

OpenSTA

Command Line Arguments

The command line arguments for sta are shown below.

```
sta [-help] [-version] [-no_init]
  -help          show help and exit
  -version       show version and exit
  -no_init       do not read ~/.sta
  -no_splash    do not print the splash message
  -threads count|max use count threads
  -exit          exit after reading cmd_file
  cmd_file      source cmd_file
```

When the STA starts up commands are first read from the user initialization file `~/.sta` if it exists. If a TCL command file `cmd_file` is specified, commands are read from the file and executed before entering an interactive TCL command interpreter. If `-exit` is specified the application exits after reading `cmd_file`. Use the TCL `exit` command to exit the application. The `-threads` option specifies how many parallel threads to use. Use `-threads max` to use one thread per processor.

A sample command file that reads a library and a Verilog netlist and reports timing checks is shown below.

```
read_liberty example1_slow.lib
read_verilog example1.v
link_design top
read_sdf example1.sdf
create_clock -name clk -period 10 {clk1 clk2 clk3}
set_input_delay -clock clk 0 {in1 in2}
report_checks
```

This example can be found in `examples/example1.tcl`.

Any number of Liberty and Verilog files can be read before linking the design. After linking the design, SDF or parasitics can be read.

An example command script using three process corners and +/-10% min/max derating is shown below.

```
define_corners wc typ bc
read_liberty -corner wc example1_slow.lib
read_liberty -corner typ example1_typ.lib
read_liberty -corner bc example1_fast.lib
read_verilog example1.v
link_design top
set_timing_derate -early 0.9
set_timing_derate -early 1.1
create_clock -name clk -period 10 {clk1 clk2 clk3}
set_input_delay -clock clk 0 {in1 in2}
report_checks -path_delay min_max
report_checks -corner typ
```

This example can be found in examples/example5.tcl. Other examples can be found in the directory examples.

TCL Interpreter

Keyword arguments to commands may be abbreviated. For example,

```
report_checks -unique
```

is equivalent to the following command.

```
report_checks -unique_paths_to_endpoint
```

Commands

all_clocks

The **all_clocks** command returns a list of all clocks that have been defined.

all_inputs

The **all_inputs** command returns a list of all input and bidirect ports of the current design.

all_outputs

The **all_outputs** command returns a list of all output and bidirect ports of the design.

all_registers	<code>[-clock <i>clock_names</i>] [-cells] [-data_pins] [-clock_pins] [-async_pins] [-output_pins] [-level_sensitive] [-edge_triggered]</code>
<i>clock_names</i>	A list of <i>clock</i> names. Only registers clocked by these clocks are returned.
-cells	Return a list of register instances.
-data_pins	Return the register data pins.
-clock_pins	Return the register clock pins.

<code>-async_pins</code>	Return the register set/clear pins.
<code>-output_pins</code>	Return the register output pins.
<code>-level_sensitive</code>	Return level-sensitive latches.
<code>-edge_triggered</code>	Return edge-triggered registers.

The `all_registers` command returns a list of register instances or register pins in the design. Options allow the list of registers to be restricted in various ways. The `-clock` keyword restricts the registers to those that are clocked by a set of clocks. The `-cells` option returns the list of registers or latches (the default). The `-data_pins`, `-clock_pins`, `-async_pins` and `-output_pins` options cause `all_registers` to return a list of register pins rather than instances.

<code>check_setup</code>	<code>[-verbose]</code> <code>[-unconstrained_endpoints]</code> <code>[-multiple_clock]</code> <code>[-no_clock]</code> <code>[-no_input_delay]</code> <code>[-loops]</code> <code>[-generated_clocks]</code> <code>[> <i>filename</i>]</code> <code>[>> <i>filename</i>]</code>
<code>-verbose</code>	Show offending objects rather than just error counts.
<code>-unconstrained_endpoints</code>	Check path endpoints for timing constraints (timing check or <code>set_output_delay</code>).
<code>-multiple_clock</code>	Check register/latch clock pins for multiple clocks.
<code>-no_clock</code>	Check register/latch clock pins for a clock.
<code>-no_input_delay</code>	Check for inputs that do not have a <code>set_input_delay</code> command.
<code>-loops</code>	Check for combinational logic loops.
<code>-generated_clocks</code>	Check that generated clock source pins have been defined as clocks.

The `check_setup` command performs sanity checks on the design. Individual checks can be performed with the keywords. If no check keywords are specified all checks are performed.

connect_pin	<i>net</i> <i>port pin</i>
<i>net</i>	A net to add connections to.
<i>port</i>	A port to connect to <i>net</i> .
<i>Pin</i>	A pin to connect to <i>net</i> .

The `connect_pin` command connects a port or instance pin to a net.

create_clock	<i>-period period</i> [-name <i>clock_name</i>] [-waveform <i>edge_list</i>] [-add] [<i>pin_list</i>]
<i>period</i>	The clock period.
<i>clock_name</i>	The name of the clock.
<i>edge_list</i>	A list of edge rise and fall time.
<i>-add</i>	Add this clock to the clocks on <i>pin_list</i> .
<i>pin_list</i>	A list of pins driven by the clock.

The `create_clock` command defines the waveform of a clock used by the design.

If no *pin_list* is specified the clock is *virtual*. A virtual clock can be referred to by name in input arrival and departure time commands but is not attached to any pins in the design.

If no clock name is specified the name of the first pin is used as the clock name.

If a waveform is not specified the clock rises at zero and falls at half the clock period. The waveform is a list with time the clock rises as the first element and the time it falls as the second element.

If a clock is already defined on a pin the clock is redefined using the new clock parameters. If multiple clocks drive the same pin, use the `-add` option to prevent the existing definition from being overwritten.

The following command creates a clock with a period of 10 time units that rises at time 0 and falls at 5 time units on the pin named `clk1`.

```
create_clock -period 10 clk1
```

The following command creates a clock with a period of 10 time units that is high at time zero, falls at time 2 and rises at time 8. The clock drives three pins named `clk1`, `clk2`, and `clk3`.

```
create_clock -period 10 -waveform {8 2} -name clk {clk1 clk2 clk3}
```

create_generated_clock [-name *clock_name*]
-source *master_pin*
[-master_clock *master_clock*]
[-pll_out *pll_out_pin*]
[-pll_feedback *pll_fdbk_pin*]
[-divide_by *divisor*]
[-multiply_by *multiplier*]
[-duty_cycle *duty_cycle*]
[-invert]
[-edges *edge_list*]
[-edge_shift *shift_list*]
[-add]
pin_list

clock_name The name of the generated clock.

master_pin A pin in the fanout of the master clock that is the source of the generated clock.

master_clock The master clock used to generate the clock waveform.

pll_out_pin The pin from *pin_list* that is the phase locked loop output pin.

pll_fdbk_pin A pin in the fanout of the *pll_out_pin* that the phased locked loop phase locks to the master clock pin.

divisor Divide the master clock period by *divisor*.

multiplier Multiply the master clock period by *multiplier*.

duty_cycle The percent of the period that the generated clock is high (between 0 and 100).

-invert Invert the master clock.

edge_list

Not supported.

-add Add this clock to the clocks on *pin_list*.

pin_list A list of pins driven by the clock.

The `create_generated_clock` command is used to generate a clock from an existing clock definition. It is used to model clock generation circuits such as clock dividers and phase locked loops.

The `-source`, `-pll_out` and `-pll_feedback` must all be pins on the same PLL instance. The delay between the PLL out and feedback pins is removed from the source latency of the generated clock.

The `-divide_by`, `-multiply_by` and `-edges` arguments are mutually exclusive.

The `-multiply_by` option is used to generate a higher frequency clock from the source clock. The period of the generated clock is divided by *multiplier*. The clock *multiplier* must be a positive integer. If a duty cycle is specified the generated clock rises at zero and falls at period * *duty_cycle* / 100. If no duty cycle is specified the source clock edge times are divided by *multiplier*.

The `-divide_by` option is used to generate a lower frequency clock from the source clock. The clock *divisor* must be a positive integer. If the clock divisor is a power of two the source clock period is multiplied by *divisor*, the clock rise time is the same as the source clock, and the clock fall edge is one half period later. If the clock divisor is not a power of two the source clock waveform edge times are multiplied by *divisor*.

The `-edges` option forms the generated clock waveform by selecting edges from the source clock waveform.

If the `-invert` option is specified the waveform derived above is inverted.

If a clock is already defined on a pin the clock is redefined using the new clock parameters. If multiple clocks drive the same pin, use the `-add` option to prevent the existing definition from being overwritten.

In the example show below generates a clock named `gclk1` on register output pin `r1/Q` by dividing it by four.

```
create_clock -period 10 -waveform {1 8} clk1
create_generated_clock -name gclk1 -source clk1 -divide_by 4 r1/Q
```

The generated clock has a period of 40, rises at time 1 and falls at time 21.

In the example shown below the duty cycle is used to define the derived clock waveform.

```
create_generated_clock -name gclk1 -source clk1 -duty_cycle 50 \
                      -multiply_by 2 r1/Q
```

The generated clock has a period of 5, rises at time .5 and falls at time 3.

In the example shown below the first, third and fifth source clock edges are used to define the derived clock waveform.

```
create_generated_clock -name gclk1 -source clk1 -edges {1 3 5} r1/Q
```

The generated clock has a period of 20, rises at time 1 and falls at time 11.

```
create_voltage_area      [-name name]
                        [-coordinate coordinates]
                        [-guard_band_x guard_x]
                        [-guard_band_y guard_y]
                        cells
```

This command is parsed and ignored by timing analysis.

current_design *[design]*

current_instance *[instance]*

instance Not supported.

define_corners *corner_name1 [corner_name2]...*

corner_name The name of a delay calculation corner.

Use the `define_corners` command to define the names of multiple process/temperature/voltage corners. The `define_corners` command must follow `set_operating_conditions -analysis_type` and precede any reference to the corner names and can only appear once in a command file. There is no support for re-defining corners.

For analysis type `single`, each corner has one delay calculation result and early/late path arrivals. For analysis type `best_case/worst_case` and `on_chip_variation`, each corner has min/max delay calculation results and early/late path arrivals.

delete_clock *[-all] clocks*

clocks A list of clocks to remove.

delete_from_list *list objects*

list A list of objects.

objects A list of objects to delete from *list*.

delete_generated_clock *[-all] clocks*

clocks A list of generated clocks to remove.

delete_instance *instance*

instance An instance to delete.

The network editing command `delete_instance` removes an instance from the design.

delete_net *net*

net A nets to delete.

The network editing command `delete_net` removes a net from the design.

disconnect_pin *net*
port|pin|-all

net The net to disconnect pins from.

port A port to connect to *net*.

pin A pin to connect to *net*.

-all Disconnect all pins from the net.

