APPENDIX

Part Two

Trial Exhibit 1440, pages 1-115

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790140	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025202
add_to_collection Adds objects to a collection, resulting in a new collection. The base collection remains unchanged.	**add_to_collection** Given a collection *base_collection* and one or more *objects*, returns a new collection with all objects of the base collection and all the listed objects. In addition, you can choose to filter out all duplicate objects from the new collection. The base collection is not modified.
SYNTAX collection **add_to_collection** *base_collection* *object_spec* [-unique] collection*base_collection* list *object_spec*	**Syntax** add_to_collection *base_collection objects* \ [-unique]
ARGUMENTS base_collection Specifies the base collection to which objects are to be added. This collection is copied to the result collection, and objects matching *object_spec* are added to the result collection. *base_collection* can be the empty collection (empty string), subject to some constraints, explained in the DESCRIPTION. object_spec Specifies a list of named objects or collections to add. If the base collection is heterogeneous, only collections can be added to it. If the base collection is homogeneous, the object class of each element in this list must be the same as in the base collection. If it is not the same class, it is ignored. From heterogeneous collections in the *object_spec*, only objects of the same class of the base collection are added. 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. The *object_spec* has some special rules when the base collection is empty, as explained in the DESCRIPTION. -unique Indicates that duplicate objects are to be removed from the resulting collection. By default, duplicate objects are not removed.	where the arguments have the following meaning: *base_collection* — The collection to which you want to add an object. *objects* — The list of objects you want to add unique — Removes duplicate objects from the resulting collection.

Case No. 3:13-cv-02965-MMC
PLNTF Exhibit No. 1440
Date Entered FEB 29 2016
Signature TRACY LUCERO

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790146	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025229
all_fanin Creates a collection of pins/ports or cells in the fanin of specified sinks.	**all_fanins** Retrieves either all pins and ports or cells that belong to the fan-in cone of one or more specified pins. Pins that are referenced in timing constraints, for example, in case-analysis, are also considered part of the fan-in cone.
SYNTAX collection **all_fanin** -to *sink_list* [-flat] [-only_cells] [-startpoints_only] [-levels level_count] [-pin_levels pin_count] [-step_into_hierarchy] list *sink_list* int *level_count*	**Syntax** all_fanins -to *collection* \ [-only_cells] \ [-flat] \ [-startpoints_only]
ARGUMENTS -to *sink_list* Specifies a list of sink pins, ports, or nets in the design. Each object is a named pin, port, or net, or a collection of pins, ports, or nets. The timing fanin of each sink in *sink_list* becomes part of the resulting collection. If a net is specified, the effect is the same as listing all driver pins on the net. This argument is required. -startpoints_only When this option is specified, only the timing startpoints will be included in the result. -only_cells The result will include only cells in the timing fanin of the *sink_list* and not pins or ports. -flat There are two major modes in which **all_fanin** functions: hierarchical (the default) and flat. When in hierarchical mode, only objects within the same hierarchical level as the current sink are included in the result. In flat mode, the only non-leaf objects in the result will be hierarchical sink pins. -levels cell_count The traversal will stop when reaching a depth of search of *cell_count* hops, where the counting is performed over the layers of cells of same distance from the sink. -pin_levels pin_count The traversal will stop when reaching a depth of search of *pin_count* hops, where the counting is performed over the layers of pins of same distance from the sink. -step_into_hierarchy This option may only be used in hierarchical mode and only has effect with either **-levels** or **-pin_levels**. Without the switch, a hierarchical block at the same level of hierarchy as the current sink is considered to be a cell;	where the arguments have the following meaning: -to *collection* — Set of pins and ports whose fan-in cone is retrieved. [-only_cells] — Return the cells of the fanin cone instead of the pins and ports. [-flat] — Hierarchical pins, that is, pins that do not exist in a flattened netlist must not be returned. [-startpoints_only] — Only return pins and ports at the start of the fan-in cones.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790148	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025232

all_fanout

Creates a collection of pins/ports or cells in the fanout of the specified sources.

SYNTAX

```
collection all_fanout -from source_list
-clock_tree [-flat]
[-only_cells] [-endpoints_only]
[-levels level_count]
[-pin_levels pin_count]
[-step_into_hierarchy]

list  source_list
int   level_count
```

ARGUMENTS

-from *source_list*
Specifies a list of source pins, ports, or nets in the design. Each object is a named pin, port, or net, or a collection of pins, ports, or nets. The timing fanout of each source in *source_list* becomes part of the resulting collection. If a net is specified, the effect is the same as listing all load pins on the net. This option is exclusive with the **-clock_tree** option.

-clock_tree
Indicates that all clock source pins and/or ports in the design are to be used as the list of sources. Clock sources are specified using **create_clock**. If there are no clocks, or if the clocks have no sources, the result is the empty collection. This option is exclusive with the **-from** option.

-endpoints_only
When this option is specified, only the timing endpoints will be included in the result.

-only_cells
The result will include only cells in the timing fanout of the *source_list* and not pins or ports.

-flat
There are two major modes in which **all_fanout** functions: hierarchical (the default) and flat. When in hierarchical mode, only objects within the same hierarchical level as the current source are included in the result. In flat mode, the only non-leaf objects in the result will be hierarchical source pins.

-levels cell_count
The traversal will stop when reaching a depth of search of *cell_count* hops, where the counting is performed over the layers of cells of same distance from the source.

all_fanouts

Retrieves either all pins and ports or cells that belong to the fan-out cone of one or more specified pins. Pins that are referenced in timing constraints, such as the effect of disable timing or case-analysis, are also considered part of the fan-out cone.

Syntax

```
all_fanouts -from collection \
            -only_cells \
            -flat \
            -endpoints_only
```

where the arguments have the following meaning:

`-from collection`	Set of pins and ports whose fan-out cone is retrieved.
`-only_cells`	Return the cells instead of the pins and ports in the fan-out cone.
`-flat`	Hierarchical pins, that is, pins that do not exist in a flattened netlist, must not be returned.
`-endpoints_only`	Only return pins and ports, or cells, at the end of the fan-out cones.

-pin_levels pin_count
The traversal will stop when reaching a depth of search of *pin_count* hops, where the counting is performed over the layers of pins of same distance from the source.

-step_into_hierarchy
This option may only be used in hierarchical mode and only has effect with either **-levels** or **-pin_levels**. Without the switch, a hierarchical block at the same level of hierarchy as the current sink is considered to be a cell; the output pins are considered a single level away from the related input pins, regardless of what is inside the block. With the switch enabled, the counting is performed as though the design were flat, and although pins inside the hierarchy are not returned, they determine the depth of the related input pins.

4

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790161	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025241

append_to_collection

Add object(s) to a collection. Modifies variable.

append_to_collection

Appends the specified collection *object_collection* to the collection held by the collection variable *base_collection*. If this variable does not exist, this command creates it. You can choose to remove the duplicate objects from the resulting collection.

SYNTAX

```
collection add_to_collection
var_name
object_spec
[-unique]
collection var_name
list     object_spec
```

Syntax

```
append_to_collection base_collection object_collection \
                     [-unique]
```

ARGUMENTS

var_name
: Specifies a variable name. The objects matching *object_spec* are added into the collection referenced by this variable.

object_spec
: Specifies a list of named objects or collections to add.

-unique
: Indicates that duplicate objects are to be removed from the resulting collection. By default, duplicate objects are not removed.

where the arguments have the following meaning:

base_collection	The variable that holds the original collection. Is created if needed.
object_collection	A collection of objects to add to the original collection.
unique	Removes duplicate objects from the resulting collection stored in the variable *base_collection*.

5

characterize_context

Captures the timing context of a list of instances.

SYNTAX

```
string characterize_context [-timing] [-environment]
[-design_rules]
[-constant_inputs]
[-no_boundary_annotations]
cell_list

list  cell_list
```

ARGUMENTS

-timing
: Characterizes timing information; for example, clocks, input and output delays, and timing exceptions.

-environment
: Characterizes environment-related information; for example, operating conditions (process, temperature, and voltage), wire load model, capacitive loads on input and output pins, and driving cell information on input pins.

-design_rules
: Characterizes design rules; for example, max_capacitance, max_transition, and max_fanout.

-constant_inputs
: Characterizes logic constants propagated to input pins of the instance being characterized by the case analysis capability of PrimeTime.

-no_boundary_annotations
: Disables characterization of annotated capacitance on boundary nets as annotated capacitance in the characterized instance. Instead, the port wire capacitance is adjusted to account for any difference between the estimated and annotated values. By default, PrimeTime characterizes annotated capacitance on boundary nets as annotated capacitance in the characterized instance.

cell_list
: Specifies a list of instances to characterize.

characterize_context

Generates timing constraints for a set of cell instances. Typically, this command is used in a top-down hierarchical design flow to derive submodule timing constraints from the top-level constraints. These timing constraints include operating conditions, clock port delays, input port arrival times, output delay constraints, path exceptions, design rule constraints, and case analysis settings. The constraints are written in SDC format to a file named *module_name.derive_sdc* in the work directory or in a specified directory. These timing constraints can then be used to design the submodule in a separate project.

Syntax

```
characterize_context module_inst_list \
        [-output_dir path_and_file_name]
```

where the arguments have the following meaning:

module_inst_list
: List of module instances for which to generate timing constraints.

[-output_dir *path_and_file_name*]
: Path name to the output file. If this value is a file name or a relative path, then the location is relative to the work directory. By default, a file named *module_name.derive_sdc* is generated in the current work directory.

6

check_timing

Shows possible timing problems for design.

SYNTAX

```
string check_timing [-verbose]
[-significant_digits digits]
[-ms_min_separation delta]
[-override_defaults check_list]
[-include check_list]
[-exclude check_list]

float delta
int   digits
list    check_list
```

ARGUMENTS

-verbose
Shows detailed information about potential problems.

-significant_digits *digits*
Specifies the number of digits of precision to be displayed by warnings that show floating point numbers. 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.

-ms_min_separation *delta*
Minimum separation value between master and slave clocks. The default minimum separation is 0.0.

-override_defaults *check_list*
Overrides the checks in **timing_check_defaults** using *check_list*. See the man page of **timing_check_defaults** for its default value.

-include *check_list*
Adds the checks listed in *check_list* to the checks in **timing_check_defaults**.

