APPENDIX

Part One

Trial Exhibit 1439

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790140	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
add_to_collection Adds objects to a collection, resulting in a new collection. The base collection remains unchanged.	Command: add_to_collection <string:base_collection> <string:obj_spec> return a new collection with objects added to the base collection
SYNTAX collection **add_to_collection** *base_collection* *object_spec* [-unique] collection*base_collection* list *object_spec*	option: -unique — Remove duplicat objects from the resulting collection --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 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.	

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790161	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

append_to_collection

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

```
Command: append_to_collection <string:collection_var>
         <string:obj_spec>
  append objects to a base collection variable
```

SYNTAX

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

```
option:
  -unique                    Remove duplicat objects from the
                             resulting collection
  --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

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.

2

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.

```
Command: characterize_context <*:module_inst_list>
  Derive timing constraints for a set of cell instances.

option:
  -timing                     not supported yet
  -environment                not supported yet
  -design_rules               not supported yet
  -constant_inputs            not supported yet
  -no_boundary_annotations    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
```

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790179

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.

Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

Command: check_timing
Checks possible design timing problems

option:

-verbose	show more details
-scenario string	specify working scenario
--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

4

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790192	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
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.	Command: compare_collections <*:collection1> <*:collection2> compare the objects in the collections
SYNTAX int **compare_collections** [-order_dependent] *collection1 collection2* collection *collection1* collection *collection2*	option: -order_dependent compare objects order --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 -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.	

5

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790201	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
connect_net Connects a net to specified pins or ports.	Command: connect <*:pin_ports> connects port/pins to a net
SYNTAX int **connect_net** *net object_spec* string*net* list *object_spec*	option: -net * net (require) -reconnect disconnect the pin/port first if it is still connected --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 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*.	

6

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790213	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
create_configuration Creates a configuration for multi-scenario analysis.	Command: create_configuration Don't support.
SYNTAX **create_configuration** -global_data *file_list* list *file_list*	option: -global_data * order dependent setup files (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
ARGUMENTS -global_data An order dependent list of files containing everything that is common across the entire analysis. These files must at the very minimum contain the commands needed to read the design netlist.	

7

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790240-41	Pl's Ex. 643 (AP 07.11.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(0.0)	process scaling factor (require)
-temperature double(0.0)	temperature value (require)
-voltage double(0.0)	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.

8

-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.

9

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790289	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

define_user_attribute

Defines a new user-defined attribute.

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
```

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.

```
Command: define_user_attribute <string:attr_name>
  user defined attribute

option:
  -type type()                data type (require)
                              type = int | float | string |
                              point
  -class string               class name of object (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
```

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790292	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
derive_clocks Creates clocks on source pins in design.	Command: derive_clocks Create clocks to make design registers constrainted.
SYNTAX string **derive_clocks** -period *period_value* [-waveform *edge_list*] float *period_value* list *edge_list*	option: -period double(0.0) clock period (require) -waveform * clock waveform --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 -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.	

11

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790302	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
filter The **filter** command, a synonym for the **filter_collection** command, is a DC Emulation command provided for compatibility with Design Compiler.	Command: filter <*:collection> <string:expression> Create a new collection from subset of a collection.

12

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790307	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
foreach_in_collection Iterates over the elements of a collection.	Command: foreach_in_collection <*:itr_var> <*:collection> <*:body> evalulate the body script with each object in a collection
SYNTAX string **foreach_in_collection** *itr_var collections body* string*itr_var* list *collections* string*body*	option: --license list required licenses --help display command help
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.	

13

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790312	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
get_attribute Retrieves the value of an attribute on an object.	Command: get_attribute <*:object_or_collection> <string:attr_name> get object attribute
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*	option: -class string — class name of object -quiet — 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 -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.	

14

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790326	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
get_generated_clocks Creates a collection of generated clocks.	Command: get_generated_clocks <*:patterns> Get generated clock objects.
SYNTAX collection **get_generated_clocks** [-quiet] [-regexp] [-nocase] [-filter *expression*] *patterns* string*expression* list *patterns*	option: -quiet — not supported yet -regexp — use regular expression for pattern -nocase — case insensitive -filter string — filter_expression --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 -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 "?".	

15

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790350	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
get_object_name Gets the name of the object in a collection of exactly one object.	Command: get_object_name <*:object_or_collection> get name of a single object
SYNTAX string **get_object_name** *collection* string*collection*	option: --license list required licenses --help display command help
ARGUMENTS collection Specifies the collection. This must be a collection of exactly one object.	

16

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790407	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
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.	Command: index_collection <*:collection> <integer:index> retrieve an object from a collection
SYNTAX collection **index_collection** *collection1 index* collection *collection1* int *index*	option: --license list required licenses --help display command help
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.	

17

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790408	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
insert_buffer Inserts a buffer at one or more pins.	Command: insert_buffer insert_buffer --interactive internal development utility

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*

option:

-net *	the net to be buffered (require) (gui)
-buffer_cell *	specify buffer library cell (gui)
-candidate_location point	buffer/inverters candidate location (require) (gui)
-skip_legalize	skip incremental placement legalization (gui)
-no_worse_timing	do not commit if timing does not improve (gui)
-inverter_pair	use inverter pair in stead of buffer (gui)
-connected_fanout *	fanouts connected with added buffer. (gui)
-module *	buffer/inverters module (gui)
-new_net_name string	specify the name of new net (gui)
-new_buf_name string	specify the name of new buffer (gui)
--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
--current_options	load option value from memory
--interactive	enter into gui interactive mode
--end_interactive	exit from gui interactive mode
--license	list required licenses
--help	display command help

18

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.

19

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790418	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

link_design

Resolves references in a design.