Disconnects a port or pin from a net. Parasitics connected to the pin are deleted.

elapsed_run_time

Returns the total clock run time in seconds as a float.

find_timing_paths	<ul style="list-style-type: none"> [-from <i>from_list</i> <ul style="list-style-type: none"> -rise_from <i>from_list</i> -fall_from <i>from_list</i> [-through <i>through_list</i> <ul style="list-style-type: none"> -rise_through <i>through_list</i> -fall_through <i>through_list</i> [-to <i>to_list</i> <ul style="list-style-type: none"> -rise_to <i>to_list</i> -fall_to <i>to_list</i> [-unconstrained] [-path_delay <i>min min_rise min_fall</i> <ul style="list-style-type: none"> <i>max max_rise max_fall</i> <i>min_max</i> [-group_count <i>path_count</i>] [-endpoint_count <i>endpoint_path_count</i>] [-unique_paths_to_endpoint] [-corner <i>corner_name</i>] [-slack_max <i>max_slack</i>] [-slack_min <i>min_slack</i>] [-sort_by_slack] [-path_group <i>group_names</i>]
<i>from_list</i>	A list of clocks, instances, ports or pins.
<i>through_list</i>	A list of instances, pins or nets.
<i>to_list</i>	A list of clocks, instances, ports or pins.
-unconstrained	Return unconstrained paths.
-path_delay min	Report min path (hold) checks.
-path_delay min_rise	Report min path (hold) checks for rising endpoints.
-path_delay min_fall	Report min path (hold) checks for falling endpoints.
-path_delay max	Report max path (setup) checks.
-path_delay max_rise	Report max path (setup) checks for rising endpoints.
-path_delay max_fall	Report max path (setup) checks for falling endpoints.
-path_delay min_max	Report max and max path (setup and hold) checks.

<i>path_count</i>	The number of paths to report in each path group.
<i>endpoint_path_count</i>	The number of paths to report for each endpoint.
-unique_paths_to_endpoint	Report multiple paths to an endpoint that traverse different pins without showing multiple paths with different rise/fall transitions.
<i>corner_name</i>	Only report paths for one process corner.
<i>max_slack</i>	Only report paths with less slack than <i>max_slack</i> .
<i>min_slack</i>	Only report paths with more slack than <i>min_slack</i> .
-sort_by_slack	Sort paths by slack rather than slack grouped by path group.
<i>group_names</i>	List of path group names to report. All path groups are reported if this option is not specified.

The `find_timing_paths` command returns a list of path objects for scripting. Use the `get_property` function to access properties of the paths.

get_cells	<ul style="list-style-type: none"> [-hierarchical] [-hsc <i>separator</i>] [-filter <i>expr</i>] [-regexp] [-nocase] [-quiet] [-of_objects <i>objects</i>] [<i>patterns</i>]
-hierarchical	Searches hierarchy levels below the current instance for matches.
<i>separator</i>	Character to use to separate hierarchical instance names in <i>patterns</i> .
<i>expr</i>	<p>A filter expression of the form</p> <p style="padding-left: 40px;"><i>attribute</i> == ~= <i>pattern</i></p> <p>where <i>attribute</i> is an attribute supported by the <code>get_property</code> command.</p>

objects The name of a pin or net, a list of pins returned by get_pins, or a list of nets returned by get_nets. The -hierarchical option cannot be used with -of_objects.

patterns A list of cell (instance) name patterns.

The get_cells command returns a list of all cell instances that match *patterns*.

Without -regexp Unix style file glob pattern matching is used. With -regexp TCL regular expression matching is used. When -nocase is used regular expressions are case insensitive. The -nocase flag can only be used with -regexp.

get_clocks [-regexp]
[-nocase]
[-quiet]
patterns

-quiet Do not report an error if *patterns* do not match anything.

patterns A list of clock name patterns.

The get_clocks command returns a list of all clocks that have been defined.

Without -regexp Unix style file glob pattern matching is used. With -regexp TCL regular expression matching is used. When -nocase is used regular expressions are case insensitive. The -nocase flag can only be used with -regexp.

get_fanin -to *sink_list*
[-flat]
[-only_cells]
[-startpoints_only]
[-levels *level_count*]
[-pin_levels *pin_count*]
[-trace_arcs timing|enabled|all]

<i>sink_list</i>	List of pins, ports, or nets to find the fanin of. For nets, the fanin of driver pins on the nets are returned.
<i>-flat</i>	Without <i>-flat</i> only pins at the same hierarchy level as the sinks are returned. With <i>-flat</i> pins in the fanin at any hierarchy level are returned.
<i>-only_cells</i>	Return the instances connected to the pins in the fanin.
<i>-startpoints_only</i>	Only return pins that are startpoints.
<i>level_count</i>	Only return pins within <i>level_count</i> instance traversals.
<i>pin_count</i>	Only return pins within <i>pin_count</i> pin traversals.
<i>-trace_arcs</i>	With 'timing' and 'enabled' values only arcs that are not disabled are traversed. With a value of 'all' even disabled arcs are traversed.

The `get_fanin` command returns traverses the design from *sink_list* pins, ports or nets backwards and return the fanin pins or instances.

get_fanout	<code>-from source_list</code> <code>[-flat]</code> <code>[-only_cells]</code> <code>[-endpoints_only]</code> <code>[-levels level_count]</code> <code>[-pin_levels pin_count]</code> <code>[-trace_arcs timing enabled all]</code>
-------------------	---

<i>source_list</i>	List of pins, ports, or nets to find the fanout of. For nets, the fanout of load pins on the nets are returned.
<i>-flat</i>	Without <i>-flat</i> only pins at the same hierarchy level as the sinks are returned. With <i>-flat</i> pins in the fanout at any hierarchy level are returned.
<i>-only_cells</i>	Return the instances connected to the pins in the fanout.
<i>-endpoints_only</i>	Only return pins that are endpoints.
<i>level_count</i>	Only return pins within <i>level_count</i> instance traversals.
<i>pin_count</i>	Only return pins within <i>pin_count</i> pin traversals.
<i>-trace_arcs</i>	With 'timing' and 'enabled' values only arcs that are not disabled are traversed. With a value of 'all' even disabled arcs are traversed.

The `get_fanout` command traverses the design from *source_list* pins, ports or nets backwards and return the fanout pins or instances.

get_lib_cells	<code>[-of_objects <i>objects</i>] [-hsc <i>separator</i>] [-regexp] [-nocase] [-quiet] <i>patterns</i></code>
<i>objects</i>	A list of cell (instance) objects.
<i>separator</i>	Character that separates the library name and cell name in <i>pattern</i> .
<i>-quiet</i>	Do not report an error if <i>patterns</i> do not match anything.
<i>patterns</i>	A list of library cell name patterns of the form <code>library_name/cell_name</code> .

The `get_lib_cells` command returns a list of library cells that match *pattern*. The library name can be prepended to the cell name pattern with the *separator* character, which defaults to `hierarchy_separator`.

Without `-regexp` Unix style file glob pattern matching is used. With `-regexp` TCL regular expression matching is used. When `-nocase` is used regular expressions are case insensitive. The `-nocase` flag can only be used with `-regexp`.

get_lib_pins	<code>[-hsc separator] [-regexp] [-nocase] [-quiet] <i>patterns</i></code>
<i>separator</i>	Character that separates the library name, cell name and port name in <i>pattern</i> .
<code>-quiet</code>	Do not report an error if <i>patterns</i> do not match anything.
<i>patterns</i>	A list of library port name patterns of the form <code>library_name/cell_name/port_name</code> .

The `get_lib_pins` command returns a list of library ports that match *pattern*. Use *separator* to separate the library and cell name patterns from the port name in *pattern*.

Without `-regexp` Unix style file glob pattern matching is used. With `-regexp` TCL regular expression matching is used. When `-nocase` is used regular expressions are case insensitive. The `-nocase` flag can only be used with `-regexp`.

get_libs	<code>[-regexp] [-nocase] [-quiet] <i>patterns</i></code>
<code>-quiet</code>	Do not report an error if <i>patterns</i> do not match anything.
<i>patterns</i>	A list of library name patterns.

The `get_libs` command returns a list of clocks that match *patterns*.

Without `-regexp` Unix style file glob pattern matching is used. With `-regexp` TCL regular expression matching is used. When `-nocase` is used regular expressions are case insensitive. The `-nocase` flag can only be used with `-regexp`.

get_nets	<code>[-hierarchical]</code> <code>[-hsc <i>separator</i>]</code> <code>[-regexp]</code> <code>[-nocase]</code> <code>[-quiet]</code> <code>[-of_objects <i>objects</i>]</code> <code>[<i>patterns</i>]</code>
<code>-hierarchical</code>	Searches hierarchy levels below the current instance for matches.
<i>separator</i>	Character that separates hierarchical instance names and the net name in <i>pattern</i> .
<code>-quiet</code>	Do not report an error if <i>patterns</i> do not match anything.
<i>objects</i>	The name of a pin or instance, a list of pins returned by <code>get_pins</code> , or a list of instances returned by <code>get_cells</code> . The <code>-hierarchical</code> option cannot be used with <code>-of_objects</code> .
<i>patterns</i>	A list of net name patterns.

The `get_nets` command returns a list of all nets that match *patterns*.

Without `-regexp` Unix style file glob pattern matching is used. With `-regexp` TCL regular expression matching is used. When `-nocase` is used regular expressions are case insensitive. The `-nocase` flag can only be used with `-regexp`.

get_full_name	<i>object</i>
<i>object</i>	A library, cell, port, instance, pin or timing arc object.

Return the name of *object*. Equivalent to `[get_property object full_name]`.

get_name	<i>object</i>
<i>object</i>	A library, cell, port, instance, pin or timing arc object.

Return the name of *object*. Equivalent to `[get_property object name]`.

get_pins	<ul style="list-style-type: none"> [-hierarchical] [-hsc <i>separator</i>] [-filter <i>expr</i>] [-regexp] [-nocase] [-quiet] [-of_objects <i>objects</i>] [<i>patterns</i>]
-hierarchical	Searches hierarchy levels below the current instance for matches.
separator	Character that separates hierarchical instance names and the port name in <i>pattern</i> .
-quiet	Do not report an error if <i>patterns</i> do not match anything.
expr	A filter expression of the form $attribute == ~= pattern$ where <i>attribute</i> is an attribute supported by the <code>get_property</code> command.
objects	The name of a net or instance, a list of nets returned by <code>get_nets</code> , or a list of instances returned by <code>get_cells</code> . The <code>-hierarchical</code> option cannot be used with <code>-of_objects</code> .
patterns	A list of pin name patterns.

The `get_pins` command returns a list of all instance pins that match *patterns*.

Without `-regexp` Unix style file glob pattern matching is used. With `-regexp` TCL regular expression matching is used. When `-nocase` is used regular expressions are case insensitive. The `-nocase` flag can only be used with `-regexp`.

get_ports	<ul style="list-style-type: none"> [-filter <i>expr</i>] [-regexp] [-nocase] [-quiet] [-of_objects <i>objects</i>] [<i>patterns</i>]
-quiet	Do not report an error if <i>patterns</i> do not match any clocks.

<i>expr</i>	A filter expression of the form $attribute == ~= pattern$ where <i>attribute</i> is an attribute supported by the <i>get_property</i> command.
<i>objects</i>	The name of a net, or a list of nets returned by <i>get_nets</i> .
<i>patterns</i>	A list of port name patterns.