-exclude *check_list*
Subtracts the checks listed in *check_list* from the checks in **timing_check_defaults**.

check_list
Gives the list of checks to be performed. Each element in this list is one of the following strings: clock_crossing, data_check_multiple_clock, data_check_no_clock, generated_clocks, generic, latch_fanout, latency_override, loops, ms_separation, multiple_clock, no_clock, no_input_delay, retain, signal_level, unconstrained_endpoints.

check_timing

Reports timing problems in the design. The following timing violations are reported:

- latch_fanout—A latch fans out to itself or to another latch connected to the same clock.
- no_clock—No clock reaches a sequential cell's clock pin.
- no_driving_cell—A port does not have a driving cell constraint.
- no_input_delay—An input port does not have an input delay constraint.
- unconstrained_endpoints— An endpoint of a timing path, such as an output port or data pin of a sequential cell, does not have a timing constraint set.

Syntax

```
check_timing [-verbose] \
             [-extra] \
             [-scenario name]
```

where the arguments have the following meaning:

[-verbose]	In the report, include the names of the pins and ports with timing constraint problems.
[-extra]	In addition to the regular timing checks, also check the completeness of the timing arcs in the Liberty files.
[-scenario *name*]	Specifies the scenario file to use. The scenario file contains the conditions under which to analyze the design such as Process-Voltage-Temperature corners. If you do not provide a scenario name, the timing checks are based on the SDC and Liberty file set for the current design.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790192	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025383
compare_collections Compares the contents of two collections. If the same objects are in both collections, the result is "0" (like string compare). If they are different, the result is nonzero. The order of the objects can optionally be considered.	**compare_collections** Compares collections. If both collections contain the same objects, the command returns *0*; otherwise, it returns *-1*. You can control whether the order of objects should be taken into account.

SYNTAX
 	
`int` **`compare_collections`** `[-order_dependent]` *`collection1 collection2`*
`collection` *`collection1`*
`collection` *`collection2`*
	Syntax
	
	`compare_collections` *`collection1 collection2`* `\`
	`[-order_dependent]`

ARGUMENTS
 	
`-order_dependent`
Indicates that the order of the objects is to be considered; that is, the collections are considered to be different if the objects are ordered differently.
 	
`collection1`
Specifies the base collection for the comparison. The empty string (the empty collection) is a legal value for the *collection1* argument.
 	
`collection2`
Specifies the collection with which to compare to *collection1*. The empty string (the empty collection) is a legal value for the *collection2* argument.
	where the arguments have the following meaning:
	
	`collection1` and *`collection2`*
	Collections to compare.
	
	`[-order_dependent]`
	Indicates that collections are only considered identical if the order of the objects in the collections is the same.

8

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790201	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025391
connect_net Connects a net to specified pins or ports.	**connect** Connects port and pins to a net. If the port and pins are already connected to other nets, you must explicitly specify that you want to reconnect them; otherwise, you get an error. If you want to connect a port to a net that is already connected a port, you must have marked these ports and the net as a feedthrough upon creation, and you must mark this connection as a feedthrough as well. By default, the *connect* Tcl command honors the *dont_touch* attribute set on a net and issues the *ChgDontTouch* error when you try to connect to such a net. You can however force a connection.

SYNTAX
 	
`int connect_net net object_spec`
`stringnet`
`list object_spec`
	Syntax
	
	`connect -net net_name \`
	`pins_and_ports \`
	`[-reconnect]\`
	`[-feedthru] \`
	`[-force]`

ARGUMENTS net Specifies the name of the net to which the pins and ports are to be connected. object_spec Specifies a list of pins or ports to connect to *net*.	where the arguments have the following meaning: `-net net_name` — Name of the net to which you want to connect the pins and ports. *pins_and_ports* — Names of the pins and ports to connect to the net. By default, only pins and ports that are not yet connected to a net are allowed to be connected. Use the *-reconnect* argument to allow all pins and ports to be connected to this net. `[-reconnect]` — Allows you to specify pins and ports that are already connected to nets. Those pins and ports are disconnected from their original net and reconnected to the specified net. `[-feedthru]` — Connection is part of a feedthrough. `[-force]` — Ignore the *dont_touch* attribute of a net, and, therefore, do not report the *ChgDontTouch* error when a net changes.

9

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790204	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025433
copy_collection Duplicates the contents of a collection, resulting in a new collection. The base collection remains unchanged.	**copy_collection** Returns a new collection that contains the same objects as a specified collection. Note that the objects are not copied, only the collection of those objects.

SYNTAX
 	
`collection copy_collection collection1`
`collection collection1`
	Syntax
	
	`copy_collection collection`

ARGUMENTS collection1 Specifies the collection to be copied. If the empty string is used for the *collection1* argument, the command returns the empty string (a copy of the empty collection is the empty collection).	where *collection* are the objects you want to copy.

10

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790240-41	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44

create_operating_conditions

Creates a new set of operating conditions in a library.

Command: create_operating_conditions
standard SDC command

SYNTAX

```
int create_operating_conditions
          -name name -library library_name
          -process process_value -temperature temperature_value
          -voltage voltage_value [-tree_type tree_type]
          [-calc_mode calc_mode]
          [-rail_voltages rail_value_pairs]

string name
string library_name
float process_value
float temperature_value
float voltage_value
string tree_type
string calc_mode
Tcl list rail_value_pairs
```

```
option:
  -name string                          name of operating condition (require)
  -library string                       name of library (require)
  -process double(0.000)                process scaling factor (require)
  -temperature double(0.000)            temperature value (require)
  -voltage double(0.000)                voltage value (require)
  -tree_type tree_type(balanced_tree)   tree type
                                        tree_type = balanced_tree |
                                        best_case_tree | worst_case_tree
  -calc_mode *                          not supported yet
  -rail_voltage *                       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
  --list_options                        list current option values
  --load_options ...                    load current option values
  --license                             list required licenses
  --help                                display command help
```

ARGUMENTS

-name *name*
Specifies the name of the new set of operating conditions.

-library *library_name*
Specifies the name of the library for the new operating conditions.

-process *process_value*
Specifies the process scaling factor for the operating conditions. Allowed values are 0.0 through 100.0.

-temperature *temperature_value*
Specifies the temperature value, in degrees Celsius, for the operating conditions. Allowed values are -300.0 through +500.0.

-voltage *voltage_value*
Specifies the voltage value, in volts, for the operating conditions. Allowed values are 0.0 through 1000.0.

-tree_type *tree_type*
Specifies the tree type for the operating conditions. Allowed values are *best_case_tree*, *balanced_tree* (the default), or *worst_case_tree*. The tree type is used to estimate interconnect delays by providing a model of the RC tree.

-calc_mode *calc_mode*
For use only with DPCM libraries. Specifies the DPCM delay calculator mode for the operating conditions; analogous to the *process* used in Synopsys libraries. Allowed values are *unknown* (the default), *best_case*, *nominal*, or *worst_case*. The default behavior (*unknown*) is to use worst case values during

analysis similarly to worst_case. If **-rail_voltages** are specified, the command sets all (worst_case, nominal, and best_case) voltage values.

-rail_voltages *rail_value_pairs*
Specifies a list of name-value pairs that defines the voltage for each specified rail. The name is one of the rail names defined in the library; the value is the voltage to be assigned to that rail. By default, rail voltages are as defined in the library; use this option to override the default voltages for specified rails.

description:
This command is the same as standard SDC command.

12

```
define_proc_attributes # Add extensions to a procedure
```

define_proc_attributes

Describes the help text of a specified Tcl procedure and describes the attributes of its arguments. This Tcl command allows you to use Aprisa's *parse_proc_arguments* Tcl procedure in your procedure to parse its arguments. By using both the *define_proc_attributes* and *parse_proc_arguments* Tcl commands, you integrate your procedure in the Aprisa environment. The meta arguments, such as *--h*, are enabled and the *info* and *help* commands also work for your procedure.

```
[-info info_text]           (Help string for the procedure)
[-define_args arg_defs]     (Procedure argument definitions for verbose help)
[-command_group group_name] (Command group for procedure. Default: Procedures)
[-permanent]                (Procedure cannot be overwritten)
[-hide_body]                (Body cannot be viewed with 'info body')
[-hidden]                   (Procedure does not show up in help or info)
[-dont_abbrev]              (Procedure can never be abbreviated)
   name                     (Procedure name)
```

Syntax

```
define_proc_attributes procedureName \
        -info string \
        -define_args * \
```

where the arguments have the following meaning:

procedureName	Name of the procedure.
`-info` *string*	One-line help string for the procedure.
`-define_args *`	Arguments and options of the procedure. This is a collection of argument definitions. Each argument definition has the following format:

{arg_name option_help value_help data_type attributes}

where:

arg_name	Name of the argument. If the name starts with a '-', it indicates a named argument. Otherwise, it is a positional argument.
option_help	Help string describing the argument.
value_help	Help string describing the acceptable values for the argument
data_type	Type of value expected for this argument, such as *float*, *string*, *boolean*, *one_of_string*
attributes	Additional attributes, such as *required*, *optional*, and for types *one_of_string*, the list of *values*.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790289	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025512
define_user_attribute Defines a new user-defined attribute.	**define_user_attribute** Defines a user attribute. You must define an attribute before using it. User attributes, similar to Aprisa attributes, have a name, a type, and can only be attached to objects of the specified type. You can create, modify, and delete user attributes. Attribute names must be unique within the class for which the attribute is defined, and the value of the attribute can only be of one data type.

SYNTAX
 	
`string` **`define_user_attribute`** `-type` *`data_type`* `-classes` *`class_list`*
`[-range_min` *`min`*`] [-range_max` *`max`*`]`
`[-one_of` *`values`*`] [-import]`
`[-quiet]` *`attr_name`*
`string` *`data_type`*
`list` *`class_list`*
`double` *`min`*
`double` *`max`*
`list` *`values`*
`string` *`attr_name`*
	Syntax
	
	`define_user_attribute` *`attr_name`* `\`
	`-type int | float | string | point \`
	`-class` *`class`*

ARGUMENTS
 	
`-type` *`data_type`*
Specifies the data type of the attribute. The supported data types are string, int, float, double, and boolean.
 	
`-classes` *`class_list`*
Defines the attribute for one or more of the classes. The valid object classes are design, port, cell, pin, net, lib, lib_cell or lib_pin.
 	
`-range_min` *`min`*
Specifies *`min`* value for numeric ranges. This is only valid when the *`data_type`* is int or double. Specifying a minimum constraint without a maximum constraint creates an attribute which accepts a value = *`min`*.
 	
`-range_max` *`max`*
Specifies *`max`* value for numeric ranges. This is only valid when the *`data_type`* is int or double. Specifying a maximum constraint without a minimum constraint creates an attribute which accepts a value = *`max`*.
 	