SYNTAX

string **link_design** [-verbose] [-remove_sub_designs] [-keep_sub_designs] [*design_name*]
string*design_name*

```
Command: link_design
  build a design database from verilog net-list and
  library cells

ption:
  -no_proto_lib_cell              do not fake any prototyping
                                  lib_cell for empty modules
  -replace_own_gds_lib_with *     use cells in specified lib to
                                  repalce cells in project's
                                  own_gds_lib
  -max_ref_count_for_proto_module integer(10)
                                  max. number of references allowed
                                  for proto-module
  -min_pin_count_for_proto_module integer(50)
                                  min. number of pins for proto-
                                  module
  --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

-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.

21

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790422	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
list_attributes Lists currently defined attributes.	Command: list_attributes print out a list of attributes of an object class
SYNTAX string **list_attributes** [-application] [-class *class_name*] string*class_name*	option: -class string — class name of object -application — 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 -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.	

22

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790428	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
list_libraries Lists all libraries that are read into PrimeTime.	Command: list_libraries [string:lib_name] list_libraries --interactive lists all libraries that are read into AP
SYNTAX string **list_libraries** [-only_used]	option: -only_used — only list libraries/lib_cell used by current project (gui) -detail — list all the used lib_cells (gui) -lib_cell * — lib cell filter (gui-only) -select boolean(1) — auto select on browsing object (gui-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 --current_options — load option value from memory --interactive — enter into gui interactive mode --end_interactive — exit from gui interactive mode --license — list required licenses --help — display command help
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.	

23

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790454	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
read_milkyway Reads in one linked design from milkyway database.	Command: read_milkyway_fram <string:mw_library_path> reads physical library data from a Milkyway FRAM library
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*	
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.	option: --license list required licenses --help display command help

24

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790454	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
read_milkyway Reads in one linked design from milkyway database.	Command: read_milkyway_tech <string:filename> reads milkyway tech files
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*	option: -rlc_model string — name for the RLC-model (default: MW) -rlc_corner cond(MAX) — read the RLC data of the specified corner; cond = MIN | NOM | MAX -routing_dir routeDir(hv) — hv: metal1(Hor), metal2(Ver), metal3(Hor), ...; vh: metal1(Ver), metal2(Hor), metal3(Ver), ...; routeDir = hv | vh --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 -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.	

25

-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.

26

Pl'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
```

Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

```
Command: read_parasitics <file_names ...>
  annotates parasitic information onto the current design

option:
  -format para_format(SPEF)    read parasitics format
                               para_format = DSPF | SPEF
  -lumped_cap_only             only the total capacitance
                               of nets is annotated
  -pin_cap_included            the RC networks include the
                               pin capacitances
  -increment                   not to discard previously
                               annotated parasitics
  -quiet                       not to perform
                               report_annotated_parasitics
  -syntax_only                 check if the SPF is valid
  -path string                 Specifies the path from the
                               current design to the
                               subdesign design
  -strip_path string           strip off the prefix path
  -merge_same_net_coupling     merge coupled cap from the
                               same net
  -condition *                 list of parasitic
                               conditions of SPEF that are
                               being loaded
  --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
```


27

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 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.

28

-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
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.

29

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790477 -79	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

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.

```
Command: read_sdf <string:file_name>
  reads timing data from an SDF file
```