The *get_ports* command returns a list of all top level ports that match *patterns*.

Without *-regexp* Unix style file glob pattern matching is used. With *-regexp* TCL regular expression matching is used. When *-nocase* is used regular expressions are case insensitive. The *-nocase* flag can only be used with *-regexp*.

get_property	<i>[-object_type object_type]</i>
	<i>object</i>
	<i>property</i>
<i>object_type</i>	cell pin net port clock timing_arc
<i>object</i>	An <i>object</i> or <i>object name</i> . <i>-object_type</i> is required if <i>object</i> is an <i>object name</i> .
<i>property</i>	A <i>property name</i> .

The properties for different objects types are shown below.

cell

base_name
filename
full_name
library
name

clock

full_name
is_generated
name
period
propagated
sources

edge

```
delay_max_fall
delay_min_fall
delay_max_rise
delay_min_rise
full_name
from_pin
sense
to_pin
```

instance (SDC cell)

```
cell
full_name
ref_name
liberty_cell
name
```

liberty_cell

```
area
base_name
dont_use
filename
full_name
is_buffer
is_inverter
library
name
```

liberty_library

```
filename
full_name
name
```

liberty_port

```
capacitance
direction
drive_resistance
drive_resistance_fall_max
drive_resistance_fall_min
drive_resistance_rise_max
drive_resistance_rise_min
full_name
intrinsic_delay
intrinsic_delay_fall_max
intrinsic_delay_fall_min
intrinsic_delay_rise_max
intrinsic_delay_rise_min
is_register_clock
lib_cell
name
```

library

```
name
full_name

net

full_name
name

path (PathEnd)

endpoint
endpoint_clock
endpoint_clock_pin
slack
startpoint
startpoint_clock
points

pin

activity
actual_fall_transition_max
actual_fall_transition_min
actual_rise_transition_max
actual_rise_transition_min
clocks
direction
full_name
is_register_clock
lib_pin_name
max_fall_slack
max_rise_slack
min_fall_slack
min_rise_slack
name

port

activity
actual_fall_transition_min
actual_fall_transition_max
actual_rise_transition_min
actual_rise_transition_max
direction
full_name
liberty_port
min_fall_slack
max_fall_slack
min_rise_slack
max_rise_slack
name

point (PathRef)
```

arrival
pin
required
slack

get_timing_edges	<code>[-from <i>from_pins</i>] [-to <i>to_pins</i>] [-of_objects <i>objects</i>] [-filter <i>expr</i>] [<i>patterns</i>]</code>
<i>from_pin</i>	A collection of pins.
<i>to_pin</i>	A collection of pins.
<i>objects</i>	A collection of instances or library cells. The -from and -to options cannot be used with -of_objects.
<i>expr</i>	A filter expression of the form <i>attribute</i> == ~= <i>pattern</i> where <i>attribute</i> is an attribute supported by the get_property command.

The get_timing_edges command returns a collection of timing edges (arcs) to, from or between pins. The result can be passed to get_property or set_disable_timing.

group_path	<code>-name <i>group_name</i> [-weight <i>weight</i>] [-critical_range <i>range</i>] [-from <i>from_list</i>] [-rise_from <i>from_list</i>] [-fall_from <i>from_list</i>] [-through <i>through_list</i>] [-rise_through <i>through_list</i>] [-fall_through <i>through_list</i>] [-to <i>to_list</i>] [-rise_to <i>to_list</i>] [-fall_to <i>to_list</i>]</code>
<i>group_name</i>	The name of the path group.
<i>weight</i>	Not supported.
<i>range</i>	Not supported.
<i>from_list</i>	A list of clocks, instances, ports or pins.

through_list A list of instances, pins or nets.

to_list A list of clocks, instances, ports or pins.

The `group_path` command is used to group paths reported by the `report_checks` command. See `set_false_path` for a description of allowed *from_list*, *through_list* and *to_list* objects.

link *[cell_name]*

Alias for `link_design`.

link_design *[cell_name]*

Link (elaborate, flatten) the design with *cell_name* as the top level cell. The design must be linked after reading netlist and library files. The default value of *cell_name* is the current design.

The linker creates empty "block box" cells for instances the reference undefined cells when the variable `link_create_black_boxes` is true. When `link_create_black_boxes` is false an error is reported and the link fails.

The `link_design` command returns 1 if the link succeeds and 0 if it fails.

make_instance *inst_path*
lib_cell

inst_path A hierarchical instance name.

lib_cell The library cell of the new instance.

The `make_instance` command makes an instance of library cell *lib_cell*.

make_net *net_name_list*

net_name_list A list of net names.

Creates a net for each hierarchical net name.

read_liberty	<code>[-corner <i>corner_name</i>] [-min] [-max] [-no_latch_infer] <i>filename</i></code>
<i>corner_name</i>	Use library for process corner <i>corner_name</i> delay calculation.
-min	Use library for min delay calculation.
-max	Use library for max delay calculation.
<i>filename</i>	The name of the liberty library file to read.

The `read_liberty` command reads a Liberty format library file. The first library that is read sets the units used by SDC/TCL commands and reporting. The `include_file` attribute is supported.

Cells that have a triad of timing arcs between three pins as shown below are inferred as latches:

```
cell (inferred_latch) {
    pin(D) {
        direction : input ;
        timing () {
            related_pin : "E" ;
            timing_type : setup_falling ;
        }
        timing () {
            related_pin : "E" ;
            timing_type : hold_falling ;
        }
    }
    pin(E) {
        direction : input;
    }
    pin(Q) {
        direction : output ;
        timing () {
            related_pin : "D" ;
        }
        timing () {
            related_pin : "E" ;
            timing_type : rising_edge ;
        }
    }
}
```

In this example a positive level-sensitive latch is inferred.

When the `read_liberty -no_latch_infer` flag is used latches are not inferred. If a cell has the `interface_timing_true` attribute, no latches are inferred in the cell.

Files compressed with gzip are automatically uncompressed.

read_sdc	[-echo] <i>filename</i>
-echo	Print each command before evaluating it.
<i>filename</i>	SDC command file.

Read SDC commands from *filename*.

The `read_sdc` command stops and reports any errors encountered while reading a file unless `sta_continue_on_error` is 1.

Files compressed with gzip are automatically uncompressed.

read_sdf	[-corner <i>corner_name</i>] [-unescaped_dividers] <i>filename</i>
<i>corner_name</i>	Process corner delays to annotate.
-unescaped_dividers	With this option path names in the SDF do not have to escape hierarchy dividers when the path name is escaped. For example, the escaped Verilog name "\inst1/inst2" can be referenced as "inst1/inst2". The correct SDF name is "inst1\inst2", since the divider does not represent a change in hierarchy in this case.
<i>filename</i>	The name of the SDF file to read.

Read SDF delay values from a file. The min and max values in the SDF tuples are used to annotate the delays for *corner*. The typical values in the SDF tuples are ignored. If multiple corners are defined `-corner` must be specified.

Files compressed with gzip are automatically uncompressed.

`INCREMENT` is supported as an alias for `INCREMENTAL`.

The following SDF statements are not supported.

`PART`
`INSTANCE` wildcards

read_spef	<code>[-min]</code> <code>[-max]</code> <code>[-path <i>path</i>]</code> <code>[-corner <i>corner</i>]</code> <code>[-keep_capacitive_coupling]</code> <code>[-coupling_reduction_factor <i>factor</i>]</code> <code>[-reduce_to pi_elmore pi_pole_residue2]</code> <code>[-delete_after_reduce]</code> <code>[-quiet]</code> <code><i>filenames</i></code>
<i>corner</i>	The name of a process corner.
<i>path</i>	Hierarchical instance path to annotate with parasitics.
<code>-keep_capacitive_coupling</code>	Keep coupling capacitors in parasitic networks rather than converting them to grounded capacitors.
<i>factor</i>	Factor to multiply coupling capacitance by when reducing parasitic networks.
<code>-reduce_to</code>	Reduce detailed parasitics to a PI/Elmore or PI/Pole residue model as each net is read.
<code>-delete_after_reduce</code>	Delete the detailed parasitic network after reducing it.
<code>-quiet</code>	Do not print error or warning messages.
<code>-save</code>	Save the parasitics database after reading it (OpenAccess only).
<i>filename</i>	The name of the parasitics file to read.

The `read_spef` command reads a file of net parasitics in SPEF format. Use the `-report_parasitic_annotation` command to check for nets that are not annotated.

Files compressed with gzip are automatically uncompressed.

Separate parasitics can be annotated for corners and min and max paths using the `-corner`, `-min` and `-max` arguments.

With the `-reduce_to` and `-delete_after_reduce` options, parasitic networks are reduced after each net is read, substantially reducing the memory footprint required to store the parasitics.

Coupling capacitors are multiplied by the `-coupling_reduction_factor` when a parasitic network is reduced.

The following SPEF constructs are ignored.

```
*DESIGN_FLOW (all values are ignored)
*S slews
*D driving cell
*I pin capacitances (library cell capacitances are used instead)
*Q r_net load poles
*K r_net load residues
```

If the SPEF file contains triplet values the first value is used.

Parasitic networks (DSPEF) can be annotated on hierarchical blocks using the `-path` argument to specify the instance path to the block. Parasitic networks in the higher level netlist are stitched together at the hierarchical pins of the blocks.

read_power_activities	<code>[-scope <i>scope</i>]</code> <code>-vcd <i>filename</i></code>
<i>scope</i>	The VCD scope of the current design to extract simulation data. Typically the test bench name and design under test instance name. Scope levels are separated with '/'.
<i>filename</i>	The name of the VCD file to read.

The `read_power_activities` command reads a VCD (Value Change Dump) file from a Verilog simulator and extracts pin activities and duty cycles for use in power estimation. Files compressed with gzip are supported.

read_verilog	<i>filename</i>
<i>filename</i>	The name of the verilog file to read.

The `read_verilog` command reads a gate level verilog netlist. After all verilog netlist and Liberty libraries are read the design must be linked with the `link_design` command.

Verilog 2001 module port declaratation support has been added. An example is shown below.

```
module top (input in1, in2, clk1, clk2, clk3,
            output out);
```

Files compressed with gzip are automatically uncompressed.

replace_cell	<i>instance_list</i> <i>replacement_cell</i>
<i>instance_list</i>	A list of instances to swap the cell.
<i>replacement_cell</i>	The replacement lib cell.

The `replace_cell` command changes the cell of an instance. The replacement cell must have the same port list (number, name, and order) as the instance's existing cell for the replacement to be successful.

report_annotated_check	<code>[-setup]</code> <code>[-hold]</code> <code>[-recovery]</code> <code>[-removal]</code> <code>[-nochange]</code> <code>[-width]</code> <code>[-period]</code> <code>[-max_skew]</code> <code>[-max_line <i>lines</i>]</code> <code>[-list_annotated]</code> <code>[-list_not_annotated]</code> <code>[-constant_arcs]</code>
<code>-setup</code>	Report annotated setup checks.
<code>-hold</code>	Report annotated hold checks.
<code>-recovery</code>	Report annotated recovery checks.
<code>-removal</code>	Report annotated removal checks.
<code>-nochange</code>	Report annotated nochange checks.
<code>-width</code>	Report annotated width checks.
<code>-period</code>	Report annotated period checks.
<code>-max_skew</code>	Report annotated max skew checks.
<code><i>lines</i></code>	Maximum number of lines listed by the <code>-list_annotated</code> and <code>-list_not_annotated</code> options.
<code>-list_annotated</code>	List annotated timing arcs.
<code>-list_not_annotated</code>	List unannotated timing arcs.