`-one_of` *`values`*
Provides a list of allowable strings. This is only valid when the data type is string.
 	
`-import`
Import this attribute from a design or library database.
 	
`-quiet`
Does not report any messages.
 	
`attr_name`
Specifies the name of the attribute.
	where the arguments have the following meaning:
	
	`attr_name` Attribute name.
	
	`-type int | float | string | point`
	Data type of its value.
	
	`-class` *`class`* Class name of object for which it is defined.

14

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790292	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025538
derive_clocks Creates clocks on source pins in design.	**derive_clocks** Creates clock definitions for all missing clocks so that design registers are constrained. These generated clocks are specified by a clock period and a set of times when clock edges occur, similar to regular clocks. The main difference is that the missing clocks are generated by Aprisa as opposed to being defined as part of the SDC constraints.
SYNTAX `string` **`derive_clocks`** `-period` *`period_value`* `[-waveform` *`edge_list`*`]` `float` *`period_value`* `list` *`edge_list`*	**Syntax** `derive_clocks -period` *`period`* `\` `-waveform` *`times`*
ARGUMENTS `-period` *`period_value`* Specifies the clock period of the automatically derived clocks. The clock period has a value greater than or equal to zero (value = 0). `-waveform` *`edge_list`* Specifies the rise and fall edge times of the clock, in library time units, over an entire clock period. It defines the clock edge specification. The first time that is listed is a rising transition; typically the first rising transition after time zero. There must be an even number of increasing times and alternating rise and fall times. If you do not specify an *`edge_list`* value, the command assumes a default waveform that has a rise edge of **0.0** and a fall edge of *`period_value`*/2.	where the arguments have the following meaning: `-period` *`period`* — Clock period for all missing clocks. `-waveform` *`times`* — Clock waveform, specified as a collection of times at which clock edges occur, starting with a rising edge.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790302	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025580
filter The **filter** command, a synonym for the **filter_collection** command, is a DC Emulation command provided for compatibility with Design Compiler.	**filter** This command is aliased to filter_collection.

16

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790303	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025581

filter_collection
 	
Filters an existing collection, resulting in a new collection. The base collection remains unchanged.
	filter_collection
	
	Retains from a given collection of objects only those objects that meet the specified criteria. Criteria are formulated as logic and pattern-matching expressions of attributes and values.
	
	The following operators are supported:
	
	`==  : equal`
	`!=  : not equal`
	`=~  : match pattern`
	`!~  : not match pattern`
	`<  : less than`
	`<=  : less or equal`
	`>  : greater than`
	`>=  : greater or equal`
	`&&  : Logic AND`
	`||  : Logic OR`
	
	The pattern matching syntax can be the Tcl regular expression or the Tcl globbing (also known as wildchar) syntax.

SYNTAX
 	
`collection` **`filter_collection`**
`base_collection expression`
`[-regexp]`
`[-nocase]`
 	
`collection`*`base_collection`*
`string` *`expression`*
	Syntax
	
	`filter_collection` *`collection expression`* `\`
	`[-regexp] \`
	`[-nocase]`

17

ARGUMENTS

base_collection
Specifies the base collection to be filtered. This collection is copied to the result collection. Objects are removed from the result collection if they are evaluated as **false** by the conditional *expression* value. Substitute the collection you want for *base_collection*.

expression
Specifies an expression with which to filter *base_collection*. Substitute the string you want for *expression*.

-regexp
Specifies that the =~ and !~ filter operators will use real regular expressions. By default, the =~ and !~ filter operators use simple wildcard pattern matching with the * and ? wildcards.

-nocase
Makes the pattern match case-insensitive. When you specify this option, you must also specify the **-regexp** option.

where the arguments have the following meaning:

collection	Collection on which the filter criteria is applied. Only objects that meet these criteria are returned by the filter.
expression	Expression using constants, object attributes, and the operators listed above.
[-regexp]	Use Tcl regular expression syntax for the pattern. The default is Tcl globbing syntax.
[-nocase]	Expression is case insensitive.

18

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790307	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025594
foreach_in_collection Iterates over the elements of a collection.	**foreach_in_collection** Executes a set of Tcl commands on each object from a given Aprisa collection. This command is equivalent to the *foreach* Tcl command but has the advantage that it operates directly on an Aprisa collection, which is much more efficient than a Tcl list.

SYNTAX
 	
`string foreach_in_collection itr_var collections body`
`stringitr_var`
`list collections`
`stringbody`
	Syntax
	
	`foreach_in_collection object collection { body }`

ARGUMENTS
 	
`itr_var`
Specifies the name of the iterator variable.
 	
`collections`
Specifies a list of collections over which to iterate.
 	
`body`
Specifies a script to execute per iteration.
	where the arguments have the following meaning:
	
	object — Tcl variable containing the object on which the body of Tcl commands is operated.
	
	collection — Collection of objects on which the body of Tcl commands is operated. This can also be a Tcl expression returning a collection.
	
	{ *body* } — Set of Tcl commands that is executed on each object in *collection*, one at a time.

19

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790312	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025613
get_attribute Retrieves the value of an attribute on an object.	**get_attribute** Returns the value of the specified attribute of an object.
SYNTAX string **get_attribute** [-class *class_name*] [-quiet] *object_spec attr_name* string*class_name* string*object_spec* or collection *object_spec* string*attr_name*	**Syntax** get_attribute *object_or_collection attr_name* \ [-class cell | net | port | pin | \ lib_cell | lib_pin]
ARGUMENTS -class *class_name* Specifies the class name of *object_spec*, if *object_spec* is a name. Valid values for *object_spec* are *design*, *port*, *cell*, *pin*, *net*, *lib*, *lib_cell*, *lib_pin*, *clock*, *timing_path*, and *timing_point*. You must use this option if *object_spec* is a name. -quiet Indicates that any error and warning messages are not to be reported. object_spec Specifies a single object from which to get the attribute value. *object_spec* must be is either a collection of exactly one object, or a name which is combined with the *class_name* to find the object. If *object_spec* is a name, you must also use the **-class** option. attr_name Specifies the name of the attribute whose value is to be retrieved.	where the arguments have the following meaning: *object_or_collection* — Object or collection of objects whose attribute value you want to retrieve. *attr_name* — Name of the attribute to retrieve. [-class cell | net | port | pin | lib_cell | lib_pin]] — Only examine objects of the specified type.

20

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790326	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025631

get_generated_clocks

Creates a collection of generated clocks.

SYNTAX

```
collection get_generated_clocks [-quiet] [-regexp] [-nocase] [-filter expression]
patterns
stringexpression
list  patterns
```

ARGUMENTS

-quiet
: Suppresses warning and error messages if no objects match. Syntax error messages are not suppressed.

-regexp
: Views the *patterns* argument as real regular expressions rather than simple wildcard patterns. Also, modifies the behavior of the =~ and !~ filter operators to compare with real regular expressions rather than simple wildcard patterns.

-nocase
: When combined with **-regexp**, makes matches case-insensitive. You can use **-nocase** only when you also use **-regexp**.

-filter *expression*
: Filters the collection with *expression*. For any generated clocks that match *patterns*, the expression is evaluated based on the generated clock's attributes. If the expression evaluates to true, the generated clock is included in the result.

patterns
: Matches generated clock names against patterns. Patterns can include the wildcard characters "*" and "?".

get_generated_clocks

Returns a collection of generated clock objects. The returned set of generated clocks may be selected by name, by the Aprisa attribute-based object filter, or by a combination of these.

Syntax

```
get_generated_clocks clockpattern\
                     [-filter attribute_constraint] \
                     [-regexp] \
                     [-nocase]
```

where the arguments have the following meaning:

clockpattern	Only return clocks whose names match the pattern. By default, Tcl globbing syntax is assumed.
[-filter *attribute_constraint*]	Only return generated clocks whose attributes meet the constraints defined in *attribute_constraint*. For more information on the syntax of the attribute constraints, see the *filter_collection* Tcl command.
[-regexp]	Treat the name patterns in *attribute_constraint* as a regular expression. By default, Tcl globbing syntax is assumed.
[-nocase]	Ignore case when performing name matches in *attribute_constraint*.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790350	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025661
get_object_name Gets the name of the object in a collection of exactly one object.	**get_object_name** Returns the name of the specified object, or returns a collection of names of a specified collection of objects.
SYNTAX `string` **`get_object_name`** *`collection`* `string`*`collection`*	**Syntax** `get_object_name` *`object`* | `-multiple` *`objects`*
ARGUMENTS `collection` Specifies the collection. This must be a collection of exactly one object.	where the arguments have the following meaning: *`object`* — Object whose name you want to retrieve. `-multiple` — Controls whether the command expects a single object or a collection of objects. *`objects`* — Collection of objects whose name you want to retrieve.

22

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790351

get_path_groups

Creates a collection of path groups from the current design. You can assign these path groups to a variable or pass them into another command.

SYNTAX

```
collection get_path_groups [-quiet] [-regexp] [-nocase] [-filter expression]
patterns
stringexpression
list  patterns
```

ARGUMENTS

-quiet
Suppresses warning and error messages if no objects match. Syntax error messages are not suppressed.

-regexp
Views the *patterns* argument as real regular expressions rather than simple wildcard patterns. Also, modifies the behavior of the =~ and !~ filter operators to compare with real regular expressions rather than simple wildcard patterns.

-nocase
When combined with **-regexp**, makes matches case-insensitive. You can use **-nocase** only when you also use **-regexp**.

-filter *expression*
Filters the collection with *expression*. For any path groups that match *patterns*, the expression is evaluated based on the path group's attributes. If the expression evaluates to true, the path group is included in the result.

patterns
Matches path group names against patterns. Patterns can include the wildcard characters "*" and "?".

Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025665

get_path_groups

Returns the list of names of all path groups given a glob-style pattern or a regular expression.

Syntax

```
get_path_groups clock_names \
                [-quiet] \
                [-regexp] \
                [-nocase]
```

where the values have the following meaning:

clock_names	Name or regular expression. All path groups with a launching clock or a capturing clock that have a name matching the regular expression are returned.
[-quiet]	Do not report errors or warnings.
[-regexp]	Treat *clock_names* as a regular expression.
[-nocase]	Do not consider case when matching clock names to the regular expression *clock_names*.

NOTE: This command returns a list of strings and not a collection of objects as most other *get** Tcl commands do. For this reason, no *-filter* argument is supported.