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
```

```
option:
  -load_delay string              not supported yet
  -analysis_type string           not supported yet
  -min_file string                not supported yet
  -max_file string                not supported yet
  -path string                    not supported yet
  -min_type min_max_type(sdf_min)
                                  not supported yet
                                  min_max_type = sdf_min | sdf_typ
                                  | sdf_max
  -max_type min_max_type(sdf_typ)
                                  not supported yet
                                  min_max_type = sdf_min | sdf_typ
                                  | sdf_max
  -type min_max_type(sdf_max)     not supported yet
                                  min_max_type = sdf_min | sdf_typ
                                  | sdf_max
  -cond_use string                not supported yet
  -strip_path string              not supported yet
  -syntax_only                    not supported yet
  -quiet                          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
```


30

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 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**.

31

-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.

32

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790484	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
read_verilog Reads in one or more Verilog files.	Command: read_verilog <string:filenames> reads Verilog files for linking
SYNTAX string **read_verilog** [-hdl_compiler] *file_names* list *file_names*	option: -no_check — do not perform additional syntax/semantic checking --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 -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.	

33

Pl's Ex. 18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
redirect # Redirect output of a command to a file	Command: redirect <string:file> <*:command> redirect output to a 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)	option: -append append output to file --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
target (Name of file/variable target for redirect) command_string (Command to redirect. Should be in braces {}.)	

34

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790495	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
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.	Command: remove_annotated_delay [*:object_list]
SYNTAX string **remove_annotated_delay** [-all] [-from *from_list*] [-to *to_list*] [*object_spec*] list *from_list* list *to_list* list *object_spec*	option: -from * — from pin or port -to * — to pin or port -all — remove all --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 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**.	

35

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790501	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_annotated_transition Removes previously-annotated transition times from pins or ports in the current design.	Command: remove_annotated_transition remove_annotated_transition <*:pin_list>
SYNTAX int **remove_annotated_transition** -all | *pin_list* list *pin_list*	option: -all — remove all annotated pin transitions --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 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.	

36

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790507	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_capacitance Removes capacitance on nets or ports.	Command: remove_capacitance <*:net_or_port_list> Remove user-specified net capacitance.
SYNTAX string **remove_capacitance** *net_or_port_lis* list *net_or_port_list*	option: --license list required licenses --help display command help
ARGUMENTS net_or_port_list Specifies a list of ports and nets in the current design, whose capacitances are removed.	

37

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790508	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_case_analysis Removes the case analysis value on input.	Command: remove_case_analysis <*:object_list> Remove user-specified case analysis.
SYNTAX string **remove_case_analysis** *port_or_pin_list* list *port_or_pin_list*	option: --license list required licenses --help display command help
ARGUMENTS port_or_pin_list Lists ports or pins for which the case analysis entry is to be removed.	

38

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790512	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_clock Removes one or more clocks from the current design.	Command: remove_clock remove_clock <*:clock_list> Remove clocks in the design.
SYNTAX string **remove_clock** -all | *clock_list* list *clock_list*	option: -all remove all clocks --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 Specifies to remove all clocks in the current design. clock_list Specifies a list of collections containing clocks or patterns matching the clock names.	

39

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790517	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_clock_latency Removes clock latency information from specified objects.	Command: remove_clock_latency <*: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 -offset remove latency offset --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.	

40

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790547	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_driving_cell Removes port driving cell information.	Command: remove_driving_cell <*:port_list> Remove driving cell constraints in the design.
SYNTAX string **remove_driving_cell** [-rise] [-fall] [-min] [-max] [-clock *clock_name*] [-clock_fall] *port_list* string*clock_name* list *port_list*	option: -rise not supported yet -fall not supported yet -min not supported yet -max not supported yet -clock_fall not supported yet -clock * 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 -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.	

41

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790550	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_from_collection Removes objects from a collection, resulting in a new collection. The base collection remains unchanged.	Command: remove_from_collection <*:collection> <string:obj_spec> remove objects from a collection

SYNTAX
 	
collection **remove_from_collection**
base_collection
x1collection*base_collection*
list *object_spec*
	option:
	  --license list required licenses
	  --help display command help

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.	

42

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790559	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_input_delay Removes input delay information from ports or pins.	Command: remove_input_delay <*:port_pin_list> Remove input delay on the 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*	option: -clock * not supported yet -clock_fall not supported yet -level_sensitive not supported yet -rise not supported yet -fall not supported yet -min not supported yet -max 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 -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.	

43

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790562	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_lib Removes one or more libraries from memory.	Command: remove_lib [*:object_list] remove library if it's not used
SYNTAX string **remove_lib** -all *libraries* list *libraries*	option: -all remove all 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 -all Removes all libraries. libraries Provides a list of libraries to remove.	

44

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790579-80	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_output_delay Removes output delay from output ports or pins.	Command: remove_output_delay <*:port_pin_list> Remove output delay on the ports or pins.
SYNTAX string **remove_output_delay** [-clock *clock_name*] [-clock_fall] [-level_sensitive] [-rise] [-fall] [-max] [-min] *port_pin_list* string*clock_name* list *port_pin_list*	option: -clock * not supported yet -clock_fall not supported yet -level_sensitive not supported yet -rise not supported yet -fall not supported yet -min not supported yet -max 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

45

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.

46

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 005677584	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_propagated_clock Removes a propagated clock specification.	Command: remove_propagated_clock <*:object_list> Remove clock propagated attribute on the objects.
SYNTAX string **remove_propagated_clock** *object_list* list *object_list*	option: --license list required licenses --help display command help
ARGUMENTS object_list Lists clocks, ports, or pins.	

47

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790603	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
remove_user_attribute Removes a user attribute from an object.	Command: remove_user_attribute <*:object_or_collection> <string:attr_name> remove user defined attribute from an object
SYNTAX string **remove_user_attribute** [-quiet] [-class *class_name*] *object_spec attr_name* string*class_name* list *object_spec* string*attr_name*	option: -class string — class name of object -quiet — 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 -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.	

48

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790640	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
report_attribute Reports the attributes on one or more objects.	Command: report_attribute <*:object_or_collection> report attributes of each object in the collection
SYNTAX string **report_attribute** [-class *class_name*] [-nosplit] [-application] *object_spec* string *class_name* list *object_spec*	option: -class string — class name of object -application — report application attributes too -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 -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.	

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790653-54	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
report_clock Reports clock-related information.	Command: report_clock [*:clock_list] Report clock information.
SYNTAX string **report_cloc** [-attributes] [-skew] [-groups] [-nosplit] [*clock_names*] list *clock_names*	option: -attributes — report clock attributes -skew — report clock latency and uncertainty -nosplit — prevents line splitting if column overflows --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
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.	

50

-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*.

51

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790672	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

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
```

```
Command: report_constraint
  Report timing violation in current design.
```

```
option:
  -all_violators              report all violators
  -pins *                     report DRV on specified pins only
  -verbose                    show more details
  -reason                     show reason optimization skips
  -path_type path_type(slack_only)
                              format of path report
                              path_type = end | slack_only
  -max_delay                  only display max delay and setup
                              constraints
  -min_delay                  only display min delay and hold
                              constraints
  -max_capacitance            only display max capacitance
                              constraints
  -min_capacitance            only display min capacitance
                              constraints
  -max_transition             only display max transition
                              constraints
  -min_transition             only display min transition
                              constraints
  -max_fanout                 only display max fanout load
                              constraints
  -min_fanout                 only display min fanout load
                              constraints
  -max_fanout_count           only display max fanout count
                              constraints
  -delay_noise                only display delay noise
                              violations
  -min_pulse_width            only display min pulse width
                              constraints
  -min_period                 not supported yet
  -recovery                   not supported yet
```

```
-removal                      not supported yet
  -max_skew                     not supported yet
  -clock_gating_setup           not supported yet
  -clock_gating_hold            not supported yet
  -clock_separation             not supported yet
  -include_clock_net            check DRV on clock nets
  -remove_clock_reconvergence_pessimism double(0.0)
                                check only slack less than
                                slack_cutoff for common path
                                pessimism
  -ignore_register_feedback double(0.0)
                                ignore path starts and ends at
                                the same registor
  -significant_digits integer(3)
                                number of digits after decimal
                                point
  -nosplit                      prevents line splitting if column
                                overflows
  -html                         HTML format
  -summary                      timing summary information
  -noenvironment                don't report environment variable
                                values
  -scenario string              specify working scenario
  --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 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.

53

-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.

54

-recovery
Indicates that only recovery constraint information is to be displayed. -**recovery** is a timing constraint used to describe the minimum allowable time 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.

55

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790688-89	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
report_delay_calculation Displays the actual calculation of a cell or net timing arc delay value.	Command: report_delay_calculation Report intermediate results of delay calculation
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* collection*objects*	option: -min reports mininum delay calculation -max reports maximum delay calculation -nosplit prevents line splitting if column overflows -thresholds reports the characterization thresholds -from_rise_transition double(0.0) specifies the value to be used for the from rise transition -from_fall_transition double(0.0) specifies the value to be used for the from fall transition -crosstalk reports the crosstalk information -from * the start point of a timing arc within the design -to * the end point of a timing arc within the design -of_objects * a collection of timing arcs --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

56

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.

-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**.

57

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790703	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
report_disable_timing Reports disabled timing arcs in the current design.	Command: report_disable_timing Report disabled timing arc information.
SYNTAX string **report_disable_timing** [-nosplit] [*cells_or_ports*] collection *cells_or_ports*	option: -nosplit — prevents line splitting if column overflows --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help
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.	

58

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790732	Pl's Ex. 643 (AP 07.11.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 [*: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 not supported yet -nosplit not supported yet -significant_digits integer(3) 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.	

59

-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.

60

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790746-78	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

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.