-constant_arcs Report separate annotation counts for arcs disabled by logic constants (set_logic_one, set_logic_zero).

The `report_annotated_check` command reports a summary of SDF timing check annotation. The `-list_annotated` and `-list_not_annotated` options can be used to list arcs that are annotated or not annotated.

report_annotated_delay	<code>[-cell] [-net] [-from_in_ports] [-to_out_ports] [-max_line <i>lines</i>] [-list_annotated] [-list_not_annotated] [-constant_arcs]</code>
<code>-cell</code>	Report annotated cell delays.
<code>-net</code>	Report annotated internal net delays.
<code>-from_in_ports</code>	Report annotated delays from input ports.
<code>-to_out_ports</code>	Report annotated delays to output ports.
<code>lines</code>	Maximum number of lines listed by the <code>-list_annotated</code> and <code>-list_not_annotated</code> options.
<code>-list_annotated</code>	List annotated timing arcs.
<code>-list_not_annotated</code>	List unannotated timing arcs.
<code>-constant_arcs</code>	Report separate annotation counts for arcs disabled by logic constants (set_logic_one, set_logic_zero).

The `report_annotated_delay` command reports a summary of SDF delay annotation. Without the `-from_in_ports` and `-to_out_ports` options annotation of arcs to and from top level ports is not reported. The `-list_annotated` and `-list_not_annotated` options can be used to list arcs that are annotated or not annotated.

report_cell	<code>[-connections] [-verbose] <i>instance_path</i> [> <i>filename</i>] [>> <i>filename</i>]</code>
<code>-connections</code>	Report the instance pins.

-verbose With **-connections** also report all pins connected to each instance pin net.

instance_path Hierarchical path to the instance.

The `report_cell` command is an alias for `report_instance`.

report_checks	<code>[-from <i>from_list</i> -rise_from <i>from_list</i> -fall_from <i>from_list</i>] [-through <i>through_list</i> -rise_through <i>through_list</i> -fall_through <i>through_list</i>] [-to <i>to_list</i> -rise_to <i>to_list</i> -fall_to <i>to_list</i>] [-unconstrained] [-path_delay <i>min min_rise min_fall</i> i^{max} max_rise max_fall i^{min} min_max] [-group_count <i>path_count</i>] [-endpoint_count <i>endpoint_path_count</i>] [-unique_paths_to_endpoint] [-corner <i>corner_name</i>] [-slack_max <i>max_slack</i>] [-slack_min <i>min_slack</i>] [-sort_by_slack] [-path_group <i>group_names</i>] [-format <i>end full short summary</i> full_clock full_clock_expanded] [-fields <i>fields</i>] [-digits <i>digits</i>] [-no_line_split] [> <i>filename</i>] [>> <i>filename</i>]</code>
<i>from_list</i>	A list of clocks, instances, ports or pins.
<i>through_list</i>	A list of instances, pins or nets.
<i>to_list</i>	A list of clocks, instances, ports or pins.

<code>-unconstrained</code>	Report unconstrained paths. By default unconstrained paths are not reported.
<code>-path_delay min</code>	Report <code>min</code> path (<code>hold</code>) checks.
<code>-path_delay min_rise</code>	Report <code>min</code> path (<code>hold</code>) checks for rising endpoints.
<code>-path_delay min_fall</code>	Report <code>min</code> path (<code>hold</code>) checks for falling endpoints.
<code>-path_delay max</code>	Report <code>max</code> path (<code>setup</code>) checks.
<code>-path_delay max_rise</code>	Report <code>max</code> path (<code>setup</code>) checks for rising endpoints.
<code>-path_delay max_fall</code>	Report <code>max</code> path (<code>setup</code>) checks for falling endpoints.
<code>-path_delay min_max</code>	Report <code>max</code> and <code>min</code> path (<code>setup</code> and <code>hold</code>) checks.
<code>path_count</code>	The number of paths to report in each path group. The default is 1.
<code>endpoint_path_count</code>	The number of paths to report for each endpoint. The default is 1.
<code>-unique_paths_to_endpoint</code>	When multiple paths to an endpoint are specified with <code>-endpoint_count</code> many of the paths may differ only in the rise/fall edges of the pins in the paths. With this option only the worst path through the set of pins is reported.
<code>corner_name</code>	Only report paths for this process corner. But default the paths across all process corners are reported.
<code>max_slack</code>	Only report paths with less slack than <code>max_slack</code> .
<code>min_slack</code>	Only report paths with more slack than <code>min_slack</code> .
<code>-sort_by_slack</code>	Sort paths by slack rather than slack grouped by path group.
<code>group_names</code>	List of path group names to report. All path groups are reported if this option is not specified.
<code>-format end</code>	Report path ends in one line with delay, required time and slack.
<code>-format full</code>	Report path start and end points and the path. This is the default path type.

<code>-format full_clock</code>	Report path start and end points, the path, and the source and target clock paths.
<code>-format full_clock_expanded</code>	Report path start and end points, the path, and the source and target clock paths. If the clock is generated and propagated, the path from the clock source pin is also reported.
<code>-format short</code>	Report only path start and end points.
<code>-format summary</code>	Report only path ends with delay.
<i>fields</i>	List of capacitance slew input_pins nets fanout
<i>digits</i>	The number of digits after the decimal point to report. The default value is the variable default_significant_digits.
<code>-no_line_splits</code>	Do not split long lines into multiple lines.

The `report_checks` command reports paths in the design. See `set_false_path` for a description of allowed *from_list*, *through_list* and *to_list* objects.

Use the `-corner` keyword to report timing for one process corner. With no `-corner` keyword report timing reports the corner with the smallest slack for each timing check.

<code>report_check_types</code>	<ul style="list-style-type: none"> [-violators] [-verbose] [-format slack_only end] [-max_delay] [-min_delay] [-recovery] [-removal] [-clock_gating_setup] [-clock_gating_hold] [-max_slew] [-min_slew] [-min_pulse_width] [-min_period] [-digits <i>digits</i>] [-no_split_lines] [> <i>filename</i>] [>> <i>filename</i>]
---------------------------------	---

<code>-violators</code>	Report all violated timing and design rule constraints.
<code>-verbose</code>	Use a verbose output format.
<code>-format</code>	<code>slack_only</code> reports the minimum slack for each check. <code>end</code> reports the endpoint for each check.
<code>-max_delay</code>	Report setup and max delay path delay constraints.
<code>-min_delay</code>	Report hold and min delay path delay constraints.
<code>-recovery</code>	Report asynchronous recovery checks.
<code>-removal</code>	Report asynchronous removal checks.
<code>-clock_gating_setup</code>	Report gated clock enable setup checks.
<code>-clock_gating_hold</code>	Report gated clock hold setup checks.
<code>-max_slew</code>	Report max transition design rule checks.
<code>-max_skew</code>	Report max skew design rule checks.
<code>-min_pulse_width</code>	Report min pulse width design rule checks.
<code>-min_period</code>	Report min period design rule checks.
<code>-min_slew</code>	Report min slew design rule checks.
<code>digits</code>	The number of digits after the decimal point to report. The default is <code>default_significant_digits</code> .
<code>-no_split_lines</code>	

The `report_check_types` command reports the slack for each type of timing and design rule constraint. The keyword options allow a subset of the constraint types to be reported.

report_clock_min_period `[-clocks clocks]`
`[-include_port_paths]`

`clocks` The clocks to report.

<code>-include_port_paths</code>	Include paths from input port and to output ports.
----------------------------------	--

Report the minimum period and maximum frequency for *clocks*. If the `-clocks` argument is not specified all clock are reported. The minimum period is determined by examining the smallest slack paths between registers the rising edges of the clock or between falling edges of the clock. Paths between different clocks, different clock edges of the same clock, level sensitive latches, or paths constrained by `set_multicycle_path`, `set_max_path` are not considered.

<code>report_clock_properties</code>	<code>[clock_names]</code>
--------------------------------------	----------------------------

<code>clock_names</code>	List of clock names to report.
--------------------------	--------------------------------

The `report_clock_properties` command reports the period and rise/fall edge times for each clock that has been defined.

<code>report_clock_skew</code>	<code>[-setup -hold]</code> <code>[-clock clocks]</code> <code>[-digits digits]</code>
--------------------------------	--

<code>-setup</code>	Report skew for setup checks.
---------------------	-------------------------------

<code>-hold</code>	Report skew for hold checks.
--------------------	------------------------------

<code>clocks</code>	The target clocks to report.
---------------------	------------------------------

<code>digits</code>	The number of digits to report for delays.
---------------------	--

Report the clock skew between the source and target clocks for setup or hold timing checks.

<code>report_dcalc</code>	<code>[-from from_pin]</code> <code>[-to to_pin]</code> <code>[-corner corner_name]</code> <code>[-min]</code> <code>[-max]</code> <code>[-digits digits]</code> <code>[> filename]</code> <code>[>> filename]</code>
---------------------------	---

<code>from_pin</code>	An instance pin.
-----------------------	------------------

<code>to_pin</code>	An instance pin.
---------------------	------------------

digits The number of digits after the decimal point to report. The default is `default_significant_digits`.

The `report_dcabc` command shows how the delays between instance pins are calculated. It is useful for debugging problems with delay calculation.

Use the `-corner` keyword to specify a process corner. The `-corner` keyword is required if more than one process corner is defined.

report_disabled_edges

The `report_disabled_edges` command reports disabled timing arcs along with the reason they are disabled. Each disabled timing arc is reported as the instance name along with the from and to ports of the arc. The disable reason is shown next. Arcs that are disabled with `set_disable_timing` are reported with constraint as the reason. Arcs that are disabled by constants are reported with constant as the reason along with the constant instance pin and value. Arcs that are disabled to break combinational feedback loops are reported with loop as the reason.

```
> report_disabled_edges
u1 A B constant B=0
```

report_instance [-connections]
[-verbose]
instance_path
[> *filename*]
[>> *filename*]

`-connections` Report the pins connected to the net.

`-verbose` Report the connections of each pin.

instance_path Hierarchical path to a instance.

report_lib_cell *cell_name*
[> *filename*]
[>> *filename*]

cell_name The name of a library cell.