23

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790407	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025723
index_collection Creates a single element collection. I.e. Given a collection and an index into it, if the index is in range, extracts the object at that index and creates a new collection containing only that object. The base collection remains unchanged.	**index_collection** Retrieves an object at a specified position from a collection.
SYNTAX collection **index_collection** *collection1 index* collection *collection1* int *index*	**Syntax** index_collection *collection index*
ARGUMENTS collection1 Specifies the collection to be searched. index Specifies the index into the collection. Allowed values are integers from 0 to **sizeof_collection** - 1.	where the arguments have the following meaning: *collection* — Collection from which to retrieve the object. *index* — An integer indicating the position in the collection. The first object is at index 0.

24

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790408

insert_buffer
Inserts a buffer at one or more pins.

SYNTAX

string **insert_buffer** [-libraries *lib_spec*] [-inverter_pair] [-new_net_names *new_net_names*] [-new_cell_names *new_cell_names*] *pin_or_port_list lib_cell*

list *new_net_names*
list *new_cell_names*
list *pin_or_port_list*
string *lib_cell*

ARGUMENTS

-libraries *lib_spec*
If this option is specified, then PrimeTime resolves *lib_cellP from the libraries contained in the lib_spec* only. Libraries are searched in the order in which they appear in *lib_spec*. *lib_spec* can be a list of library names, or collections of libraries loaded into PrimeTime; the latter can be obtained using the **get_libs** command. You cannot specify this option if a full library cell name has been specified.

-inverter_pair
Indicates that a pair of inverting library cells is to be inserted instead of a single non-inverting library cell.

-new_net_names *new_net_names*
Specifies the net name to be given to the new net that PrimeTime inserts. This option can only be used if only one buffer or an inverter pair is being inserted. If one buffer is being inserted, you have to pass only one net name. If an inverter pair is being inserted, you have to pass two net names. These names can be any valid net names, but must be the leaf names i.e. not the hierarchical names. The new names must not contain embedded hierarchical separators. The new names must be unique in the current context (as specified by current_instance). If you use this option, you have to also use the **-new_cell_names** option.

-new_cell_names *new_cell_names*
Specifies the cell name to be given to the new cell that PrimeTime inserts. This option can only be used if only one buffer or an inverter pair is being inserted. If one buffer is being inserted, you have to pass only one cell name. If an inverter pair is being inserted, you have to pass two cell names. These names can be any valid cell names, but must be the leaf names i.e. not the hierarchical names. The new names must not contain embedded hierarchical separators. The new names must be unique in the current context (as specified by current_instance). If you use this option, you have to also use the **-new_net_names** option.

pin_or_port_list
Specifies a list of pins or ports to buffer.

Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44

Command: insert_buffer --interactive
internal development utility

option:

-net collection	the net to be buffered (require)
-buffer_cell collection	specify buffer library cell
-candidate_location point	buffer/inverters candidate location (require)
-skip_legalize	skip incremental placement legalization
-no_worse_timing	do not commit if timing does not improve
-inverter_pair	use inverter pair in stead of buffer
-connected_fanout collection	fanouts connected with added buffer.
-module collection	buffer/inverters module
-new_net_name string	specify the name of new net
-new_buf_name string	specify the name of new buffer

description:
This command is for AtopTech internal use only.

25

link_design

Resolves references in a design.

link_design

Builds the complete design by resolving references from instances to cells. This command performs the following functions:

- The different LEF, GDS, and Liberty libraries are combined to build an internal project library with cells that have all views needed by Aprisa (timing view, layout view, abstract view, and so on).
- The references in the imported design are replaced by references to cells from this project library.
- In a hierarchical design, this step also resolves the references from blocks in the design to cells representing hard blocks.

To build the design, the *link_design* Tcl command uses the following information:

- Verilog netlist (loaded with the *read_verilog* Tcl command)
- Logic/timing library (loaded with the *read_liberty* Tcl command)
- Physical library (loaded with the *read_lef*, *read_milkyway_fram*, *load_library* Tcl commands)
- PR_LIB or GDS abstract libraries stored with the project, when available. These need not be loaded explicitely. They allow you to examine the abstract in the context of the complete design.

For references to Liberty models that are not yet loaded, Aprisa uses the search path as set by *set_link_path* Tcl command and the content of the *liberty_search_path* variable to find missing timing information. Only Liberty libraries can be loaded on demand. All physical libraries must be loaded explicitly for the *link_design* Tcl command to add them to the project library.

Re-executing the *link_design* Tcl command on a design causes Aprisa to rebuild the internal project library using the current settings of the search path and library variables, and re-establishes the binding of instances in the design to cells in the project library.

When saving a design, you have the option to save a local copy of the physical cells of the internal project library with the design. The *gdslib* library contains the full layout in GDS format as loaded from GDS or OASIS. The *prlib* library contains the routing abstract in PR_LIB format, as loaded from LEF, Milkyway FRAM, or PR_LIB libraries. Bot the *prlib* and *gdslib* libraries are stored with the project. Once these project libraries exist, they are always loaded when the design is loaded. By default, the *link_design* Tcl command links these libraries after external libraries that were loaded using *read_lef*, *read_gds*, *read_oasis* or *load_library* Tcl commands with the *-link_first* argument, that is, cells of these external libraries will overrule cells saved in the project libraries. Using the *db* parameter, *use_own_lib_before_link_first*, the *prlib* and *gdslib* project libraries are linked first, that is, only cells that do not yet exist in the project libraries are picked up from external libraries, even if the *-link_first* argument was used. When the *use_own_lib_before_link_first* parameter is set, the *link_first* argument only affects the order of libraries linked after the project libraries.

You can control how Aprisa deals with missing information. Either a dummy cell, that is, an empty module in Verilog and an empty Liberty (.lib) cell with the proper size and correct pins is created, or an error is issued. A dummy cell representing a missing module in Verilog is called a *proto-module*. A dummy cell representing a missing library cell is called a *proto-lib_cell*. These proto-objects allow for better support of top down design.

Note that, even though they contain the same information, Aprisa makes a distinction between empty modules and missing modules. An empty module has an (empty) Verilog definition; A missing module is a module who was not found and whose Verilog definition was inferred. You control whether empty modules should be considered errors. Missing modules are always considered errors.

After linking the design the proto-module or proto-lib_cell can be converted again to the macro or the module using the *convert_macro_to_module* and *convert_module_to_macro* Tcl commands.

SYNTAX

```
string link_design [-verbose] [-remove_sub_designs] [-keep_sub_designs]
[design_name]
stringdesign_name
```

Syntax

```
link_design [-proto] \
          [-replace_own_pr_lib_with { string [string]...}] \
          [-replace_own_gds_lib_with { string [string]...}] \
          [-reload_lg_lib_with_diff_path] \
          [-no_proto_lib_cell] \
          [-strict] \
          [-max_ref_count_for_proto_module integer] \
          [-min_pin_count_for_proto_module integer] \
          [-allow_defined_empty_modules] \
          [-bind_lib_cell_only_to_empty_module] \
```

ARGUMENTS

-verbose
: Indicates that the linker is to display verbose messages.

-remove_sub_designs
: Indicates that subdesigns are to be removed after linking. By default, subdesigns are removed. Use this option to free up memory and improve performance. For more information, see the section entitled "Performance Considerations."

-keep_sub_designs
: Indicates that subdesigns are to be kept after linking. By default, subdesigns are removed. Use this option to keep the sub-designs around so that current_design can be changed to other designs later.

design_name
: Specifies the name of the design to be linked; the default is the current design.

where the arguments have the following meaning:

[-proto]
: Create prototype modules and library cells for all cells referenced in the netlist for which physical models are missing.

[-replace_own_pr_lib_with { *string* [*string*]...}]
: In the project library, replace all cells whose names are in the specified list with cells that have the same name from one of the loaded libraries. The project library can be saved on disk when the project is saved, and contains a copy of all cells with an Aprisa layout view that are used in the design.

`[-replace_own_gds_lib_with { string [string]...}]`
Replace all cells in the projects's own *gdslib* whose name is in the specified list of cell names with cells that have the same name from one of the loaded libraries. The project's own *gdslib* is created when the project is saved and contains a copy of all cells with a GDS layout view that are used in the design.

`[-reload_lg_lib_with_diff_path]`
Reload Liberty files from different Linux paths based on the current *search_path* or *link_path* settings. By default, a Liberty file is not loaded if another Liberty file with that same file name and same internal library name is already loaded, even if the path name is different.

Use this argument if you want to replace the Liberty file with a new version that is in a different path but has the same file name and internal library name.

After using this argument, the old version of the Liberty file can be removed from memory using the *remove_library -all_unused* Tcl command.

`[-no_proto_lib_cell]` Do not create a dummy library cell if no valid cell is found.

`[-strict]` Do not link unless all logical and physical library cells are present.This is the default behavior. If the *-proto* argument is used, this argument is not applicable. If for a module no abstract is found, the abstract is generated automatically, and the warning *LnkNoAbs* is issued.

`[-max_ref_count_for_proto_module maxinst]`
Maximum number of references allowed for proto-modules. If more than *maxinst* instances of the missing cell exist, no proto-module is created. The default value is *10*.

`[-min_pin_count_for_proto_module minpin]`
Minimum number of pins required for proto-module. If a missing cell has less than *minpin* pins, no proto-module is created. The default value is *50*.

`[-allow_defined_empty_modules]`
Do not treat empty modules as errors.

`[-bind_lib_cell_only_to_empty_module]`
Only bind a module to a corresponding library cell if that module is empty.

28

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790422	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025769
list_attributes Lists currently defined attributes.	**list_attributes** Lists the attributes of an object type if the *-class* argument is specified. Otherwise, it lists attributes of all available object types. For a list of supported object types, see Aprisa Classes.
SYNTAX string **list_attributes** [-application] [-class *class_name*] string*class_name* **ARGUMENTS** -application Lists application attributes as well as user-defined attributes. -class *class_name* Limit the listing to attributes of a single class. Valid classes are design, port, cell, net, and so on.	**Syntax** list_attributes [-class *class*]
ARGUMENTS -application Lists application attributes as well as user-defined attributes. -class *class_name* Limit the listing to attributes of a single class. Valid classes are design, port, cell, net, and so on.	where [-class *class*] is the class name of the object.

29

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790428	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025776
list_libraries Lists all libraries that are read into PrimeTime.	**list_libraries** Lists all libraries that are loaded into an Aprisa session. The command returns the full paths to libraries, or, if the project was saved with the *-all_libs*, *-prlib*, or *-gdslib* arguments, the paths are shown as *own_pr_lib*, *lg_lib*,or *gds_pr_lib*.