```
Command: report_noise
  report functional noise slack
```

```
option:
  -violation_only            noise violation only
  -threshold double(0.0)     reporting threshold
  --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
```


61

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

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

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

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

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

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

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

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

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

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

62

-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.

63

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790759	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
report_port Displays port information within the design.	Command: report_port [*:port_list] Report boundary port related timing information.
SYNTAX string **report_port** [-verbose] [-design_rule] [-drive] [-input_delay] [-output_delay] [-wire_load] [-nosplit] [*port_names*] list *port_names*	option: -verbose — report all port information -design_rule — report maxCap, manLoad, and maxFanout -drive — report on input and inout ports only -input_delay — report input delay -output_delay — report output delay -wire_load — report port wire load -nosplit — prevents line splitting if column overflows --get_option arg<1> — get option value --set_option ... — set option value --get_default arg<1> — get default value --set_default ... — set default value --list_options — list current option values --load_options ... — load current option values --license — list required licenses --help — display command help

64

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.

65

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790822	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

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]
[-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]
```

```
Command: report_timing [*:collection]
  Report timing path information for current design.

option:
  -skip_summary              skip DRC and wire-length
  -all                       report everything
  -global                    report global route only
  -connection                check and report wiring
                             connectivity
  -antenna                   report antenna violations
  -double_via_rate           double via rate
  -shield                    shield coverage
  -nets *                    net lists
  --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
```


66

```
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.

-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.

67

-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.

-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.

68

-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.

-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.

69

-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.

-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.

70

-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.

71

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790863-65	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

reset_path

Resets specified paths to single-cycle behavior.

```
Command: reset_path
  reset path to default timing 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
```

```
option:
  -setup                    for setup only
  -hold                     for hold only
  -rise                     rise transition only
  -fall                     fall transition only
  -from *                   from list
  -to *                     to list
  -through *                through list
  --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

-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.

72

-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**.

73

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790887-89	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

set_annotated_delay

Sets the net or cell delay value between two pins.

Command: set_annotated_delay <double:delay>
[*:object_list]

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

string*load_delay_type*
list *from_pins*
list *to_pins*
string*sdf_expression*
float *delay_value*

option:

-from *	from pins (require)
-to *	to pins (require)
-net	for net only
-cell	for cell only
-rise	for rise only
-fall	for fall only
-min	for min only
-max	for max only
-cond string	not supported yet
-load_delay string	not supported yet
-increment	increment only
-delta_only	delta only
-worst	not apply 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

-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.

74

-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.

-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.

75

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.

76

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1790893	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
set_annotated_transition Sets the transition time to be annotated on specified pins in the current design.	Command: set_annotated_transition <double:transition> <*:objects>
SYNTAX int **set_annotated_transition** [-rise][-fall][-min][-max] [-delta_only] *slew_value pin_list* float *slew_value* list *pin_list*	option: -rise — rise transition only -fall — fall transition only -min — for min mode only -max — for max mode only -delta_only — delta delay 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 -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*.	

77

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.

SYNTAX

string **set_min_library**
[-min_version *min_library*]
[-none]
max_library

string*min_library*
string*max_library*

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**.

```
Command: set_min_library <string:max_library>
  standard SDC command
```

```
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
```


78

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791081-82	Pl's Ex. 643 (AP 07.11.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 * — 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

79

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.

80

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791103	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
set_user_attribute Sets a user attribute to a specified value on an object.	Command: set_user_attribute <*:object_or_collection> <string:attr_name> <*:value> set user defined attribute
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*	option: -class string — class name of object -quiet — suppress error message --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 -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.	

81

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791121	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
sort_collection Sorts a collection based on one or more attributes, resulting in a new, sorted collection. The sort is ascending by default.	Command: sort_collection <*:collection> <*:criteria> sort a collection by the attributes
SYNTAX collection **sort_collection** [-descending] *collection1 criteria* collection *collection1* list *criteria*	option: -descending — sort in descending order --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 -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.	

82

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791125	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
swap_cell Swaps one or more cells with a new design or library cell.	Command: swap_cell <*:cell> Swap a (spare) cell to a new hierarchical name for reuse
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*	option: -rename string — new hierarchical name (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
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*.	

83

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791136	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
update_noise Performs static crosstalk noise analysis for the current design.	Command: update_noise update functional noise
SYNTAX int **update_noise** [-full]	option: --license list required licenses --help display command help
ARGUMENTS -full By default, update_noise performs the noise analysis only if the design is not up to date for noise analysis. Using **-full**, forces the update_noise to perform the noise analysis regardless whether the design is out of date or not.	

84

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791142	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
update_timing Updates timing information on the current design.	Command: update_timing Timing analysis for current design.
SYNTAX string **update_timing** [-full]	option: -opt ignore -full force timing update from scratch -inc incremental update timing -scenario string specify working scenario --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 -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.	

85

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791164	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
write_parasitics Writes out annotated parasitics information for the current design.	Command: write_parasitics <file_names ...> outputs parasitic information to a user specified file
SYNTAX Boolean **write_parasitics** -format *file_fmt* *file_name* string *file_fmt* string *file_name*	option: -use_name_map — use name map in spf file -no_coupling_cap — don't write of coupling capacitance -min — write out spf for min condition -max — write out spf for max condition -flat — write spf for flatten netlist -format para_format(SPEF) — write out parasitics para_format = DSPF | SPEF --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 -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.	

86

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791175	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44
write_sdc Writes out a script in Synopsys Design Constraints (SDC) format.	Command: write_sdc <string:file_name> Write SDC constraints to one file.
SYNTAX int **write_sdc** *file_name* [-version *sdc_version*] [-compress *compression*] [-include *categories list*] [-nosplit] string*version* string*file_name* string*compression* list *categories list*	option: -port_latency — AP defined latency in input/output port -extension — all SDC constraints including SDC extentions -latency_offset_only — Write clock offset latency 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 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.	

87

Pl's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791179-83	Pl's Ex. 643 (AP 07.11.rel.1 Executable) ZZATopTech 0000035-44

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]
[-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
```

```
Command: write_sdf <string:file_name>
  writes timing data to an SDF file

option:
  -no_cell_delays              not supported yet
  -no_timing_checks            not supported yet
  -no_net_delays               not supported yet
  -input_port_nets             not supported yet
  -output_port_nets            not supported yet
  -enabled_arcs_only           not supported yet
  -no_internal_pins            not supported yet
  -no_edge                     not supported yet
  -annotated                   not supported yet
  -no_negative_delays          not supported yet
  -significant_digits integer(3)
                               number of digits after
                               decimal point
  -levels string               not supported yet
  -instance string             not supported yet
  -context string              not supported yet
  -version string              not supported yet
  -compression string          not supported yet
  -map *                       not supported yet
  -no_edge_merging *           not supported yet
  -increment                   delta delay
  --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
```


88

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.

-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

89

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.

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.

90

• 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 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

91

PI's Ex. 335 (PrimeTime 2006.12) at ATopTech 1791189-92	PI's Ex. 643 (AP 07.11.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 [*: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* string*header_file_name* float *delay* string*v1name* float *v1* string*v0name* float *v0* float *margin_value* float *trans* string*paths* string*file_name* string*spice_sub_circuit_file* unsigned*number_of_points* float *num* unsigned*precision* float *tran_size* float *tran_step* int *number_of_samples*	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

92

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.

-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 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.

93

-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 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

94

-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.

95

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. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param db collection_result_display_limit 100
#             Type    : int (cannot_save)
#             Usage   : max number of objects in
                        collection to display in shell
```