Describe the liberty library cell *cell_name*.

report_net	<code>[-connections] [-verbose] [-significant_digits <i>digits</i>] <i>net_path</i> [> <i>filename</i>] [>> <i>filename</i>]</code>
<code>-connections</code>	Report the net pins.
<code>-verbose</code>	With <code>-connections</code> also report all pins connected to each instance pin net.
<code>net_path</code>	Hierarchical path to a net.

report_parasitic_annotation	<code>[-report_unannotated] [> <i>filename</i>] [>> <i>filename</i>]</code>
------------------------------------	--

Report SPEF parasitic annotation completeness. When the `-report_unannotated` option is used unannotated and partially annotated nets and pins are reported.

report_pin	<code><i>pin_path</i> [> <i>filename</i>] [>> <i>filename</i>]</code>
<code><i>pin_path</i></code>	Hierarchical path to a pin.

report_power	<code>[-instances <i>instances</i>] [-digits <i>digits</i>] [> <i>filename</i>] [>> <i>filename</i>]</code>
<code><i>instances</i></code>	Report the power for <i>instances</i> .
<code><i>digits</i></code>	The number of digits after the decimal point to report. The default is <code>default_significant_digits</code> .

The `report_power` command uses static power analysis based on propagated or annotated pin activities in the circuit using Liberty power models. The internal, switching, leakage and total power are reported. Design power is reported separately for combinational, sequential, macro and pad groups. Power values are reported in watts.

Use `-instances` to report power for specific instances.

The `read_power_activities` command can be used to read activities from a file based on simulation. If no simulation activities are available, the `set_power_activity` command should be used to set the activity of input ports or pins in the design. The default input activity and duty for inputs are 0.1 and 0.5 respectively. The activities are propagated from annotated input ports or pins through gates and used in the power calculations.

Group	Internal Power	Switching Power	Leakage Power	Total Power	
<hr/>					
Sequential	3.29e-06	3.41e-08	2.37e-07	3.56e-06	92.4%
Combinational	1.86e-07	3.31e-08	7.51e-08	2.94e-07	7.6%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.0%
<hr/>					
Total	3.48e-06	6.72e-08	3.12e-07	3.86e-06	100.0%
	90.2%	1.7%	8.1%		

report_pulse_width_checks [-verbose]

[-digits *digits*]
[-no_line_splits]
[*pins*]
[> *filename*]
[>> *filename*]

-verbose Use a verbose output format.

digits The number of digits after the decimal point to report. The default is `default_significant_digits`.

-no_line_splits

pins List of pins or ports to report.

The `report_pulse_width_checks` command reports min pulse width checks for pins in the clock network. If `pins` is not specified all clock network pins are reported.

report_units

Report the command units.

```
report_units
  time 1ns
  capacitance 1pF
```

```
resistance 1kohm
voltage 1v
current 1A
power 1pW
distance 1um
```

report_worst_slack	<code>[-min] [-max] [-digits <i>digits</i>]</code>
<code>-max</code>	Report the worst max/setup slack.
<code>-min</code>	Report the worst min/hold slack.
<code><i>digits</i></code>	The number of digits after the decimal point to report. The default is <code>default_significant_digits</code> .
set_assigned_check	<code>-setup -hold -recovery -removal [-rise] [-fall] [-corner <i>corner_name</i>] [-min] [-max] [-from <i>from_pins</i>] [-to <i>to_pins</i>] [-clock rise fall] [-cond <i>sdf_cond</i>] [-worst] <i>margin</i></code>
<code>-setup -hold -recovery -removal</code>	The timing check type to annotate.
<code>-rise</code>	Annotate the rising delays.
<code>-fall</code>	Annotate the falling delays.
<code><i>corner_name</i></code>	The name of a process corner.
<code>-min</code>	Annotate the minimum value of the process corner.
<code>-max</code>	Annotate the maximum value of the process corner.
<code><i>from_pins</i></code>	A list of pins for the clock.

<i>to_pins</i>	A list of pins for the data.
<i>-clock</i>	The clock pin transition.
<i>-worst</i>	Ignored.
<i>margin</i>	The timing check value.

The `set_assigned_check` command is used to annotate the timing checks between two pins on an instance. The annotated delay overrides the calculated delay. This command is a interactive way to back-annotate delays like an SDF file.

Use the `-corner` keyword to specify a process corner. The `-corner` keyword is required if more than one process corner is defined.

<code>set_assigned_delay</code>	<code>-cell -net</code> <code>[-rise]</code> <code>[-fall]</code> <code>[-corner <i>corner_name</i>]</code> <code>[-min]</code> <code>[-max]</code> <code>[-from <i>from_pins</i>]</code> <code>[-to <i>to_pins</i>]</code> <code><i>delay</i></code>
<code>-cell</code>	Annotate the delays between two pins on an instance.
<code>-net</code>	Annotate the delays between two pins on a net.
<code>-rise</code>	Annotate the rising delays.
<code>-fall</code>	Annotate the falling delays.
<code>-min</code>	Annotate the minimum delays.
<code>-max</code>	Annotate the maximum delays.
<i>from_pins</i>	A list of pins.
<i>to_pins</i>	A list of pins.
<i>delay</i>	The delay between <i>from_pins</i> and <i>to_pins</i> .

The `set_assigned_delay` command is used to annotate the delays between two pins on an instance or net. The annotated delay overrides the calculated delay. This command is a interactive way to back-annotate delays like an SDF file.

Use the `-corner` keyword to specify a process corner. The `-corner` keyword is required if more than one process corner is defined.

set_assigned_transition	<code>[-rise] [-fall] [-corner <i>corner_name</i>] [-min] [-max] <i>slew</i> <i>pin_list</i></code>
<code>-rise</code>	Annotate the rising transition.
<code>-fall</code>	Annotate the falling transition.
<code>-min</code>	Annotate the minimum transition time.
<code>-max</code>	Annotate the maximum transition time.
<code>slew</code>	The pin transition time.
<code>pin_list</code>	A list of pins.

The `set_assigned_transition` command is used to annotate the transition time (slew) of a pin. The annotated transition time overrides the calculated transition time.

Use the `-corner` keyword to specify a process corner. The `-corner` keyword is required if more than one process corner is defined.

set_case_analysis	<code>0 1 zero one rise rising fall falling <i>port_or_pin_list</i></code>
<code>port_or_pin_list</code>	A list of ports or pins.

The `set_case_analysis` command sets the signal on a port or pin to a constant logic value. No paths are propagated from constant pins. Constant values set with the `set_case_analysis` command are propagated through downstream gates.

set_clock_gating_check	[-setup <i>setup_time</i>] [-hold <i>hold_time</i>] [-rise] [-fall] [-high] [-low] [<i>objects</i>]
<i>setup_time</i>	Clock enable setup margin.
<i>hold_time</i>	Clock enable hold margin.
-rise	The setup/hold margin is for the rising edge of the clock enable.
-fall	The setup/hold margin is for the falling edge of the clock enable.
-high	The gating clock is active high (pin and instance objects only).
-low	The gating clock is active low (pin and instance objects only).
<i>objects</i>	A list of clocks, instances, pins or ports.

The `set_clock_gating_check` command is used to add setup or hold timing checks for data signals used to gate clocks.

If no objects are specified the setup/hold margin is global and applies to all clock gating circuits in the design. If neither of the `-rise` and `-fall` options are used the setup/hold margin applies to the rising and falling edges of the clock gating signal.

Normally the library cell function is used to determine the active state of the clock. The clock is active high for AND/NAND functions and active low for OR/NOR functions. The `-high` and `-low` options are used to specify the active state of the clock for other cells, such as a MUX.

If multiple `set_clock_gating_check` commands apply to a clock gating instance the priority of the commands is shown below (highest to lowest priority).

```
clock enable pin
instance
clock pin
clock
global
```

set_clock_groups	<ul style="list-style-type: none"> [-name <i>name</i>] [-logically_exclusive] [-physically_exclusive] [-asynchronous] [-allow_paths] -group <i>clocks</i>
<i>name</i>	The clock group name.
-logically_exclusive	The clocks in different groups do not interact logically but can be physically present on the same chip. Paths between clock groups are considered for noise analysis.
-physically_exclusive	The clocks in different groups cannot be present at the same time on a chip. Paths between clock groups are <i>not</i> considered for noise analysis.
-asynchronous	The clock groups are asynchronous. Paths between clock groups are considered for noise analysis.
-allow_paths	
<i>clocks</i>	A list of clocks in the group.
<p>The <code>set_clock_groups</code> command is used to define groups of clocks that interact with each other. Clocks in different groups do not interact and paths between them are not reported. Use a <code>-group</code> argument for each clock group.</p>	
set_clock_latency	<ul style="list-style-type: none"> [-source] [-clock <i>clock</i>] [-rise] [-fall] [-min] [-max] <i>delay</i> <i>objects</i>
-source	The latency is at the clock source.
<i>clock</i>	If multiple clocks are defined at a pin this use this option to specify the latency for a specific clock.
-rise	The latency is for the rising edge of the clock.

-fall	The latency is for the falling edge of the clock.
-min	<i>delay</i> is the minimum latency.
-max	<i>delay</i> is the maximum latency.
delay	Clock source or insertion delay.
objects	A list of clocks, pins or ports.

The `set_clock_latency` command describes expected delays of the clock tree when analyzing a design using ideal clocks. Use the `-source` option to specify latency at the clock source, also known as insertion delay. Source latency is delay in the clock tree that is external to the design or a clock tree internal to an instance that implements a complex logic function.

set_clock_transition	<code>[-rise]</code> <code>[-fall]</code> <code>[-min]</code> <code>[-max]</code> <i>transition</i> <i>clocks</i>
-rise	The transition time is for the rising edge of the clock.
-fall	The transition time is for the falling edge of the clock.
-min	<i>transition</i> is the minimum transition time.
-max	<i>transition</i> is the maximum transition time.
transition	Clock transition time (slew).
clocks	A list of clocks.

The `set_clock_transition` command describes expected transition times of the clock tree when analyzing a design using ideal clocks.

set_clock_uncertainty	<code>[-from -rise_from -fall_from <i>from_clock</i>] [-to -rise_to -fall_to <i>to_clock</i>] [-rise] [-fall] [-setup] [-hold] <i>uncertainty</i> [<i>objects</i>]</code>
<i>from_clock</i>	Inter-clock uncertainty source clock.
<i>to_clock</i>	Inter-clock uncertainty target clock.
-rise	Inter-clock target clock rise edge, alternative to -rise_to. Inter-clock target clock rise edge, alternative to -rise_to.
-fall	Inter-clock target clock rise edge, alternative to -fall_to.
-setup	<i>uncertainty</i> is for setup checks.
-hold	<i>uncertainty</i> is for hold checks.
<i>uncertainty</i>	Clock uncertainty.
<i>objects</i>	A list of clocks, ports or pins.