SYNTAX
 	
`string list_libraries [-only_used]`
	Syntax
	
	`list_libraries [lib_names] \`
	`  [-only_used] \`
	`  [-detail] \`
	`  [-lib_cell cells]`

ARGUMENTS
 	
`-only_used`
`  Indicates only the list libraries in use. A library is in use if a linked design links to library cells from the library.`
	where the arguments have the following meaning:
	
	`[lib_names]` — Only report on the specified libraries.
	
	`[-only_used]` — Only list the libraries if they contain cells that are used in the current project.
	
	`[-detail]` — List for each library all the cells that are used in the current project.
	
	`[-lib_cell cells]` — Only report on cells from the specified list. If this argument is used with the `-only_used` argument, only used cells from the list are reported.

30

Pl's Ex. 18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
parse_proc_arguments # Parse arguments to a procedure	Command: parse_proc_arguments <string:result> parse procedural arguments
-args arg_list (Argument list to be parsed) result_array (Name of array to use to store parse results)	option: -args { <string> ... } procedure arguments (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 --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
	description: This command parses prcedural arguments with their attributes defined by define_proc_attributes command. It must be used inside a procedure, and usually it is the first command to call. The usage is meaningful only if the procedure is defined with variable arguments 'args'. If the input arguments/options have no type error, the values will be assigned to the variable array indexed by the argument/option name.

31

PI's Ex. 631 (PrimeTime 2007.06 Executable) at SNPS_EXE_011	PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025942
read_aocvm # Read AOCVM data from aocvm file aocvm_file (File containing aocvm data)	**read_aocvm** Reads an advanced on-chip variation (OCV) derating model from a text file. This is the recommend method to build such a model. The *set_aocvm_component* Tcl command will be phased out eventually.
	Syntax `read_aocvm file`

	where `file` is the name of the advanced OCV model file to read.
	The syntax of the file is as follows:
	`version version_number`
	`object_type design | lib_cell | cell`
	`rf_type rise | fall | rise fall`
	`delay_typ cell | net | cell net`
	`derate_type early | late`
	`object_spec string`
	`depth set_of_M_floats`
	`distance set_of_N_floats`
	`table N_rows_M_colums`
	where the lines have the following meaning:
	`object_type design | lib_cell | cell`
	Derating model holds for standard cell, a specific block, or for a complete design.
	`rf_type rise | fall | rise fall`
	Derating factor holds for a rising event, a falling event, or both. By default, the setting holds for both.
	`delay_typ cell | net | cell net`
	Derating factor applies to cells, nets, or both.
	`derate_type early | late`
	Setting holds for an early path (signal path for hold analysis, or clock path for setup analysis) or for a late path (signal path for setup analysis, clock path for hold analysis).

32

`object_spec` *string*	This line is ignored for now. Will be implemented in future release.
`depth` *set_of_M_floats*	Different values of logic depth for which a column of derating factors is provided. Note that *M* can be zero, indicating that this is a one-dimensional model that has derating factors that are only a function of the distance.
`distance` *set_of_N_floats*	Different values of distance for which a row of derating factors is provided. Note that *N* can be zero, indicating that this is a one-dimensional model that has derating factors that are only a function of the depth.
`table` *N_rows_M_columns*	*N* rows, with in each row *M* values. Each row corresponds to a distance. Each column corresponds to a depth.

33

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790454	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025965
read_milkyway Reads in one linked design from milkyway database.	**read_milkyway_fram** Reads physical library data from a Milkyway FRAM library. The FRAM library contains the cell frame views, that is, for each cell the location of its pins and the blockages for the router on the various layers. In addition the route abstracts, also the *no_pg* and *no_signal* route guides are read. The reader fully supports the Milkyway 2008 standard.
SYNTAX int **read_milkyway** [-version version] [-netlist_only] [-library design_library] [-scenario scenario_name] *CEL_name* string *CEL_name* string *scenario_name* string *design_library*	**Syntax** read_milkyway_fram *mw_library_path*
ARGUMENTS -version *version* Specifies the version of the design to be read. For example, there are design files under the CEL view in the milkyway design library design_lib: 'design_lib/CEL/design1_pre_route1:1', 'design_lib/CEL/design1_post_route:2' etc. The 1 or 2 after the ':' is the version number of the design. The default is to read the most current version. -netlist_only Indicates that only the netlist is to be read; constraints are not read. The default is to read both netlist and constraints. -library *design_library* Specifies the absolute or relative path to the MW design library. This option can be left out if the variable **mw_design_library** specifies the path to the MW design library. -scenario *scenario_name* MW database is capable of storing multiple constraints that can correspond to various scenarios of running the design. This option specifies the name of the scenario for reading in constraints from MW database. The default is to not use a scenario. CEL_name Specifies the design filename to be read. For example, there are design files under the CEL view in the milkyway design library design_lib: 'design_lib/CEL/design1_pre_route1:1', 'design_lib/CEL/design1_post_route:2' etc. The design1_pre_route or design1_post_route are the CEL_name argument. Do not include version number in this argument.	where *mw_library_path* is the path name of the Milkyway database to read.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790454	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025967
read_milkyway Reads in one linked design from milkyway database.	**read_milkyway_tech** Imports the technology information from a Milkyway database.
SYNTAX int **read_milkyway** [-version version] [-netlist_only] [-library design_library] [-scenario scenario_name] *CEL_name* string *CEL_name* string *scenario_name* string *design_library*	**Syntax** read_milkyway_tech *filename* \ [-rlc_model *rlc_model*] \ [-rlc_corner MIN | NOM | MAX] \ [-routing_dir hv | vh]
ARGUMENTS -version *version* Specifies the version of the design to be read. For example, there are design files under the CEL view in the milkyway design library design_lib: 'design_lib/CEL/design1_pre_route1:1', 'design_lib/CEL/design1_post_route:2' etc. The 1 or 2 after the ':' is the version number of the design. The default is to read the most current version. -netlist_only Indicates that only the netlist is to be read; constraints are not read. The default is to read both netlist and constraints. -library *design_library* Specifies the absolute or relative path to the MW design library. This option can be left out if the variable **mw_design_library** specifies the path to the MW design library. -scenario *scenario_name* MW database is capable of storing multiple constraints that can correspond to various scenarios of running the design. This option specifies the name of the scenario for reading in constraints from MW database. The default is to not use a scenario. CEL_name Specifies the design filename to be read. For example, there are design files under the CEL view in the milkyway design library design_lib: 'design_lib/CEL/design1_pre_route1:1', 'design_lib/CEL/design1_post_route:2' etc. The design1_pre_route or design1_post_route are the CEL_name argument. Do not include version number in this argument.	where the arguments have the following meaning: *filename* — Name of the Milkyway file to read. [-rlc_model *rlc_model*] — Name of the RLC model that is created as part of the technology import. The default value is *MW*. [-rlc_corner MIN | NOM | MAX] — Read the RLC data of the specified corner. The default value is *MAX*. [-routing_dir hv | vh] — Routing direction for all layers. For the *hv* routing direction, which is the default value, metal1 is routed horizontally; metal2 is routed vertically; metal3 is routed horizontally, and so on. For the *vh* routing direction, metal1 is routed vertically; metal2 is routed horizontally; metal3 is routed vertically, and so on.

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790456-58

read_parasitics

Reads net parasitics information from an SPEF, DSPF, RSPF, or binary parasitics file and uses it to annotate the currently linked design.

SYNTAX

```
Boolean read_parasitics
[-format file_fmt]
[-complete_with completion_type]
[-lumped_cap_only]
[-pin_cap_included] [-increment]
[-path prefix]
[-keep_capacitive_coupling]
[-coupling_reduction_factor factor]
[-triplet_type ttype]
[-quiet] [-syntax_only]
    [-eco]
    [-original_file_name file_name]
[-ilm_context]
[-keep_variations]
    [-create_default_variations]
file_names

string file_fmt
string completion_type
string path_name
string file_names
string ofname
float  factor
```

ARGUMENTS

-format *file_fmt*
Specifies the format of the parasitics file. Allowed values are SPEF, DSPF, RSPF and SBPF (Synopsys Binary Parasitics Format). If **-format** is not specified, the application can determine whether the file is SPEF, DSPF, RSPF, or a compressed version of those three ascii formats. However, to read a file in SBPF, you must specify **-format SBPF**.

-complete_with *completion_type*
This option does not apply to the RSPF format. Indicates that a net with partially annotated parasitics is to be completed by inserting capacitances and resistances according to *completion_type*. Allowed values are *zero*, which completes the net by inserting zero capacitances and resistances; and *wlm*, which completes the net by inserting capacitances and resistances derived from wire load models. This option is equivalent to reading the parasitics file and then using the command **complete_net_parasitics -complete_with**.
Note: complete_net_parasitics and **read_parasitics -complete_with** complete a net only if all missing segments are between two pins and the nets are partially annotated (nets are not affected if they are fully annotated or have no annotation at all). Also, the net must be hierarchical, so that if the parasitics for the block-level parts of a net are missing, those

PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025972

read_parasitics

Loads parasitic information extracted from a third-party tool onto the current design. By default, this command reads parasitic data in SPEF format.

Syntax

```
read_parasitics filenames \
                [-format DSPF | SPEF] \
                [-lumped_cap_only] \
                [-pin_cap_included] \
                [-increment] \
                [-quiet] \
                [-syntax_only] \
                [-path path] \
                [-strip_path prefix] \
                [-merge_same_net_coupling] \
                [-condition { condition [condition]... }]
```

where the arguments have the following meaning:

filename	Name of files with parasitic information to load.
[-format DSPF | SPEF]	Format of the parasitic data. The default file format is *SPEF*.
[-lumped_cap_only]	Only annotate the total capacitance of the nets.
[-pin_cap_included]	RC networks already include the pin capacitances.
[-increment]	Add these parasitics to previously annotated parasitics instead of replacing them.
[-quiet]	Do not report the annotated parasitics in the log file.

36

parasitics could exist in the top-level net. . If any of these conditions are not met, you must correct the SPEF or DSPF file manually.

-lumped_cap_only
This option does not apply to the SBPF format. Indicates that only the total capacitance of nets is to be annotated as a lumped capacitance on the annotated nets. The RC networks specified in the parasitics file are discarded. The annotated lumped capacitance is the capacitance specified when the net is declared in the parasitics file.