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. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta default_oc_per_lib false
#             Type    : bool(persistent)
#             Usage   : Determines whether use default
                        operating condition per each
                        individual library
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793666	Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

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**.

```
set_param ta disable_case_analysis false
#             Type    : bool(persistent)
#             Usage   : Disables or enables case analysis
                        in timing analysis
```

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

rc_degrade_min_slew_when_rd_less_than_rnet

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

TYPE

Boolean

DEFAULT

false

DESCRIPTION

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

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

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

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

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

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

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

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

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param rom rc_degrade_min_slew_when_rd_less_than_rnet
false
#             Type    : bool
#             Usage   : min degrade variable
```

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

si_enable_analysis

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

TYPE

Boolean

DEFAULT

false

DESCRIPTION

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

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

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

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

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

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta si_enable_analysis false
#             Type    : bool(persistent)
#             Usage   : enable crosstalk noise and
                        induced-delay analysis
```

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

si_filter_accum_aggr_noise_peak_ratio

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

TYPE

float

DEFAULT

0.03

DESCRIPTION

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

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

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

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

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 filter_per_aggr_noise_peak_ratio 0.010
#             Type    : float              (persistent)
#             Usage   : ignore aggressor if noise bump
                        peak to supply voltage ratio is
                        less than the value
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793821

si_filter_per_aggr_noise_peak_ratio

Specifies the threshold for the voltage bump introduced by an aggressor at a victim node, divided by Vcc, below which the aggressor net can be filtered out during electrical filtering.

TYPE

float

DEFAULT

0.01

DESCRIPTION

Specifies the threshold for the voltage bump introduced by an aggressor at a victim node; the default is 0.01. This variable, along with **si_filter_accum_aggr_noise_peak_ratio**, makes up a pair of variables used by PrimeTime-SI during the electrical filtering phase, to determine whether an aggressor net can be filtered.

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

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

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

Parasitic filtering criteria previously checked are controlled by the variables **si_filter_per_aggr_xcap**, **si_filter_per_aggr_xcap_to_gcap_ratio**, and **fi_filter_single_to_average_all_xcap_ratio**.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 filter_per_aggr_noise_peak_ratio 0.010
#     Type    : float          (persistent)
#     Usage   : ignore aggressor if noise bump peak to supply
      voltage ratio is less than the value
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793858

si_xtalk_exit_on_max_iteration_count

Specifies a maximum number of incremental timing iterations, after which PrimeTime-SI exits the analysis loop.

TYPE

integer

DEFAULT

2

DESCRIPTION

Specifies a maximum number of incremental timing iterations. PrimeTime-SI exits the analysis loop after performing this number of iterations.

The default value of this variable is 2, meaning that PrimeTime-SI exits the analysis loop after performing two iterations. You can override this default by setting the variable to another integer; the minimum allowed value is 1.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si2 xtalk_exit_on_max_iteration_count 3
#             Type    : uint
#             Usage   : maximum number of iterations to
                        compute crosstalk-induced delay
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793872

si_xtalk_reselect_max_mode_slack

Specifies the max mode pin slack threshold, below which PrimeTime-SI reselects a net for subsequent delay calculations.

TYPE

float

DEFAULT

0

DESCRIPTION

This variable specifies the pin slack threshold in the max mode. Nets that have at least one pin with a max mode slack below this threshold are selected for the next iteration of PrimeTime-SI delay calculations. Max-mode pin slack is the slack of the worst max-mode (setup) path through the pin.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay
si_xtalk_reselect_delta_delay_ratio
si_xtalk_reselect_min_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 xtalk_reselect_max_mode_slack 0.000
#             Type    : float              (persistent)
#             Usage   : only do window-filtering on the
                        victim nets whose set up slack is
                        worse than the value
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793873

si_xtalk_reselect_min_mode_slack

Specifies the min mode pin slack threshold, below which PrimeTime-SI reselects a net for subsequent delay calculations.

TYPE

float

DEFAULT

0

DESCRIPTION

This variable specifies the pin slack threshold in the min mode. Nets that have at least one pin with a min mode slack below this threshold are selected for the next iteration of PrimeTime-SI delay calculations. Min-mode pin slack is the slack of the worst min-mode (hold) path through the pin.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay
si_xtalk_reselect_delta_delay_ratio
si_xtalk_reselect_max_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 xtalk_reselect_min_mode_slack 0.000
#             Type    : float              (persistent)
#             Usage   : only do window-filtering on the
                        victim nets whose hold slack is
                        worse than the value
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793884

timing_clock_reconvergence_pessimism

Select signal transition sense matching for computing clock reconvergence pessimism removal.

TYPE

string

DEFAULT

normal

DESCRIPTION

Determines how the value of the clock reconvergence pessimism removal (crpr) is computed with respect to transition sense. Allowed values are *normal* (the default) and *same_transition*.

When set to *normal*, the crpr value is computed even if the clock transitions to the source and destination latches are in different directions on the common clock path. It is computed separately for rise and fall transitions and the value with smaller absolute value is used.

When set to *same_transition*, the crpr value is computed only when the clock transition to the source and destination latches have a common path and the transition is in the same direction on each pin of the common path. Thus if the source and destination latches are triggered by different edge types, crpr has a value of zero.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_clock_reconvergence_pessimism normal
#             Values:  normal same_transition
              (persistent)
#             Usage  : Determines method to choose
                       transition sense for removing clock
                       reconvergence pessimism
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793889

timing_crpr_threshold_ps

Specifies amount of pessimism that clock reconvergence pessimism removal (CRPR) is allowed to leave in the report.

TYPE

float

DEFAULT

20

DESCRIPTION

Specifies amount of pessimism that clock reconvergence pessimism removal (CRPR) is allowed to leave in the report. The unit is in pico seconds (ps), regardless of the units of the main library.