The `set_clock_uncertainty` command specifies the uncertainty or jitter in a clock. The uncertainty for a clock can be specified on its source pin or port, or the clock itself.

```
set_clock_uncertainty .1 [get_clock clk1]
```

Inter-clock uncertainty between the source and target clocks of timing checks is specified with the `-from|-rise_from|-fall_from` and `-to|-rise_to|-fall_to` arguments.

```
set_clock_uncertainty -from [get_clock clk1] -to [get_clocks clk2] .1
```

The following commands are equivalent.

```
set_clock_uncertainty -from [get_clock clk1] -rise_to [get_clocks clk2] .1
set_clock_uncertainty -from [get_clock clk1] -to [get_clocks clk2] -rise .1
```

set_cmd_units	<code>[-capacitance <i>cap_unit</i>] [-resistance <i>res_unit</i>] [-time <i>time_unit</i>] [-voltage <i>voltage_unit</i>] [-current <i>current_unit</i>] [-power <i>power_unit</i>] [-distance <i>distance_unit</i>]</code>
<i>cap_unit</i>	The capacitance scale factor followed by 'f'.
<i>res_unit</i>	The resistance scale factor followed by 'ohm'.
<i>time_unit</i>	The time scale factor followed by 's'.
<i>voltage_unit</i>	The voltage scale factor followed by 'v'.
<i>current_unit</i>	The current scale factor followed by 'A'.
<i>power_unit</i>	The power scale factor followed by 'w'.
<i>distance_unit</i>	The distance scale factor followed by 'm'.

The `set_cmd_units` command is used to change the units used by the STA command interpreter when parsing commands and reporting results. The default units are the units specified in the first Liberty library file that is read.

Units are specified as a scale factor followed by a unit name. The scale factors are as follows.

```
M 1E+6
k 1E+3
m 1E-3
u 1E-6
n 1E-9
p 1E-12
f 1E-15
```

An example of the `set_units` command is shown below.

```
set_cmd_units -time ns -capacitance pF -current mA -voltage V
               -resistance kOhm -distance um
```

set_data_check	<ul style="list-style-type: none"> [-from <i>from_object</i>] [-rise_from <i>from_object</i>] [-fall_from <i>from_object</i>] [-to <i>to_object</i>] [-rise_to <i>to_object</i>] [-fall_to <i>to_object</i>] [-setup] [-hold] [-clock <i>clock</i>] <p><i>margin</i></p>
<i>from_object</i>	A pin used as the timing check reference.
<i>to_object</i>	A pin that the setup/hold check is applied to.
-setup	Add a setup timing check.
-hold	Add a hold timing check.
<i>clock</i>	The setup/hold check clock.
<i>margin</i>	The setup or hold time margin.

The `set_data_check` command is used to add a setup or hold timing check between two pins.

set_disable_inferred_clock_gating	
<i>objects</i>	A list of clock gating instances, clock gating pins, or clock enable pins.

The `set_disable_inferred_clock_gating` command disables clock gating checks on a clock gating instance, clock gating pin, or clock gating enable pin.

set_disable_timing	<ul style="list-style-type: none"> [-from <i>from_port</i>] [-to <i>to_port</i>] <p><i>objects</i></p>
<i>from_port</i>	
<i>to_port</i>	

objects A list of instances, ports, pins, cells or [library/]cell/port.

The `set_disable_timing` command is used to disable paths through pins in the design. There are many different forms of the command depending on the objects specified in *objects*.

All timing paths through an instance are disabled when *objects* contains an instance. Timing checks in the instance are *not* disabled.

```
set_disable_timing u2
```

The `-from` and `-to` options can be used to restrict the disabled path to those from, to or between specific pins on the instance.

```
set_disable_timing -from A u2
set_disable_timing -to Z u2
set_disable_timing -from A -to Z u2
```

A list of top level ports or instance pins can also be disabled.

```
set_disable_timing u2/Z
set_disable_timing in1
```

Timing paths through all instances of a library cell in the design can be disabled by naming the cell using a hierarchy separator between the library and cell name. Paths from or to a cell port can be disabled with the `-from` and `-to` options or a port name after library and cell names.

```
set_disable_timing liberty1/snl_bufx2
set_disable_timing -from A liberty1/snl_bufx
set_disable_timing -to Z liberty1/snl_bufx
set_disable_timing liberty1/snl_bufx2/A
```

set_drive	<code>[-rise]</code> <code>[-fall]</code> <code>[-max]</code> <code>[-min]</code> <i>resistance</i> <i>port_list</i>
------------------	---

`-rise` This is the drive resistance of the rising edge of the input.

`-fall` This is the drive resistance of the falling edge of the input.

`-max` This is the drive resistance for maximum path delays.

`-min` This is the drive resistance for minimum path delays.

<i>resistance</i>	The external drive resistance.
<i>port_list</i>	A list of ports.

The `set_drive` command describes the resistance of an input port external driver.

set_driving_cell	<code>[-lib_cell <i>cell_name</i>] [-library <i>library</i>] [-rise] [-fall] [-min] [-max] [-pin <i>pin</i>] [-from_pin <i>from_pin</i>] [-input_transition_rise <i>trans_rise</i>] [-input_transition_fall <i>trans_fall</i>] <i>port_list</i></code>
<i>cell_name</i>	The cell of driver.
<i>library</i>	The library of the driving cell.
<code>-rise</code>	This is the driving cell for the rising edge of the input.
<code>-fall</code>	This is the driving cell for the falling edge of the input.
<code>-max</code>	This is the driving cell for maximum path delays.
<code>-min</code>	This is the driving cell for minimum path delays.
<i>pin</i>	The output port of the driving cell.
<i>from_pin</i>	Use paths through the driving cell from this pin to the output pin.
<i>trans_rise</i>	The transition time for a rising input at <i>from_pin</i> .
<i>trans_fall</i>	The transition time for a falling input at <i>from_pin</i> .
<i>port_list</i>	A list of ports.

The `set_driving_cell` command describes an input port external driver.

set_false_path	[-setup] [-hold] [-rise] [-fall] [-from <i>from_list</i>] [-rise_from <i>from_list</i>] [-fall_from <i>from_list</i>] [-through <i>through_list</i>] [-rise_through <i>through_list</i>] [-fall_through <i>through_list</i>] [-to <i>to_list</i>] [-rise_to <i>to_list</i>] [-fall_to <i>to_list</i>] [-reset_path]
-setup	Only apply to setup checks.
-hold	Only apply to hold checks.
-rise	Only apply to rising path edges.
-fall	Only apply to falling path edges.
-reset_path	Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first.
<i>from_list</i>	A list of clocks, instances, ports or pins.
<i>through_list</i>	A list of instances, pins or nets.
<i>to_list</i>	A list of clocks, instances, ports or pins.

The `set_false_path` command disables timing along a path from, through and to a group of design objects.

Objects in *from_list* can be clocks, register/latch instances, or register/latch clock pins. The `-rise_from` and `-fall_from` keywords restrict the false paths to a specific clock edge.

Objects in *through_list* can be nets, instances, instance pins, or hierarchical pins,. The `-rise_through` and `-fall_through` keywords restrict the false paths to a specific path edge that traverses through the object.

Objects in *to_list* can be clocks, register/latch instances, or register/latch clock pins. The `-rise_to` and `-fall_to` keywords restrict the false paths to a specific transition at the path end.

set_fanout_load	<i>fanout</i> <i>port_list</i>
------------------------	-----------------------------------

This command is ignored.

set_hierarchy_separator	<i>separator</i>
--------------------------------	------------------

<i>separator</i>	Character used to separate hierarchical names.
------------------	--

Set the character used to separate names in a hierarchical instance, net or pin name. This separator is used by the command interpreter to read arguments and print results. The default separator is '/'.

set_ideal_latency	<i>[-rise] [-fall] [-min] [-max] delay objects</i>
--------------------------	--

The `set_ideal_latency` command is parsed but ignored.

set_ideal_network	<i>[-no_propagation] objects</i>
--------------------------	----------------------------------

The `set_ideal_network` command is parsed but ignored.

set_ideal_transition	<i>[-rise] [-fall] [-min] [-max] transition_time objects</i>
-----------------------------	--

The `set_ideal_transition` command is parsed but ignored.

set_input_delay	<i>[-rise] [-fall] [-max] [-min] [-clock <i>clock</i>] [-clock_fall] [-reference_pin <i>ref_pin</i>] [-source_latency_included] [-network_latency_included] [-add_delay] <i>delay</i> <i>port_pin_list</i></i>
------------------------	--

-rise	This is the arrival time for the rising edge of the input.
--------------	--

<code>-fall</code>	This is the arrival time for the falling edge of the input.
<code>-max</code>	This is the minimum arrival time.
<code>-min</code>	This is the maximum arrival time.
<code>clock</code>	The arrival time is from this clock.
<code>-clock_fall</code>	The arrival time is from the falling edge of <code>clock</code>
<code>ref_pin</code>	The arrival time is with respect to the clock that arrives at <code>ref_pin</code> .
<code>-source_latency_included</code>	If <code>-source_latency_included</code> is not specified the clock source latency (insertion delay) is added to the delay value.
<code>-network_latency_included</code>	If <code>-network_latency_included</code> is not specified and the clock is ideal the clock latency is added to the delay value.
<code>-add_delay</code>	Add this arrival to any existing arrivals on <code>port_pin_list</code> .
<code>delay</code>	The arrival time after <code>clock</code> .
<code>pin_port_list</code>	A list of pins or ports.

The `set_input_delay` command is used to specify the arrival time of an input signal. Unless the `-add_delay` option is specified, any existing arrival time is replaced.

The `-reference_pin` option is used to specify an arrival time with respect to the arrival on a pin in the clock network. For propagated clocks, the input arrival time is relative to the clock arrival time at the reference pin (the clock source latency and network latency from the clock source to the reference pin). For ideal clocks, input arrival time is relative to the reference pin clock source latency. With the `-clock_fall` flag the arrival time is relative to the falling transition at the reference pin. If no clocks arrive at the reference pin the `set_input_delay` command is ignored. If no `-clock` is specified the arrival time is with respect to all clocks that arrive at the reference pin. The `-source_latency_included` and `-network_latency_included` options cannot be used with `-reference_pin`.

Paths from inputs that do not have an arrival time defined by `set_input_delay` are not reported. Set the `sta_input_port_default_clock` variable to 1 to report paths from inputs without a `set_input_delay`.

set_input_transition	[-rise] [-fall] [-max] [-min] <i>transition</i> <i>port_list</i>
-rise	This is the transition time for the rising edge of the input.
-fall	This is the transition time for the falling edge of the input.
-max	This is the minimum transition time.
-min	This is the maximum transition time.
<i>transition</i>	The transition time (slew).
<i>port_list</i>	A list of ports.

The `set_input_transition` command is used to specify the transition time (slew) of an input signal.

set_level_shifter_strategy [-rule *rule_type*]

This command is parsed and ignored by timing analysis.

set_level_shifter_threshold [-voltage *voltage*]

This command is parsed and ignored by timing analysis.

set_load	[-rise] [-fall] [-max] [-min] [-subtract_pin_load] [-pin_load] [-wire_load] <i>capacitance</i> <i>objects</i>
-rise	The capacitance is for rising edge delays.

-fall	The capacitance is for falling edge delays.
-max	The capacitance is for maximum path delays.
-min	The capacitance is for minimum path delays.
-subtract_pin_load	Subtract the capacitance of all instance pins connected to the net from <i>wire capacitance</i> .
-pin_load	<i>capcitanee</i> is external instance pin capacitance (ports only).
-wire_load	<i>capcitanee</i> is external wire capacitance (ports only).
<i>capacitance</i>	The capacitance, in library capacitance units.
<i>objects</i>	A list of nets or ports.

The `set_load` command annotates capacitance on a net or port.

