced receiver model is part of the Synopsys Composite Current-Source (CCS) model.

The advanced receiver model will only be used when the network is driven by the advanced driver model; please see the man page on the **rc_driver_model_mode** shell variable for more information.

When the shell variable **rc_receiver_model_mode** is set to *basic*, RC delay calculation will always use the pin capacitances specified in the design libraries. When set to *advanced*, RC delay calculation will use the advanced receiver model if data for it is present *and* if the network is driven by the advanced driver model. The **report_delay_calculation** command used on a network arc will show the message "Advanced receiver-modeling used" as appropriate.

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

pt_shell **printvar rc_receiver_model_mode**

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018512

rc_receiver_model_mode

Persistent parameter that controls whether the basic or the advanced CCS model is used for input pins driven by nets.l

Syntax:

`rc_receiver_model_mode basic | advanced | auto`

where the values have the following meaning:

`basic`	Use the simplified CCS model for selected net receivers.
`advanced`	Use the advanced CCS model for all net receivers.
`auto`	Automatically select between the basic and the advanced model, depending on the criticality.

The default value is *auto*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793791

rc_slew_lower_threshold_pct_fall

Specifies the percentage threshold of the lower slew endpoint to the voltage source for a falling transition.

TYPE

float

DEFAULT

20

DESCRIPTION

Specifies the threshold voltage that defines the endpoint of the falling slew calculation. The value is a percent of the voltage source. Allowed values are 0.0 - 100.0 inclusive; the default is 20.0.

This variable is one of 8 variables, listed in Table 1, that you must specify in order to perform delay calculation in the presence of annotated parasitics using the command **read_parasitics**. These variables interpret the cell delays and transition times from the Synopsys library.

Table 1

Variable Name	Default
rc_slew_lower_threshold_pct_rise	20.0
rc_slew_lower_threshold_pct_fall	20.0
rc_slew_upper_threshold_pct_rise	80.0
rc_slew_upper_threshold_pct_fall	80.0
rc_input_threshold_pct_fall	50.0
rc_input_threshold_pct_rise	50.0
rc_output_threshold_pct_fall	50.0
rc_output_threshold_pct_rise	50.0

The default values specify that a cell delay is defined from 50% of the voltage value for the input transition to 50% of the voltage value for the output transition. The default values also specify that a transition time, or slew, is defined from 20% to 80% of the voltage.

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

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta rc_slew_lower_threshold_pct_fall 0.000 ; # expert
#       Type    : float
#       Usage   : slew lower threshold for fall
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793793

rc_slew_lower_threshold_pct_rise

Specifies the percentage threshold of the lower slew startpoint to the voltage source for a rising transition.

TYPE

float

DEFAULT

20

DESCRIPTION

Specifies the threshold voltage that defines the startpoint of the rising slew calculation. The value is a percent of the voltage source. Allowed values are 0.0 - 100.0 inclusive; the default is 20.0.

This variable is one of 8 variables, listed in Table 1, that you must specify in order to perform delay calculation in the presence of annotated parasitics using the command **read_parasitics**. These variables interpret the cell delays and transition times from the Synopsys library.

Table 1

Variable Name	Default
rc_slew_lower_threshold_pct_rise	20.0
rc_slew_lower_threshold_pct_fall	20.0
rc_slew_upper_threshold_pct_rise	80.0
rc_slew_upper_threshold_pct_fall	80.0
rc_input_threshold_pct_fall	50.0
rc_input_threshold_pct_rise	50.0
rc_output_threshold_pct_fall	50.0
rc_output_threshold_pct_rise	50.0

The default values specify that a cell delay is defined from 50% of the voltage value for the input transition to 50% of the voltage value for the output transition. The default values also specify that a transition time, or slew, is defined from 20% to 80% of the voltage.

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

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta rc_slew_lower_threshold_pct_rise 0.000 ; # expert
#       Type    : float
#       Usage   : slew lower threshold for rise
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793795

rc_slew_upper_threshold_pct_fall

Specifies the percentage threshold of the slew startpoint to the voltage source for a falling transition.

TYPE

float

DEFAULT

80

DESCRIPTION

Specifies the threshold voltage that defines the startpoint of the falling slew calculation. The value is a percent of the voltage source. Allowed values are 0.0 - 100.0 inclusive; the default is 80.0.

This variable is one of 8 variables, listed in Table 1, that you must specify in order to perform delay calculation in the presence of annotated parasitics using the command **read_parasitics**. These variables interpret the cell delays and transition times from the Synopsys library.

Table 1

Variable Name	Default
rc_slew_lower_threshold_pct_rise	20.0
rc_slew_lower_threshold_pct_fall	20.0
rc_slew_upper_threshold_pct_rise	80.0
rc_slew_upper_threshold_pct_fall	80.0
rc_input_threshold_pct_fall	50.0
rc_input_threshold_pct_rise	50.0
rc_output_threshold_pct_fall	50.0
rc_output_threshold_pct_rise	50.0

The default values specify that a cell delay is defined from 50% of the voltage value for the input transition to 50% of the voltage value for the output transition. The default values also specify that a transition time, or slew, is defined from 20% to 80% of the voltage.