The threshold is per reported slack: setting the this variable to the *TH1* value means that reported slack is no worse than *S* - *TH1*, where *S* is the reported slack when **timing_crpr_threshold_ps** is set close to zero (the minimum allowed value is **1** picosecond).

The variable has no effect if CRPR is not active (**timing_remove_clock_reconvergence_pessimism** is false). The larger the value of **timing_crpr_threshold_ps**, the faster the runtime when CRPR is active. The recommended setting is about one half of the stage (gate plus net) delay of a typical stage in the clock network. It provides a reasonable trade-off between accuracy and runtime in most cases. You may want to use different settings throughout the design cycle: larger during the design phase, smaller for sign-off. You might have to experiment and set a different value when moving to a different technology.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_crpr_threshold_ps 20.000
#             Type    : float   [range: 1.000 ..
                        10000002004087734272.000]
                        (persistent)
#             Usage   : Defines the threshold for crpr
                        common points grouping
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793890

timing_disable_bus_contention_check

Disable checking for timing violations resulting from transient contention on design busses.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

Applies only to bus designs that have multiple three-state drivers.

When *true*, PrimeTime ignores timing setup and hold (max and min) violations that occur as a result of transient bus contention. When *false* (the default), PrimeTime reports these timing violations.

Bus contention occurs when more than one driver is enabled at the same time. By default, PrimeTime treats the bus as if it is in an unknown state during this region of contention, and reports a timing violation if the setup and hold regions extend into the contention region. Note that checking is done only for timing violations, and not for logical and excessive power dissipation violations, which are outside the scope of static timing analysis tools.

Set this variable to *true* only if you are certain that transient bus contention regions will never occur. By setting the value to *true*, you guarantee that on a multi-driven three-state bus, the drivers in the previous clock cycle are disabled before the drivers in the current clock cycle are enabled. If you set this variable to *true*, you must ensure that the variable **timing_disable_bus_contention_check** is *false*. The variables **timing_disable_bus_contention_check** and **timing_disable_floating_bus_check** cannot both be *true* at the same time.

During the switching between the high-impedance (Z) state and the high/low state, the timing behavior (for example, intrinsic delay) of three-state buffers is captured in the Synopsys library using the timing arc types three_state_disable and three_state_enable. These timing arcs connect the enable pin to the output pin of the three-state buffers. For details, see the *Library Compiler Reference Manual*.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_disable_bus_contention_check false
#             Type    : bool(persistent)
#             Usage   : Disables or enables timing check
                        for contention buses in design.
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793891	Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

timing_disable_clock_gating_checks

Disable checking for setup and hold clock gating violations.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, disables clock-gating setup and hold checks. When *false* (the default), PrimeTime automatically determines clock-gating and performs clock-gating setup and hold checks.

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

```
set_param ta timing_disable_clock_gating_checks false
#             Type    : bool(persistent)
#             Usage   : Disables or enables clock gating
                        check in timing analysis
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793893

timing_disable_floating_bus_check

Disable checking for timing violations resulting from transient floating design buses.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

Applies only to bus designs that have multiple three-state drivers.

When *true*, PrimeTime ignores timing setup and hold (max and min) violations that occur as a result of transient floating buses. When *false* (the default), PrimeTime reports these timing violations.

Floating bus condition occurs when no driver controls the bus at a given time. By default, PrimeTime treats the bus as if it is in an unknown state during this region of contention, and reports a timing violation if the setup and hold regions extend into the floating region. Note that checking is done only for timing violations, and not for logical violations, which are outside the scope of static timing analysis tools.

Set this value to *true* only if you are certain that transient floating bus regions will never occur. By setting the value to *true*, you guarantee that on a multi-driven three-state bus, the drivers in the previous clock cycle are disabled before the new drivers in the current clock cycle are enabled. If you set this variable to *true*, you must ensure that the variable **timing_disable_bus_contention_check** is *false*. The variables **timing_disable_floating_bus_check** and **timing_disable_bus_contention_check** cannot both be *true* at the same time.

During the switching between the high-impedance (Z) state and the high/low state, the timing behavior (for example, intrinsic delay) of three-state buffers is captured in the Synopsys library using the timing arc types three_state_disable and three_state_enable. These timing arcs connect the enable pin to the output pin of the three-state buffers. For details, see the *Library Compiler Reference Manual*.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_disable_floating_bus_check false
#             Type    : bool(persistent)
#             Usage   : Disables or enables timing check
                        for floating buses in design.
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793896	Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

timing_disable_recovery_removal_checks

Disable or enable the timing analysis of recovery and removal checks in the design.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, disables recovery and removal timing analysis. When *false* (the default), PrimeTime performs recovery and removal checks; for descriptions of these checks, see the man page for the **report_constraint** command.

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

```
set_param ta timing_disable_recovery_removal_checks false
#             Type    : bool(persistent)
#             Usage   : Disables or enables recovery and
                        removal checks in timing analysis.
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793906	Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

timing_enable_preset_clear_arcs

Controls whether PrimeTime enables or disables preset and clear arcs.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, permanently enables asynchronous preset and clear timing arcs, so that you use them to analyze timing paths. When *false* (the default), PrimeTime disables all preset and clear timing arcs.

Note that if there are any minimum pulse width checks defined on asynchronous preset and clear pins they are performed regardless of the value of this variable. Also note the the -true and the -justify options of report_timing cannot be used unless this variable is at its default value.