Ports can have external wire or pin capacitance that is annotated separately with the `-pin_load` and `-wire_load` options. Without the `-pin_load` and `-wire_load` options pin capacitance is annotated. External capacitances are used by delay calculator to find output driver delays and transition times.

Net wire capacitance can also be annotated with the `set_load` command. If the `-subtract_pin_load` option is specified the capacitance of all instance pins connected to the net is subtracted from *capacitance*. `set_load` command annotates capacitance has precedence over RC SPEF parasitics.

set_logic_dc	<i>port_list</i>
<i>port_pin_list</i>	List of ports or pins.

Set a port or pin to a constant unknown logic value. No paths are propagated from constant pins.

set_logic_one	<i>port_list</i>
<i>port_pin_list</i>	List of ports or pins.

Set a port or pin to a constant logic one value. No paths are propagated from constant pins. Constant values set with the `set_logic_one` command are **not** propagated through downstream gates.

set_logic_zero	<i>port_list</i>
<i>port_pin_list</i>	List of ports or pins.

Set a port or pin to a constant logic zero value. No paths are propagated from constant pins. Constant values set with the `set_logic_zero` command are **not** propagated through downstream gates.

set_max_area *area*

area

The `set_max_area` command is ignored during timing but is included in SDC files that are written.

set_max_capacitance *capacitance objects*

capacitance

objects List of ports or cells.

The `set_max_capacitance` command is ignored during timing but is included in SDC files that are written.

set_max_delay [-rise]
[-fall]
[-from *from_list*]
[-rise_from *from_list*]
[-fall_from *from_list*]
[-through *through_list*]
[-rise_through *through_list*]
[-fall_through *through_list*]
[-to *to_list*]
[-rise_to *to_list*]
[-fall_to *to_list*]
[-reset_path]
[-ignore_clock_latency]
delay

-rise Only constrain paths to rising edges.

-fall Only constrain paths to falling edges.

from_list A list of clocks, instances, ports or pins.

through_list A list of instances, pins or nets.

to_list A list of clocks, instances, ports or pins.

<code>-ignore_clock_latency</code>	Ignore clock latency at the source and target registers.
<code>-reset_path</code>	Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first.
<i>delay</i>	The maximum delay.

The `set_max_delay` command constrains the maximum delay through combinational logic paths. See `set_false_path` for a description of allowed *from_list*, *through_list* and *to_list* objects. If the *to_list* ends at a timing check the setup/hold time is included in the path delay.

When the `-ignore_clock_latency` option is used clock latency at the source and destination of the path delay is ignored. The constraint is reported in the default path group (**default**) rather than the clock path group when the path ends at a timing check.

set_max_dynamic_power *power [unit]*

The `set_max_dynamic_power` command is ignored.

set_max_fanout *fanout*
 objects

fanout

objects List of ports or cells.

The `set_max_fanout` command is ignored during timing but is included in SDC files that are written.

set_max_leakage_power *power [unit]*

The `set_max_leakage_power` command is ignored.

set_max_time_borrow *delay*
 objects

delay The maximum time the latches can borrow.

objects List of clocks, instances or pins.

The `set_max_time_borrow` command specifies the maximum amount of time that latches can borrow. Time borrowing is the time that a data input to a transparent latch arrives after the latch opens.

set_max_transition	<ul style="list-style-type: none"> [-data_path] [-clock_path] [-rise] [-fall] <i>transition</i> <i>objects</i>
<i>transition</i>	The maximum transition time (slew).
<i>objects</i>	List of clocks, ports or designs.

The `set_max_transition` command specifies the maximum transition time (slew) design rule checked by the `report_constraint -max_transition` command.

If specified for a design, the default maximum transition is set for the design.

If specified for a clock, the maximum transition is applied to all pins in the clock domain. The `-clock_path` option restricts the maximum transition to clocks in clock paths. The `-data_path` option restricts the maximum transition to clocks data paths. The `-clock_path`, `-data_path`, `-rise` and `-fall` options only apply to clock objects.

set_min_capacitance	<ul style="list-style-type: none"> <i>capacitance</i> <i>objects</i>
<i>capacitance</i>	
<i>objects</i>	List of ports or cells.

The `set_min_capacitance` command is ignored during timing but is included in SDC files that are written.

set_min_delay	<ul style="list-style-type: none"> [-rise] [-fall] [-from <i>from_list</i>] [-rise_from <i>from_list</i>] [-fall_from <i>from_list</i>] [-through <i>through_list</i>] [-rise_through <i>through_list</i>] [-fall_through <i>through_list</i>] [-to <i>to_list</i>] [-rise_to <i>to_list</i>] [-fall_to <i>to_list</i>] [-ignore_clock_latency] [-reset_path]
	<i>delay</i>

<code>-rise</code>	Only constrain paths to rising edges.
<code>-fall</code>	Only constrain paths to falling edges.
<code>from_list</code>	A list of clocks, instances, ports or pins.
<code>through_list</code>	A list of instances, pins or nets.
<code>to_list</code>	A list of clocks, instances, ports or pins.
<code>-ignore_clock_latency</code>	Ignore clock latency at the source and target registers.
<code>-reset_path</code>	Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first.
<code>delay</code>	The minimum delay.

The `set_min_delay` command constrains the minimum delay through combinational logic. See `set_false_path` for a description of allowed `from_list`, `through_list` and `to_list` objects. If the `to_list` ends at a timing check the setup/hold time is included in the path delay.

When the `-ignore_clock_latency` option is used clock latency at the source and destination of the path delay is ignored. The constraint is reported in the default path group (`**default**`) rather than the clock path group when the path ends at a timing check.

<code>set_min_pulse_width</code>	<code>[-high]</code> <code>[-low]</code> <code>min_width</code> <code>objects</code>
<code>-high</code>	Set the minimum high pulse width.
<code>-low</code>	Set the minimum low pulse width.
<code>min_width</code>	
<code>objects</code>	List of pins, instances or clocks.

If `-low` and `-high` are not specified the minimum width applies to both high and low pulses.

set_multicycle_path	<ul style="list-style-type: none"> [-setup] [-hold] [-rise] [-fall] [-start] [-end] [-from <i>from_list</i>] [-rise_from <i>from_list</i>] [-fall_from <i>from_list</i>] [-through <i>through_list</i>] [-rise_through <i>through_list</i>] [-fall_through <i>through_list</i>] [-to <i>to_list</i>] [-rise_to <i>to_list</i>] [-fall_to <i>to_list</i>] [-reset_path] <p><i>path_multiplier</i></p>
-setup	Only apply to setup checks.
-hold	Only apply to hold checks.
-rise	Only apply to rising path edges.
-fall	Only apply to falling path edges.
-start	Multiply the source clock period by <i>period_multiplier</i> .
-end	Multiply the target clock period by <i>period_multiplier</i> .
-reset_path	Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first.
<i>from_list</i>	A list of clocks, instances, ports or pins.
<i>through_list</i>	A list of instances, pins or nets.
<i>to_list</i>	A list of clocks, instances, ports or pins.
<i>path_multiplier</i>	The number of clock periods to add to the path required time.

Normally the path between two registers or latches is assumed to take one clock cycle. The `set_multicycle_path` command overrides this assumption and allows multiple clock cycles for a timing check. See `set_false_path` for a description of allowed *from_list*, *through_list* and *to_list* objects.

set_operating_conditions	<code>[-analysis_type {single bc_wc on_chip_variation}]</code> <code>[-library <i>lib</i>]</code> <code>[<i>condition</i>]</code> <code>[-min <i>min_condition</i>]</code> <code>[-max <i>max_condition</i>]</code> <code>[-min_library <i>min_lib</i>]</code> <code>[-max_library <i>max_lib</i>]</code>
single	Use one operating condition for min and max paths.
bc_wc	Best case, worst case analysis. Setup checks use <i>max_condition</i> for clock and data paths. Hold checks use the <i>min_condition</i> for clock and data paths.
on_chip_variation	The min and max operating conditions represent variations on the chip that can occur simultaneously. Setup checks use <i>max_condition</i> for data paths and <i>min_condition</i> for clock paths. Hold checks use <i>min_condition</i> for data paths and <i>max_condition</i> for clock paths. This is the default analysis type.
<i>lib</i>	The name of the library that contains <i>condition</i> .
<i>condition</i>	The operating condition for analysis type single.
<i>min_condition</i>	The operating condition to use for min paths and hold checks.
<i>max_condition</i>	The operating condition to use for max paths and setup checks.
<i>min_lib</i>	The name of the library that contains <i>min_condition</i> .
<i>max_lib</i>	The name of the library that contains <i>max_condition</i> .

The `set_operating_conditions` command is used to specify the type of analysis performed and the operating conditions used to derive library data.

set_output_delay	[-rise] [-fall] [-max] [-min] [-clock <i>clock</i>] [-clock_fall] [-reference_pin <i>ref_pin</i>] [-source_latency_included] [-network_latency_included] [-add_delay] <i>delay</i> <i>port_pin_list</i>
-rise	This is the arrival time for the rising edge of the input.
-fall	This is the arrival time for the falling edge of the input.
-max	This is the minimum arrival time.
-min	This is the maximum arrival time.
<i>clock</i>	The departure time is from this clock.
-clock_fall	The departure time is from the falling edge of <i>clock</i> .
<i>ref_pin</i>	The departure time is with respect to the clock that arrives at <i>ref_pin</i> .
-add_delay	Add this departure to any existing arrivals on <i>port_pin_list</i> .
<i>delay</i>	The departure time after <i>clock</i> .
<i>pin_port_list</i>	A list of pins or ports.

The `set_output_delay` command is used to specify the departure time of an output signal. Unless the `-add_delay` option is specified any existing departure time is replaced.

The `-reference_pin` option is used to specify a departure time with respect to the arrival on a pin in the clock network. For propagated clocks, the output departure time is relative to the clock arrival time at the reference pin (the clock source latency and network latency from the clock source to the reference pin). For ideal clocks, output departure time is relative to the reference pin clock source latency. With the `-clock_fall` flag the departure time is relative to the falling transition at the reference pin. If no clocks arrive at the reference pin the `set_output_delay` command is ignored. If no `-clock` is specified the departure time is with respect to all

clocks that arrive at the reference pin. The `-source_latency_included` and `-network_latency_included` options cannot be used with `-reference_pin`.

set_port_fanout_number	<code>[-min] [-max] <i>fanout</i> <i>port_list</i></code>
<code>-min</code>	The fanout for minimum path delay calculation.
<code>-max</code>	The fanout for maximum path delay calculation.
<code><i>fanout</i></code>	The external fanout of the ports.
<code><i>port_list</i></code>	A list of ports.

set_power_activity	<code>[-global] [-input] [-input_ports <i>ports</i>] [-pins <i>pins</i>] [-activity <i>activity</i>] [-duty <i>duty</i>]</code>
<code>-global</code>	Set the activity/duty for all non-clock pins.
<code>-input</code>	Set the default input port activity/duty.
<code><i>input_ports</i></code>	Set the input port activity/duty.
<code><i>pins</i></code>	Set the pin activity/duty.
<code><i>activity</i></code>	The activity, or number of transitions per clock cycle.
<code><i>duty</i></code>	The duty, or probability the signal is high. Defaults to 0.5.