-keep_capacitive_coupling
Indicates that the cross capacitors are to be kept in the RC networks data structure. This facilitates the capacitive crosstalk analysis, but does not turn it on. This option disables the **-coupling_reduction_factor** option; the command will fail if both options are specified. All coupling capacitors are split to ground with a factor of 1.0 if crosstalk analysis is not activated. This option applies to both the SPEF and the SBPF format. This option requires a PrimeTime SI license.

-pin_cap_included
Indicates that the RC networks are to include the pin capacitances. By default, the RC network does not include pin capacitances. This option does not apply to the RSPF format. The RC pi model in RSPF format has to always include effect of pin capacitances.

-increment
Indicates that previously annotated parasitics on the nets listed in the parasitics file are not to be overwritten. Additionally, any incomplete annotations in the parasitics file are not to be rejected. By default, the RC annotation specified in the parasitics file overwrites the previous parasitics annotations of the nets listed in the parasitics file. Use this option for annotating hierarchical parasitics files.

-path *prefix*
Specifies a relative path from the current design to the hierarchical design name for which the parasitics file has been created. By default, absolute pathnames are used. Use this option if the parasitics file refers to an object (for example, *net*) in a hierarchy (for example, *hier*). Do not use this option if the parasitics file refers to an absolute path (for example, *hier/net*).

-coupling_reduction_factor *factor*
This option applies only to the SPEF format and the SBPF format. A positive floating point number that specifies the factor to apply when reducing coupling capacitances to grounded capacitances. The default value is 1.0. This option is disabled if the **-keep_capacitive_coupling** option is specified. The command will fail if both options are specified.

-triplet_type *ttype*
This option applies only to the SPEF and PARA formats. Several values in SPEF and PARA, such as capacitor and resistor values, can be specified as triplets - min:typ:max. By default, PrimeTime takes the max value. Using this option, the user can select the min or typ value. Allowed values are *max* (the default), *typ*, and *min*.

-quiet
Indicates that the **report_annotated_parasitics** report is not to be generated when the parasitics file has been read. By default, after reading the parasitics file, the **report_annotated_parasitics -check** command is executed. This command reports the number of annotated nets, verifies the completeness of annotated RC networks on nets, and checks that no RC elements dangle. It is recommended that you use the **-quiet** option when reading multiple parasitics files in incremental mode.

[-syntax_only]	Do not load the parasitics but check if the SPEF syntax is valid. Note that this is only a syntax check. It does not check whether the parasitic file matches the netlist.
[-path *path*]	Relative path from the current design to the hierarchical design name for which the parasitic file has been created.
[-strip_path *prefix*]	Prefix of all SPEF or DSPF objects that needs to be stripped.
[-merge_same_net_coupling]	Merge coupled capacitances between same nets.
[-condition { *condition* [*condition*]... }]	List of SPEF parasitic conditions to load.

37

-syntax_only
Indicates that **read_parasitics** is to parse the file for syntax errors without performing any parasitic annotation. Use this option to troubleshoot your parasitics file and avoid generating error messages during the actual annotation. No design is required to use **-syntax_only**.

-ilm_context
Indicates that the annotation is being performed in the presence of Inteface Logic Models (ILMs). An original design parasitics can be used to annotate a design with ILMs using this option. This option does not issue error messages for missing nets, cells and pins.

-eco
Indicates that the files being currently annotated are ECO parasitics from Star-RCXT. PTSI can read ECO parasitics that are written out by Star-RCXT only. The ECO parasitics can be annotated only when there are some existing parasitics that are already annotated. ECO parasitic files contain re-extracted parasitics for just the ECO nets and their immediate coupling neighbours only and do not contain all the nets of the design. Incremental analysis can be performed after reading ECO parasitics.

-original_file_name *orig_file_name*
This option can only be used when *-eco* option is being used. If the original annotation is performed via multiple parasitic files into PTSI, then the ECO parasitic file corresponds to one of the original files (because it corresponds to one extracted database in Star-RCXT). PTSI will try to determine the corresponding original file but it is not always possible. You can use this option to specify which original parasitic file does the ECO file correspond to.

file_names
When the format is one of SPEF, DSPF, RSPF and SBPF, it specifies a list of files from which parasitics information is to be read.

-keep_variations
Indicates that the statistical parasitic information are to be kept in the RC networks data structure. This facilitates the variation aware timing analysis, but does not turn it on. This option applies only to SBPF format for now. Also, currently, this option does not work with either **-eco** option or **-increment** option. This option requires a PrimeTime VA license.

-create_default_variations
Specifies that default parasitic variations should be created for all the variation parameters. The default variations created are all assumed to be of normal distribution. The mean and sigma values are already present in the parasitic file.

38

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790477 -79

read_sdf

Reads leaf cell and net timing information from a file in Standard Delay Format (SDF) and uses that information to annotate the current design.

SYNTAX

```
string read_sdf [-load_delay net | cell]
[-analysis_type single | bc_wc | on_chip_variation]
[-min_file min_fname]
[-max_file max_fname]
[-path path_name]
[-type sdf_min | sdf_typ | sdf_max]
[-min_type sdf_min | sdf_typ | sdf_max]
[-max_type sdf_min | sdf_typ | sdf_max]
[-cond_use min | max | min_max] [-syntax_only]
[-strip_path strip_path_name]
[-quiet] [-worst ]
file_name

string path_name
string sdf_file_name
string min_sdf_file_name
string max_sdf_file_name
string strip_path_name
```

ARGUMENTS

-load_delay net | cell
: Indicates whether load delays are included in net delays or in cell delays in the timing file being read. The default is *cell*. The load delay is the portion of cell delay arising from the capacitive load of the net driven by the cell.

-analysis_type single | bc_wc | on_chip_variation
: Use this option only if you have not already set an analysis type with **set_operating_conditions -analysys_type**. If you are in min_max mode, the default is *bc_wc*. *single* indicates that only one operating condition is to be used. Specifying either *bc_wc* or *on_chip_variation* switches to min_max mode and causes both minimum and maximum delays to be read from the SDF file. Delays in SDF are represented in the form of triplets (sdf_min:sdf_typ:sdf_max). By default, the **-analysis_type bc_wc | on_chip_variation** option reads the *sdf_min* and *sdf_max* delays, respectively. To change this, use the **-min_type** and **-max_type** options.

-min_file *min_sdf_file_name*
: Use this option only if the minimum and maximum delays are in two separate SDF files. Specifies the file from which minimum delay timing information is to be read. The timing file must be in SDF format version v1.0, v2.0, v2.1 or v3.0.

-max_file *max_sdf_file_name*
: Use this option only if the minimum and maximum delays are in two separate

Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025975

read_sdf

Reads timing data from a Standard Delay Format (SDF) file and back-annotates the design.

NOTE: The listed unsupported options will be supported in a future release.

Syntax

```
read_sdf file \
          [-load_delay load_delay] \
          [-analysis_type analysis_type] \
          [-min_file min_file] \
          [-max_file max_file] \
          [-path path] \
          [-min_type sdf_min | sdf_type | sdf_max] \
          [-max_type sdf_min | sdf_type | sdf_max] \
          [-type sdf_min | sdf_type | sdf_max] \
          [-cond_use cond_use] \
          [-strip_path strip_path] \
          [-syntax_only] \
          [-quiet]
```

where the arguments have the following meaning:

file	Name of the SDF file to read.
[-load_delay *load_delay*]	Not supported yet.
[-analysis_type *analysis_type*]	Not supported yet.
[-min_file *min_file*]	Not supported yet.
[-max_file *max_file*]	Not supported yet.
[-path *path*]	Not supported yet.

39

SDF files. Specifies the file from which maximum delay timing information is to be read. The timing file must be in SDF format version v1.0, v2.0, v2.1 or v3.0.

-path *path_name*
Specifies the path from the current design to the subdesign for which the timing file has been created.

-type sdf_min | sdf_typ | sdf_max
Indicates which of the SDF triplet delay values are to be read from the SDF file. Delays in SDF are represented in the form of triplets (sdf_min:sdf_typ:sdf_max). By default, **read_sdf** reads the maximum delays *sdf_max*.
Note: If you use **-type** while in min/max mode (for example, if you use **-operating_conditions bc_bw | on_chip_variation**), a single value is annotated onto both min and max values of an arc.

-min_type sdf_min | sdf_typ | sdf_max
Specifies which of the SDF triplet delay values are to be read from the SDF file for minimum delay. Delays in SDF are represented in the form of triplets (sdf_min:sdf_typ:sdf_max). By default, **read_sdf** reads the minimum delays *sdf_min*. Use this option only with option **-analysis_type bc_wc | on_chip_variation**.

-max_type sdf_min | sdf_typ | sdf_max
Specifies which of the SDF triplet delay values are to be read from the SDF file for maximum delay. Delays in SDF are represented in the form of triplets (sdf_min:sdf_typ:sdf_max). By default, **read_sdf** reads the maximum delays *sdf_max*. Use this option only with option **-analysis_type bc_wc | on_chip_variation**.

-cond_use min | max | min_max
Use this option only if the SDF file includes some conditional delays using the SDF construct COND, and if the Synopsys library in use does not specify conditional delays. *min* indicates that the minimum of all conditional delays is to be used to annotate the corresponding timing arc. *max* indicates to use the maximum; *min_max* indicates min_max operating conditions; the minimum of all conditional delays is to be used for the minimum operating condition, and the maximum of all conditional delays is to be used for the maximum operating condition, You cannot use *min_max* with a single operating condition; you must be in min_max mode.

-syntax_only
Indicates that no timing annotation is to be performed; syntax only is to be processed. Use this option to verify that your SDF syntax is correct and will not issue any error messages.

-strip_path *strip_path_name*
Specifies a prefix path that is to be stripped from all SDF objects. Such a prefix path is usually a result of generating an SDF file for a subdesign, and using this subdesign as the current design.

-quiet
Use this option to skip execution of **report_annotated_delay** and **report_annotated_check** after reading SDF.

-worst
Indicates that **read_sdf** is to annotate the current design only with delays worse than the current annotated delays; applies to annotated net and cell delays and annotated timing checks. The worst delay is defined as the most pessimistic delay. This means primetime annotates the min of minima, and max of maxima values.

sdf_file_name
Specifies the file from which timing information is to be read. The timing file must be in SDF format version v1.0, v2.0, v2.1 or v3.0.

[-min_type sdf_min | sdf_typ | sdf_max]
Not supported yet.

[-max_type sdf_min | sdf_typ | sdf_max]
Not supported yet.

[-type sdf_min | sdf_type | sdf_max]
Not supported yet.

[-cond_use *cond_use*] Not supported yet.