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

```
set_param ta timing_enable_preset_clear_arcs false
#             Type    : bool(persistent)
#             Usage   : Enables or disables preset and
                        clear timing arcs checks in timing
                        analysis.
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793911

timing_input_port_default_clock

Determines whether a default clock is assumed at input ports for which the user has not defined a clock with set_input_delay.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

This Boolean variable affects the behavior of PrimeTime when the user sets an input delay without a clock on an input port. When *true* (the default value), the input delay on the port is set with respect to one imaginary clock so that the inputs are constrained. This also causes the clocks along the paths driven by these input ports to become related. When *false*, no such imaginary clock is assumed.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_input_port_default_clock true
#             Type    : bool(persistent)
#             Usage   : Determines whether analyze timing
                        for paths which start from input
                        port with default clock
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793921-22

timing_remove_clock_reconvergence_pessimism

Enables or disables clock reconvergence pessimism removal

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When this variable is set to *true*, PrimeTime removes clock reconvergence pessimism from slack calculation and minimum pulse width checks. This variable replaces the following discontinued options:

-report_clock_reconvergence_pessimism
-remove_clock_reconvergence_pessimism

of the **report_timing**, **report_constraint**, and **get_timing_paths** commands.

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

CRP is independently calculated for rise and fall clock paths. You can use the variable **timing_clock_reconvergence_pessimism** to control CRP calculation with respect to transition sense. In the case of the capturing device being a level-sensitive latch two CRP values will be caculated:
• crp_open, which is the CRP corresponding to the opening edge of the latch
• crp_close, which is the CRP corresponding to the closing edge of the latch The required time at the latch will be increased by the value of crp_open and hence reduce the amount of borrowing (if any) at the latch. Meanwhile, the maximum time borrow allowed at the latch is affected by shifting the closing edge by crp_close. For more details, see the *PrimeTime User Guide: Fundamentals*.

For a more detailed description of a CRP calulation, use the **report_crpr** command.

CRP is calculated differently for minimum pulse-width checks. It is given as the minimum of (maximum rise arrival time - minimum rise arrival time) and (maximum fall arrival time - minimum fall arrival time) at the pin where the check is being made.

If the variable **si_enable analysis** is set to *true* delays in the clock network may also include delta delays resulting from crosstalk interaction. Such delays are dynamic in nature, that is, they may vary from one clock cycle to the next, causing different delay variations (either speed-up or slow-down) on the same network, but during different clock cycles.

Starting with U-2003.03 release PrimeTime only considers SI delta delays as part of the CRP calculation if the type of timing check deployed derives its data from the same clock cycle.

In transparent-latch based designs, it is recommended that the variable **timing_early_launch_at_borrowing_latches** should be set to *false* when CRP removal (CRPR) is enabled. In this case, CRPR will apply even to paths whose startpoints are borrowing, leading to better pessimism reduction overall.

Any effective change in the value of the **timing_remove_clock_reconvergence_pessimism** variable causes full update_timing. You cannot perform one report_timing operation that considers CRP and one that does not without full update_timing in between.

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param ta timing_remove_clock_reconvergence_pessimism
false
#             Type   : bool(persistent)
#             Usage  : Enables or disables clock
                       reconvergence pessimism in timing
                       analysis
```

For backward compatibility, the discontinued options will appear for the first few releases after they are obsoleted. However, if the design is not up to date at the time they are executed, they will only set timing_remove_clock_reconvergence_pessimism to *true*

If the design is up to date, then the command with the discontinued option fails. Since the discontinued command options only set timing_remove_clock_reconvergence_pessimism to *true*, the **-report_clock_reconvergence_pessimism** option behavior is not backward compatible. It causes slack to be removed prior to selecting the worst path. In other words, it behaves the same as the discontinued **-remove_clock_reconvergence_pessimism** option of the **report_timing**, **reyport_constraint**, and **get_timing_paths** commands. As soon as possible, update your scripts to set the **timing_remove_clock_reconvergence_pessimism** variable to *true* instead of using the discontinued options.

Limitations: CRPR does not support paths that fan out directly from clock source pins to the data pins of sequential devices. To enable support for such paths the variable **timing_crpr_remove_clock_to_data_crp** must be set to TRUE. CRPR does not support ideal clock latency set on pins or ports. CRPR does not support propagated clocks set on pins or ports as opposed to clock objects.

To turn CRP removal on:

```
   pt_shell set timing_remove_clock_reconvergence_pessimism TRUE
TRUE
pt_shell report_timing
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793854

si_xtalk_exit_on_coupled_reevaluated_nets_pct

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of coupled nets, below which PrimeTime-SI exits the analysis loop.

TYPE

float

DEFAULT

0

DESCRIPTION

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of coupled nets. PrimeTime-SI exits the analysis loop after completing the current iteration, when the percentage of nets selected for reevaluation in the next iteration is less than this number. The number of coupled nets is based on detailed parasitics as read in by **read_parasitics**. That is, crosstalk filtering does not impact the count of coupled nets for the purpose of this variable. The number of coupled nets counts all individual net segments in the same way that [get_nets -hierarchical *] counts all nets in the design.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable. the analysis loop.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 xtalk_exit_on_coupled_reevaluated_nets_pct 0.000
#     Type    : float
#     Usage   :
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793863

si_xtalk_exit_on_number_of_reevaluated_nets

Specifies a maximum number of nets selected for reevaluation, below which PrimeTime-SI exits the analysis loop.

TYPE

integer

DEFAULT

0

DESCRIPTION

Specifies a maximum number of nets selected for reevaluation. PrimeTime-SI exits the analysis loop after completing the current iteration, when the number of nets selected for reevaluation in the the next iteration is less than this number.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 xtalk_exit_on_coupled_reevaluated_nets_pct
0.000
#             Type    : float
#             Usage   :
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793865

si_xtalk_exit_on_reevaluated_nets_pct

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of nets, below which PrimeTime-SI exits the analysis loop.

TYPE

float

DEFAULT

0

DESCRIPTION

Specifies a maximum percentage of nets reselected for evaluation, relative to the total number of nets. PrimeTime-SI exits the analysis loop after completing the current iteration, when the percentage of nets selected for reevaluation in the next iteration is less than this number.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

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

Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44