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

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta rc_slew_upper_threshold_pct_fall 0.000 ; # expert
#       Type    : float
#       Usage   : slew upper threshold for fall
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793797

rc_slew_upper_threshold_pct_rise

Specifies the percentage threshold of the upper slew endpoint to the voltage source for a rising transition.

TYPE

float

DEFAULT

80

DESCRIPTION

Specifies the threshold voltage that defines the endpoint of the rising slew calculation. The value is a percent of the voltage source. Allowed values are 0.0 - 100.0 inclusive; the default is 80.0.

This variable is one of 8 variables, listed in Table 1, that you must specify in order to perform delay calculation in the presence of annotated parasitics using the command **read_parasitics**. These variables interpret the cell delays and transition times from the Synopsys library.

Table 1

Variable Name	Default
rc_slew_lower_threshold_pct_rise	20.0
rc_slew_lower_threshold_pct_fall	20.0
rc_slew_upper_threshold_pct_rise	80.0
rc_slew_upper_threshold_pct_fall	80.0
rc_input_threshold_pct_fall	50.0
rc_input_threshold_pct_rise	50.0
rc_output_threshold_pct_fall	50.0
rc_output_threshold_pct_rise	50.0

The default values specify that a cell delay is defined from 50% of the voltage value for the input transition to 50% of the voltage value for the output transition. The default values also specify that a transition time, or slew, is defined from 20% to 80% of the voltage.

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

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param ta rc_slew_upper_threshold_pct_rise 0.000 ; # expert
#       Type    : float
#       Usage   : slew upper threshold for rise
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793816

si_analysis_logical_correlation_mode

Enables or disables logical correlation analysis during PrimeTime-SI delay calculation.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

When true (the default), PrimeTime-SI performs logical correlation analysis while calculating delta delays and slews. In logical correlation analysis, PrimeTime-SI considers the logical relationships between multiple aggressor nets where buffers and inverters are used, so that the analysis is less pessimistic. When false, PrimeTime-SI assumes that the aggressor nets switch together in the direction that causes a worst-case slowdown or speedup of a transition on a victim net. Logical correlation analysis requires some CPU time; if you want a faster but less accurate and more pessimistic analysis, set this variable to false.

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

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018558

si_analysis_logic_correlation_mode

Persistent parameter that controls whether logic relationships among aggressor signals are considered during noise analysis. This method, while more compute intensive, is much more accurate and thus reduces the pessimism. In the current release, only the logic of inverter chains can be considered.

Syntax:

```
si_analysis_logic_correlation true | false
```

where the values have the following meanings:

`true`	Enable logic correlation analysis.
`false`	Do not perform logic correlation analysis.

The default value is *true* when the value of the *ta* parameter *si_xtalk_model* is *1* or *2*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793819

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**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018516-7

si_enable_analysis

Persistent parameter that controls whether to calculate the noise, and the impact of noise on timing.

Syntax:

```
si_enable_analysis true | false
```

where the values have the following meaning:

`true`	Perform a noise analysis and take the impact noise has on timing into account during timing analysis.
`false`	Ignore the effects of noise during timing analysis.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793820

si_filter_accum_aggr_noise_peak_ratio

Specifies the threshold for the accumulated voltage bumps introduced by aggressors at a victim node, divided by Vcc, below which aggressor nets can be filtered out during electrical filtering.

TYPE

float

DEFAULT

0.03

DESCRIPTION

Specifies the threshold for the accumulated voltage bumps introduced by aggressors at a victim node; the default is 0.03. This variable, along with **si_filter_per_aggr_noise_peak_ratio**, makes up a pair of variables used by PrimeTime-SI during the electrical filtering phase, to determine whether an aggressor net can be filtered.

An aggressor net, along with its coupling capacitors, is filtered when either of the following are true:

1. The peak voltage of the voltage bump induced on the victim net divided by Vcc is less than the value of **si_filter_per_aggr_noise_peak_ratio**.

2. The accumulated peak voltage of voltage bumps induced on the victim by aggressor to the victim net divided by Vcc is less than the value of **si_filter_accum_aggr_noise_peak_ratio**.

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

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018566

filter_accum_aggr_noise_peak_ratio

Persistent parameter that specifies the aggregated noise peak voltage threshold below which all aggressors on a net are ignored. The aggregated noise peak contains the contributions of all aggressors. The threshold is specified as a fraction of the supply voltage.

Syntax:

`filter_accum_aggr_noise_peak_ratio` *`ratio`*

where *`ratio`* is the minimum aggregated noise peak value as a fraction of the supply voltage below which the noise from all aggressors on a net are ignored.

The default value is *0.030*.