[-strip_path *strip_path*]
Not supported yet.

[-syntax_only] Not supported yet.

[-quiet] Not supported yet.

40

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790484	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025983
read_verilog Reads in one or more Verilog files.	**read_verilog** Reads the hierarchy and connectivity information of a design from a set of Verilog files and builds a netlist ready for linking, that is, binding instances to modules and library cells using the *link_design* Tcl command. The actual design database is only created during the *link_design* step.

SYNTAX
 	
`string read_verilog [-hdl_compiler] file_names`
`list file_names`
	Syntax
	
	`read_verilog files \`
	`[-no_check] \`

ARGUMENTS
 	
`-hdl_compiler`
Indicates that the Verilog files are to be read using the PrimeTime external reader (ptxr) that uses HDL Compiler. Reading files in this way requires an HDL Compiler license while the read is in progress. HDL Compiler supports the complete Verilog language, but uses more CPU and memory than does the native PrimeTime Verilog reader.
 	
`file_names`
Specifies names of one or more files to be read.
	where the arguments have the following meaning:
	
	files — Names of the Verilog files to read.
	
	`[-no_check]` — Do not perform additional syntax and semantic checking. This expedites the reading, but may cause fatal errors down the road. Use this argument only if you read in Verilog files that have been previously checked.

41

Pl's Ex. 18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025987
`redirect  # Redirect output of a command to a file`	**redirect** Redirects the output of any Tcl command to a user-specified file or to a Tcl variable. The redirect command allows you to send the output to more than one destination, such as to the screen and a file using the *-tee* argument. If you do not want to tee the output, you can redirect the output via the standard Tcl method as follows: *command* > *file*

`[-append]  (Append output to the file)`
`[-tee]  (Tee output to the current output stream)`
`[-file]  (Output to a file (default))`
`[-variable]  (Output to a variable)`
	Syntax
	
	`redirect` *target command* \
	`[-append] \`
	`[-tee] \`
	`[-variable]`
	
	Or, to pass arguments to *command*:
	
	`redirect` *target {command command_options}*\
	`[-append] \`
	`[-tee]`
	`[-variable]`

`Target  (Name of file/variable target for redirect)` `command_string  (Command to redirect.  Should be in braces {}.)`	where the arguments have the following meaning: *target* — Name of the file to which to write the output, or, if the *-variable* argument is used, name of the Tcl variable in which to store the output. *command* — Name of the Tcl command whose output you want to redirect. *command_options* — Command options to *command*. [-append] — Append output to the file instead of overwriting the file. [-tee] — Copy the output to the screen. [-variable] — Redirect output to a Tcl variable instead of to a file.

42

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790495	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025994

remove_annotated_delay

Removes annotated delays from the design, either on specific cells or nets, between specific pins, or all annotated delays in the design.

remove_annotated_delay

Removes the delay information on selected nets, ports, or pins that was loaded from an external timing tool. You may want to use this command to ensure only Aprisa-calculated delay information is used and no delays slipped in from an external source.

SYNTAX

```
string remove_annotated_delay
[-all]
[-from from_list]
[-to to_list]
[object_spec]

list  from_list
list  to_list
list  object_spec
```

Syntax

```
remove_annotated_delay objects \
                       [-from pin_or_port] \
                       [-to pin_or_port] \
                       [-all]
```

ARGUMENTS

-all
: Indicates that all annotated delays in the design are to be removed. This option is exclusive of the **-from**, **-to**, and *object_spec* options.

-from *from_list*
: Specifies a list of pins or ports that are the startpoints of the timing arcs for which annotated delays are to be removed. You cannot combine this option with *object_spec*.

-to *to_list*
: Specifies a list of pins or ports that are the endpoints of the timing arcs for which annotated delays are to be removed. You cannot combine this option with *object_spec*.

object_spec
: Specifies a list of leaf cells or nets for which all annotated delays are to be removed. You cannot combine this option with **-from** and **-to**.

where the arguments have the following meaning:

objects	Names of objects for which to remove the delay.
[-from *pin_or_port*]	Remove delay from the specified pin or port.
[-to *pin_or_port*]	Remove delay to the specified pin or port.
[-all]	Remove all annotated delay.

43

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790501	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025996
remove_annotated_transition Removes previously-annotated transition times from pins or ports in the current design.	**remove_annotated_transition** Removes the transition time information from the specified pins that was loaded from an external timing tool. You may want to do this to ensure only Aprisa-calculated transition time information is used and no transition times slipped in from an external source.
SYNTAX int **remove_annotated_transition** -all | *pin_list* list *pin_list*	**Syntax** remove_annotated_transition *pin_list* \ [-all]
ARGUMENTS -all Indicates that all annotated transition times in the design are to be removed. **-all** and *pin_list* are mutually exclusive; you must use one of these, but not both. pin_list Specifies a list of pins or ports from which annotated transition times are to be removed. **-all** and *pin_list* are mutually exclusive; you must use one of these, but not both.	where the arguments have the following meaning: *pin_list* — Names of the pins for which to remove the annotated transition. [-all] — Remove all annotated pin transitions.

44

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790507	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0025998
remove_capacitance Removes capacitance on nets or ports.	**remove_capacitance** Removes user-specified capacitances from ports and nets that were set using the *set_load* Tcl command. The real extracted capacitances will be used instead.
SYNTAX string **remove_capacitance** *net_or_port_lis* list *net_or_port_list*	**Syntax** remove_capacitance *net_or_port_list*
ARGUMENTS net_or_port_list Specifies a list of ports and nets in the current design, whose capacitances are removed.	where *net_or_port_list* is the names of the nets or ports for which to remove capacitances.

45

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790508	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026000
remove_case_analysis Removes the case analysis value on input.	**remove_case_analysis** Removes case analysis that was set using the *set_case_analysis* Tcl command. Case analysis allows you to specify constant values for selected nets that are propagated through the design.
SYNTAX string **remove_case_analysis** *port_or_pin_list* list *port_or_pin_list*	**Syntax** remove_case_analysis *objects*
ARGUMENTS port_or_pin_list Lists ports or pins for which the case analysis entry is to be removed.	where *objects* is the list of objects for which to remove case analysis.

46

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790512	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026002
remove_clock Removes one or more clocks from the current design.	**remove_clock** Removes all or a selected set of clocks in the design.

SYNTAX
 	
`string remove_clock -all | clock_list`
`list clock_list`
	Syntax
	
	`remove_clock [clock_list] \`
	`[-all]`

ARGUMENTS
 	
`-all`
Specifies to remove all clocks in the current design.
 	
`clock_list`
Specifies a list of collections containing clocks or patterns matching the clock names.
	were the arguments have the following meaning:
	
	`[clock_list]` List of clocks to remove.
	
	`[-all]` Remove all clocks.

47

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790515	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
remove_clock_groups Removes specific exclusive or asynchronous clock groups from the current design.	Command: remove_clock_groups remove clock groups
SYNTAX Boolean **remove_clock_groups** -physically_exclusive | -exclusive | -asynchronous -name *name_list* | -all list *name_list*	option: -all remove all clock groups -name * clok group list -physically_exclusive physically exclusive -logically_exclusive logically exclusive -asynchronous asynchronous --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -physically_exclusive Specifies that groups set for physically exclusive clocks are to be removed. The **-physically_exclusive**, **-logically_exclusive** and **-asynchronous** options are mutually exclusive; you must choose only one. -logically_exclusive Specifies that groups set for logically exclusive clocks are to be removed. The **-physically_exclusive**, **-logically_exclusive** and **-asynchronous** options are mutually exclusive; you must choose only one. -asynchronous Specifies that groups set for asynchronous clocks are to be removed. The **-physically_exclusive**, **-logically_exclusive** and **-asynchronous** options are mutually exclusive; you must choose only one. -name *name_list* Specifies a list of clock groups to be removed, which matches the groups in the given names. You should use the **set_clock_groups** command to predefine these names. Substitute the list you want for *name_list*. The **-name** and **-all** options are mutually exclusive. -all Specifies to remove all groups set for exclusive or asynchronous clocks in the current design. The **-name** and **-all** options are mutually exclusive.	description: %remove_clock_groups -asynchronous -all

48

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790517	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
remove_clock_latency Removes clock latency information from specified objects.	Command: remove_clock_latency [db:object_list] standard SDC command
SYNTAX string **remove_clock_latency** [-source] [-clock *clock_list*] *object_list* list *clock_list* list *object_list*	option: -source remove source latency -all remove all latency offsets -offset remove latency offset -non_leaf apply to non-leaf offset -ocv remove ocv latency --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help
ARGUMENTS -source Specifies that clock source latency should be removed. -clock *clock_list* Removes any network latency defined on the pin/port objects in object_list which refers the clocks in clock_list from the design. If the **-clock** option is supplied when object_list refers to clock objects, a warning is issued that the option is not relevant in this case and execution of the command proceeds as if -clock was not given. This option does not remove a more general latency setting without any specific clock. object_list Provides a list of clocks, ports, or pins.	description: This command is the same as standard SDC command.

49

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790519	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
remove_clock_sense Removes unateness information defined on pins.	Command: remove_clock_sense <db:object_list> standard SDC command
SYNTAX string **remove_clock_sense** [-all] [-clocks *clock_list*] *object_list* list *clock_list* list *object_list*	option: -all — remove all clock unateness from current design -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 --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
ARGUMENTS -clocks *clock_list* Optionally specifies a list of clock objects to be associated with the given pin objects in object_list. If the **-clocks** option is specifed, only the unateness specified for that particular clock domain will be removed. Otherwise, unateness information for all clocks passing through the given pin objects will be removed. The **-clocks** option can only remove clock sense predefined by **set_clock_sense -clock**. It does not remove the default clock sense setting for this given pin. -all Remove all unateness information in current design. object_list Lists of pins with predefined unateness to remove.	description: This command is the same as standard SDC command.

50

remove_clock_uncertainty

Removes clock uncertainty information previously set by the **set_clock_uncertainty** command.

SYNTAX

```
string remove_clock_uncertainty
[object_list |
 -from from_clock
   | -rise_from rise_from_clock
   | -fall_from fall_from_clock
 -to to_clock
   | -rise_to rise_to_clock
   | -fall_to fall_to_clock]
[-rise]
[-fall]
[-setup]
[-hold]
[object_list]

list  from_clock
list  rise_from_clock
list  fall_from_clock
list  rise_to_clock
list  fall_to_clock
list  to_clock
list  object_list
```

ARGUMENTS

-from *from_clock* -to *to_clock*
: These two options specify the source and destination clocks for interclock uncertainty. You must specify either the pair of **-from/-rise_from/-fall_from** and **-to/-rise_to/-fall_to**, or *object_list*; you cannot specify both.