```
set_param si1 xtalk_exit_on_reevaluated_nets_pct 0.000
#             Type    : float
#             Usage   :
```

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002587-89			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the cell Object Class			
area	float	The area of a cell. If the cell is hierarchical, this includes net area.	area : double (read-only)
base_name	string	The leaf name of a cell. For example, the base_name of cell U1/U2/U3 is U3.	base_name : string (read-only)
dont_touch	boolean	Identifies cells to be excluded from optimization in Design Compiler. Cells with the dont_touch attribute set to true are not modified or replaced during compilation in Design Compiler. Setting dont_touch on a hierarchical cell sets the attribute on all cells below it. Set with set_dont_touch, and used by characterize_context and create_timing_context. You can set and unset the dont_touch attribute.	dont_touch : bool
full_name	string	The complete name of a cell. For example, the full name cell U3 within cell U2 within cell U1 is U1/U2/U3. The full_name attribute is not affected by current_instance.	full_name : string (read-only)
is_sequential	boolean	A cell is sequential if it is not combinational.	is_sequential : bool (read-only)
number_of_pins	integer	Number of pins on the cell. The number of pins can be different before and after linking. For example, if some pins were unconnected in a Verilog instance, after linking to the lower-level design, additional pins can be created on the cell.	number_of_pins : uint (read-only)

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011				PI's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the cell Object Class				
is_clock_gating_check	cell	boolean	A	is_clock_gating_check : bool (read-only)
ref_name	cell	string	A	ref_name : string (read-only)

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002592-95			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the clock Object Class			
`full_name`	`string`	The name of the clock. This is set with `create_clock`. It is either the name given with the `-name` option, or the name of the first object to which the clock is attached. Once set, this attribute is read only.	`full_name : string (read-only)`
`period`	`float`	The clock period (or cycle time) is the shortest time during which the clock waveform repeats. For a simple waveform with one rising and one falling edge, the period is the difference between successive rising edges. Set with `create_clock -period`.	`period : float`
`propagated_clock`	`boolean`	Specifies that clock latency (insertion delay) be determined by propagating delays from the clock source to destination register clock pins. If this attribute is not present, ideal clocking is assumed. Set with `set_propagated_clock`.	`propagated_clock : bool`
`sources`	`string`	This is a collection of the source pins or ports of the clock. The sources are defined with the `create_clock` command.	`sources : collection (read-only)`

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the clock Object Class	
`is_generated clock boolean A`	`is_generated : bool`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002605-06			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_cell Object Type			
area	float	A floating-point value representing the area of a library cell.	area : double (read-only)
base_name	string	The name of a library cell. For example, the base_name of library cell tech1/AN2 is AN2.	base_name : string (read-only)
dont_touch	boolean	Identifies library cells to be excluded from optimization. Values are true (the default) or false. Library cells with the dont_touch attribute set to true are not modified or replaced during compile. Set in Design Compiler with set_dont_touch.	dont_touch : bool
full_name	string	The fully qualified name of a library cell. This is the name of the library followed by the library cell name. For example, the full_name of library cell AN2 in library tech1 is tech1/AN2.	full_name : string (read-only)
is_sequential	boolean	This attribute is true if the library cell is sequential.	is_sequential : bool (read-only)

Pl's Ex.27 (PrimeTime 2010.06) at SYNPS-00006973			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_cell Object Type			
`is_level_shifter`	`Boolean`	This attribute is `true` if the library cell is a level shifter cell.	`is_level_shifter : bool (read-only)`

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_cell Object Type	
`dont_use lib_cell boolean A`	`dont_use : bool`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002607-11			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_pin Object Type			
`base_name`	`string`	The leaf name of the library cell pin. For example, the base_name of tech1/AN2/Z is Z.	`base_name : string (read-only)`
`full_name`	`string`	The fully qualified name of a library cell pin. This is the name of the library followed by the library cell name followed by a pin name. For example, the full_name of pin Z on library cell AN2 in library tech1 is tech1/AN2/Z.	`full_name : string (read-only)`
`pin_capacitance`	`float`	A floating-point value representing the capacitance of a library pin.	`pin_capacitance : float (read-only)`

Pl's Ex.27 (PrimeTime 2010.06) at SYNPS-00006976			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_pin Object Type			
`is_pad`	`Boolean`	This attribute is `true` if the library pin is a pad. See the Library Compiler documentation.	`is_pad : bool (read-only)`

Pl's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the lib_pin Object Type	
`pin_direction lib_pin string A`	`pin_direction : in out inout internal unknown (read-only)`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002611-14			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44	
Attributes of the net Object Type				
base_name	string	The leaf name of a net. For example, the base name of net i1/i1z1 is i1z1. You cannot set this attribute.	base_name : string	(read-only)
dont_touch	boolean	Identifies nets to be excluded from optimization in Design Compiler. Values are true (the default) or false. Nets with the dont_touch attribute set to true are not modified or replaced during compile with Design Compiler. Set with set_dont_touch.	dont_touch : bool	
full_name	string	The complete name of a net. For example, the full_name of net i1z1 within cell i1 is i1/i1z1. The full_name attribute is not affected by current instance. The full_name attribute is read-only.	full_name : string	(read-only)
total_capacitance_max	float	A floating-point value representing the sum of all pin capacitances and the wire capacitance of a net for maximum conditions. You cannot set this attribute.	total_capacitance_max : float	(read-only,application)
total_capacitance_min	float	A floating-point value representing the sum of all pin capacitances and the wire capacitance of a net for minimum conditions. You cannot set this attribute.	total_capacitance_min : float	(read-only,application)

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002616-25			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the pin Object Class			
actual_fall_transition_max	float	A floating-point value representing the largest falling transition time for a pin.	actual_fall_transition_max : float (read-only,application)
actual_fall_transition_min	float	A floating-point value representing the smallest falling transition time for a pin.	actual_fall_transition_min : float (read-only,application)
actual_rise_transition_max	float	A floating-point value representing the largest rising transition time for a pin.	actual_rise_transition_max : float (read-only,application)
actual_rise_transition_min	float	A floating-point value representing the smallest rising transition time for a pin.	actual_rise_transition_min : float (read-only,application)
clocks	string	The collection of clock objects which propagate through this pin. It is undefined if no clocks are present.	clocks : collection (read-only,application)
direction	string	The direction of a pin. Value can be in, out, inout, or internal. The pin_direction attribute is a synonym for direction. Directions can change as a result of linking a design, as references are resolved.	direction : in out inout internal unknown (read-only)
full_name	string	The complete name of a pin to the top of the hierarchy. For example, the full name of pin Z on cell U2 within cell U1 is U1/U2/Z. The setting of the current instance has no effect on the full name of a pin. See also the lib_pin_name attribute.	full_name : string (read-only)
is_three_state	boolean	This attribute is true if a pin is a three-state driver.	is_three_state : bool (read-only)

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the pin Object Class	
`is_clock_gating_clock pin boolean A`	`is_clock_gating_clock : bool (read-only)`
`is_clock_gating_enable pin boolean A`	`is_clock_gating_enable : bool (read-only)`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002630-42			Pl's Ex. 643 (AP 07.11.rel.1 Executable) at ZZATopTech 0000035-44
Attributes of the port Object Class			
`direction`	`string`	The direction of a port. Value can be in, out, inout, or internal. The `port_direction` attribute is a synonym for direction. You cannot set this attribute.	`direction : in out inout internal unknown`
`full_name`	`string`	The name of a port. You cannot set this attribute.	`full_name : string (read-only)`