-rise_from *rise_from_clock*
: Same as the **-from** option, but indicates that *uncertainty* applies only to rising edge of the source clock. You can use only one of the **-from**, **-rise_from**, or **-fall_from** options. Use **-rise_from** instead of the obsolete option **-from_edge rise**.

-fall_from *fall_from_clock*
: Same as the **-from** option, but indicates that *uncertainty* applies only to falling edge of the source clock. You can use only one of the **-from**, **-rise_from**, or **-fall_from** options. Use **-fall_from** instead of the obsolete option **-from_edge fall**.

-rise_to *rise_to_clock*
: Same as the **-to** option, but indicates that *uncertainty* applies only to rising edge of the destination clock. You can use only one of the **-to**, **-rise_to**, or **-fall_to** options. Use **-rise_to** instead of the obsolete option **-to_edge rise**.

```
Command: remove_clock_uncertainty [db:object_list]
  Remove clock uncertainty constraints.
```

```
option:
  -all                      remove all uncertainty
  -append                   remove append uncertainty
  -end                      remove end uncertainty
  --get_option arg<1>       get option value
  --set_option ...          set option value
  --get_default arg<1>      get default value
  --set_default ...         set default value
  --list_options            list current option values
  --load_options ...        load current option values
  --license                 list required licenses
  --help                    display command help
```

```
description:
  Remove clock uncertainty constraints from design.
```


51

-fall_to *fall_to_clock*
Same as the **-to** option, but indicates that *uncertainty* applies only to falling edge of the destination clock. You can use only one of the **-to**, **-rise_to**, or **-fall_to** options. Use **-fall_to** instead of the obsolete option **-to_edge fall**.

object_list
Specifies a list of clocks, ports, pins, or cells from which uncertainty information is to be removed. You can use either the pair of **-from/-rise_from/ -fall_from** and **-to/-rise_to/-fall_to** options or the *object_list* option, but you cannot specify both; they are mutually exclusive.

-rise
Specifies that uncertainty is to be removed for only the rising clock edge. By default, uncertainty is removed for both rising and falling clock edges. This option is valid only for interclock uncertainty, and is now obsolete. Unless you need this option for backward-compatibility, use **-rise_to** instead.

-fall
Specifies that uncertainty is to be removed for only the falling clock edge. By default, uncertainty is removed for both rising and falling clock edges. This option is valid only for interclock uncertainty, and is now obsolete. Unless you need this option for backward-compatibility, use **-fall_to** instead.

-setup
Specifies that only setup check uncertainty is to be removed. By default, both setup and hold check uncertainties are removed.

-hold
Specifies that only hold check uncertainty is to be removed. By default, both setup and hold check uncertainties are removed.

52

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790547	PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026006
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.

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790550	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026008
remove_from_collection Removes objects from a collection, resulting in a new collection. The base collection remains unchanged.	**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 collection **remove_from_collection** *base_collection* x1collection*base_collection* list *object_spec*	**Syntax** remove_from_collection *base_collection subtract_collection*
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.	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.

54

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790559	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026010

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.

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. 252 (AP 09.10.rel.1) at ATopTech 0026016
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*
ARGUMENTS object_list Lists clocks, ports, or pins.	where *object_list* is a collection of clocks, pins, or ports.

57

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790589	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026020
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.

58

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790603	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026024
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.

59

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790640	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026032
report_attribute Reports the attributes on one or more 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
 	
`string **report_attribute** [-class *class_name*] [-nosplit] [-application] *object_spec*`
`string *class_name*`
`list *object_spec*`
	Syntax
	
	`report_attribute *objects* \`
	`[-class *class_name*] \`
	`[-application]`

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.
	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.

60

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*.

61

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.

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]
```

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.

63

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 number specified with the -nworst and -max_paths options.

-significant_digits *digits2*
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. The **-nosplit** option prevents line splitting and facilitates writing software to extract information from the report output.

[-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.

[-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.

64

	`[-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.

65

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790688-89	PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026058

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.

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.

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]
```

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.

66

-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**.

[-thresholds]	Report the voltage treshold 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*.
[-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.

67

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790703	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026067
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.

68

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790732	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
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*	option: -verbose show more details -all_violators not supported yet -significant_digits integer(3) number of digits after decimal point -nosplit 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 --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. -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.	description: Report minimum pulse width check information in current design.

69

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.

-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.

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.

Syntax

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

where the arguments have the following meaning:

[-threshold *val*]	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 *val*]	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.
[-threshold_high *val*]	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.

70

-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.

-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.

[-nets *nets*]	Only report noise information for the specified nets.
[-victim_only]	Report only victim peak value.

71

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. 252 (AP 09.10.rel.1) at ATopTech 0026111

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.

72

report_timing

Reports timing paths.

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]
```


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 variations.

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.

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] \
```


73

```
[-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
```

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.

```
[-nets] \
[-slack_greater_than threshold] \
[-slack_lesser_than threshold] \
[-ignore_register_feedback value] \
[-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] \
[-unconstrained_path]\
[-prefix filename_prefix] \
[-aocvm]
```

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.

74

-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.

`[-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.

`[-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	Paths with a maximum delay for a rising signal
max_fall	Paths with a maximum delay for a falling signal
min_rise	Paths with a minimum delay for a rising signal
min_fall	Paths with a minimum delay for a falling signal

75

-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.

`[-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:

D	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

76

-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.

[-true] Not supported yet.

[-true_threshold *threshold*] Not supported yet.

[-justify] Not supported yet.

[-false] Not supported yet.

[-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.

77

-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.

[-report_ignored_register_feedback]
Not supported yet.

[-group { name [name]... }]
Report on the specified groups.

[-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.

78

`[-brief_summary]`	Include a brief timing summary information.
`[-histogram]`	Include an end pin slack histogram.
`[-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.

79

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790863-65	PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026144

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.

80

-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**.

[-fall]	Only remove path exceptions that were defined for falling signals.
[-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.

81

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790887-89	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026201

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
```

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]
```

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.

83

-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.

84

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790893	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026204

set_annotated_transition

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

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

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

Syntax

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

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*.

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.

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790897

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.

PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026218

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

86

		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*).

87

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790953	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026279
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** Sets the *dont_touch* attribute on cells and nets. These cells or nets cannot be deleted during optimization, though they can be physically moved. If you set the *dont_touch* attribute on a module, then all its cells and nets inherit the attribute.
SYNTAX string **set_dont_touch** *object_list* [*value*] list *object_list* Boolean*value*	**Syntax** set_dont_touch *nets_or_cells* [0 | 1]
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 *nets_or_cells* is a collection of nets and cells for which to set the *dont_touch* attribute.

88

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791020	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
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 --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**.	

89

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791081-82	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
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 --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help

90

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.

-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.

description:
This command is for AtopTech internal use only.

91

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791103	PI's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026524
set_user_attribute Sets a user attribute to a specified value on an object.	**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 string **set_user_attribute** [-class *class_name*] [-quiet] *object_spec* *attr_name* *value* string*class_name* list *object_spec* string*attr_name* string*value*	**Syntax** set_user_attribute *objects att_name att_value* [-quiet]
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.	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.

92

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790974	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026565
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*
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.	where *collection* is a collection of objects.

93

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791121	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026570
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. By default, the objects are sorted in ascending order.
SYNTAX collection **sort_collection** [-descending] *collection1 criteria* collection *collection1* list *criteria*	**Syntax** sort_collection *objects* {*criteria_list*} \ [-descending]
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.

94

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791125	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026595
swap_cell Swaps one or more cells with a new design or library cell.	**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 int **swap_cell** *cell_list swap_in* [-dont_preserve_constraints] [-file *file_name*] [-format *file_format*] list *cell_list* string*swap_in* string*file_name* string *file_format*	**Syntax** swap_cell *cell* \ -rename *name* \
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*.	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.

95

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791142	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026662
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. 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.

96

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791164	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026719
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*.

97

write_sdc

Writes out a script in Synopsys Design Constraints (SDC) format.

SYNTAX

```
int write_sdc file_name [-version sdc_version]
[-compress compression] [-include categories list] [-nosplit]

stringversion
stringfile_name
stringcompression
list    categories 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.

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

```
write_sdc file \
          [-port_latency] \
          [-port_latency_only] \
          [-pin_latency_only] \
          [-latency_offset_only] \
          [-balanced_source_latency_only] \
          [-cancel_out] \
          [-extension] \
```

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.

	[-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.

99

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791179-83	Pl's Ex. 252 (AP 09.10.rel.1) at ATopTech 0026733

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) 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

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.

Syntax

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


100

```
[-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.

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

101

-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.

102

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

103

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

104

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791189-92	Pl's Ex. 646 (AP 09.10.rel.1 Executable) ZZATopTech 0000035-44
write_spice_deck Writes to a SPICE deck the paths or arcs generated by **get_timing_paths** or **get_timing_arcs**.	Command: write_spice <db:object_list> internal development utility
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*	option: -min min condition -max max condition --get_option arg<1> get option value --set_option ... set option value --get_default arg<1> get default value --set_default ... set default value --list_options list current option values --load_options ... load current option values --license list required licenses --help display command help

105

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.

description:
This command is for AtopTech internal use only

106

-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

107

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

108

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.

109

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. 235 (AP 09.10.rel.1) at ATopTech 0012934

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 true | false
```

where the values have the following meaning:

true	Propagate conditions on conditional arcs across sequential cells.
false	Do not propagate conditions on conditional arcs across sequential cells.

The default value is *true*.

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. 235 (AP 09.10.rel.1) at ATopTech 0012885

collection_result_display_limit

Runtime parameter that specifies the maximum number of objects in a collection to display in the shell.

Syntax:

```
collection_result_display_limit integer
```

where *integer* is the number of objects in the collection to display in the shell.

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. 235 (AP 09.10.rel.1) at ATopTech 0012936

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. 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. 235 (AP 09.10.rel.1) at ATopTech 0012937

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*.

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**.

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 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. 235 (AP 09.10.rel.1) at ATopTech 0012952

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
```

where the values have the following meaning:

basic	Use the simplified CCS model on selected net drivers if the ccsd_auto_switch parameter is enabled; Otherwise, use no CCS model.
advanced	Use the advanced CCS model on all net drivers, regardless of the ccsd_auto_switch parameter setting.

The default value is *basic*.