The `set_power_activity` command is used to set the activity and duty used for power analysis globally or for input ports or pins in the design.

The default input activity and duty for inputs are 0.1 and 0.5 respectively, which is equivalent to the following command:

```
set_power_activity -input -activity 0.1 -duty 0.5
```

set_propagated_clock	<i>objects</i>
-----------------------------	----------------

<i>objects</i>	A list of clocks, ports or pins.
----------------	----------------------------------

The **set_propagated_clock** command changes a clock tree from an ideal network that has no delay one that uses calculated or back-annotated gate and interconnect delays. When *objects* is a port or pin, clock delays downstream of the object are used.

set_pvt	<i>instances</i>
----------------	------------------

[-min]	
[-max]	
[-process <i>process</i>]	
[-voltage <i>voltage</i>]	
[-temperature <i>temperature</i>]	

<i>instances</i>	A list instances.
------------------	-------------------

-min	Only set the PVT values for max delay paths.
------	--

-max	Only set the PVT values for min delay paths.
------	--

<i>process</i>	A process value (float).
----------------	--------------------------

<i>voltage</i>	A voltage value (float).
----------------	--------------------------

<i>temperature</i>	A temperature value (float).
--------------------	------------------------------

The **set_pvt** command sets the process, voltage and temperature values used during delay calculation for a specific instance in the design.

set_sense	[-type clock data]
------------------	--------------------

[-positive]	
[-negative]	
[-pulse <i>pulse_type</i>]	
[-stop_propagation]	
[-clock <i>clocks</i>]	
<i>pins</i>	

-positive	The clock sense is positive unite.
-----------	------------------------------------

-negative	The clock sense is negative unite.
pulse_type	<code>rise_triggered_high_pulse</code> <code>rise_triggered_low_pulse</code> <code>fall_triggered_high_pulse</code> <code>fall_triggered_low_pulse</code>
-stop_propagation	Stop propagating <i>clocks</i> clocks at <i>pins</i> .
clocks	A list of clocks to apply the sense
pins	A list of pins.

The `set_sense` command is used to modify the propagation of a clock signal. Note that `-type` data is not supported. The clock sense is set with the `-positive` and `-negative` flags. Use the `-stop_propagation` flag to stop the clock from propagating beyond a pin. The `-positive`, `-negative`, `-stop_propagation`, and `-pulse` options are mutually exclusive. If the `-clock` option is not used the command applies to all clocks that traverse *pins*. The `-pulse` option is currently not supported.

set_timing_derate	<code>[-rise]</code> <code>[-fall]</code> <code>[-early]</code> <code>[-late]</code> <code>[-clock]</code> <code>[-data]</code> <code>[-net_delay]</code> <code>[-cell_delay]</code> <code>[-cell_check]</code> <code>derate</code> <code>[objects]</code>
-early	Derate early (min) paths.
-late	Derate late (max) paths.
-clock	Derate paths in the clock network.
-data	Derate data paths.
-net_delay	Derate net (interconnect) delays.
-cell_delay	Derate cell delays.
-cell_check	Derate cell timing check margins.

<i>derate</i>	The derating factor to apply to delays.
<i>objects</i>	A list of instances, library cells, or nets.

The `set_timing_derate` command is used to derate delay calculation results used by the STA. If the `-early` and `-late` flags are omitted the both min and max paths are derated. If the `-clock` and `-data` flags are not used the derating both clock and data paths are derated.

Use the `unset_timing_derate` command to remove all derating factors.

set_resistance	<code>[-max] [-min] <i>resistance</i> <i>net_list</i></code>
<code>-min</code>	The resistance for minimum path delay calculation.
<code>-max</code>	The resistance for maximum path delay calculation.
<i>resistance</i>	The net resistance.
<i>net_list</i>	A list of nets.

set_units	<code>[-capacitance <i>cap_unit</i>] [-resistance <i>res_unit</i>] [-time <i>time_unit</i>] [-voltage <i>voltage_unit</i>] [-current <i>current_unit</i>] [-power <i>power_unit</i>] [-distance <i>distance_unit</i>]</code>
<i>cap_unit</i>	The capacitance scale factor followed by 'f'.
<i>res_unit</i>	The resistance scale factor followed by 'ohm'.
<i>time_unit</i>	The time scale factor followed by 's'.
<i>voltage_unit</i>	The voltage scale factor followed by 'v'.
<i>current_unit</i>	The current scale factor followed by 'A'.

power_unit The power scale factor followed by 'w'.

The `set_units` command is used to **check** the units used by the STA command interpreter when parsing commands and reporting results. If the current units differ from the `set_unit` value a warning is printed. Use the `set_cmd_units` command to change the command units.

Units are specified as a scale factor followed by a unit name. The scale factors are as follows.

```
M 1E+6
k 1E+3
m 1E-3
u 1E-6
n 1E-9
p 1E-12
f 1E-15
```

An example of the `set_units` command is shown below.

```
set_units -time ns -capacitance pF -current mA -voltage V -resistance kOhm
```

```
set_wire_load_min_block_size size
```

The `set_wire_load_min_block_size` command is not supported.

```
set_wire_load_mode top|enclosed|segmented
```

top

enclosed

segmented

The `set_wire_load_mode` command is ignored during timing but is included in SDC files that are written.

```
set_wire_load_model -name model_name
                     [-library library]
                     [-max]
                     [-min]
                     [objects]
```

model_name The name of a wire load model.

library Library to look for *model_name*.

<code>-max</code>	The wire load model is for maximum path delays.
<code>-min</code>	The wire load model is for minimum path delays.
<i>objects</i>	Not supported.

```
set_wire_load_selection_group [-library library]  
    [-max]  
    [-min]  
    group_name  
    [objects]
```

<i>library</i>	Library to look for <i>group_name</i> .
<code>-max</code>	The wire load selection is for maximum path delays.
<code>-min</code>	The wire load selection is for minimum path delays.
<i>group_name</i>	A wire load selection group name.
<i>objects</i>	Not supported.

The `set_wire_load_selection_group` command is parsed but not supported.

source	<code>[-echo] [-verbose] <i>filename</i> > <i>filename</i>] ">>> <i>filename</i>]</code>
<code>-echo</code>	Print each command before evaluating it.
<code>-verbose</code>	Print each command before evaluating it as well as the result it returns.
<i>filename</i>	The name of the file containing commands to read.

Read STA/SDC/Tcl commands from *filename*.

The `source` command stops and reports any errors encountered while reading a file unless `sta_continue_on_error` is 1.

unset_case_analysis *port_or_pin_list*

port_or_pin_list A list of ports or pins.

The unset_case_analysis command removes the constant values defined by the set_case_analysis command.

unset_clock_latency [-source]
objects

-source Specifies source clock latency (clock insertion delay).

objects A list of clocks, pins or ports.

The unset_clock_latency command removes the clock latency set with the set_clock_latency command.

unset_clock_transition *clocks*

clocks A list of clocks.

The unset_clock_transition command removes the clock transition set with the set_clock_transition command.

unset_clock_uncertainty [-from|-rise_from|-fall_from *from_clock*]
[-to|-rise_to|-fall_to *to_clock*]
[-rise]
[-fall]
[-setup]
[-hold]
[*objects*]

from_clock

to_clock

-rise The uncertainty is for the rising edge of the clock.

-fall The uncertainty is for the falling edge of the clock.

-setup *uncertainty* is the setup check uncertainty.

-hold *uncertainty* is the hold uncertainty.

<i>uncertainty</i>	Clock uncertainty.
<i>objects</i>	A list of clocks, ports or pins.

The unset_clock_uncertainty command removes clock uncertainty defined with the set_clock_uncertainty command.

unset_data_check	<ul style="list-style-type: none"> [-from <i>from_object</i>] [-rise_from <i>from_object</i>] [-fall_from <i>from_object</i>] [-to <i>to_object</i>] [-rise_to <i>to_object</i>] [-fall_to <i>to_object</i>] [-setup] [-hold] [-clock <i>clock</i>]
<i>from_object</i>	A pin used as the timing check reference.
<i>to_object</i>	A pin that the setup/hold check is applied to.
-setup	Add a setup timing check.
-hold	Add a hold timing check.
<i>clock</i>	The setup/hold check clock.

The unset_clock_transition command removes a setup or hold check defined by the set_data_check command.

unset_disable_inferred_clock_gating	
<i>objects</i>	A list of clock gating instances, clock gating pins, or clock enable pins.

The unset_disable_inferred_clock_gating command removes a previous set_disable_inferred_clock_gating command.

unset_disable_timing	<ul style="list-style-type: none"> [-from <i>from_port</i>] [-to <i>to_port</i>]
<i>objects</i>	

from_port

to_port

objects A list of instances, ports, pins, cells or [library/]cell/port.

The unset_disable_timing command is used to remove the effect of previous set_disable_timing commands.

unset_input_delay	[-rise] [-fall] [-max] [-min] [-clock <i>clock</i>] [-clock_fall] <i>port_pin_list</i>
-rise	This is the arrival time for the rising edge of the input.
-fall	This is the arrival time for the falling edge of the input.
-max	This is the minimum arrival time.
-min	This is the maximum arrival time.
<i>clock</i>	The arrival time is from this <i>clock</i> .
-clock_fall	The arrival time is from the falling edge of <i>clock</i>
<i>pin_port_list</i>	A list of pins or ports.

The unset_input_delay command removes a previously defined set_input_delay.

unset_output_delay	[-rise] [-fall] [-max] [-min] [-clock <i>clock</i>] [-clock_fall] <i>port_pin_list</i>
---------------------------	---

<code>-rise</code>	This is the arrival time for the rising edge of the input.
<code>-fall</code>	This is the arrival time for the falling edge of the input.
<code>-max</code>	This is the minimum arrival time.
<code>-min</code>	This is the maximum arrival time.
<code>clock</code>	The arrival time is from this clock.
<code>-clock_fall</code>	The arrival time is from the falling edge of <code>clock</code>
<code>pin_port_list</code>	A list of pins or ports.

The `unset_output_delay` command removes a previously defined `set_output_delay`.

<code>unset_path_exceptions</code>	<code>[-setup]</code> <code>[-hold]</code> <code>[-rise]</code> <code>[-fall]</code> <code>[-from <i>from_list</i>]</code> <code>[-rise_from <i>from_list</i>]</code> <code>[-fall_from <i>from_list</i>]</code> <code>[-through <i>through_list</i>]</code> <code>[-rise_through <i>through_list</i>]</code> <code>[-fall_through <i>through_list</i>]</code> <code>[-to <i>to_list</i>]</code> <code>[-rise_to <i>to_list</i>]</code> <code>[-fall_to <i>to_list</i>]</code>
<code>-setup</code>	Only apply to setup checks.
<code>-hold</code>	Only apply to hold checks.
<code>-rise</code>	Only apply to rising path edges.
<code>-fall</code>	Only apply to falling path edges.
<code><i>from_list</i></code>	A list of clocks, instances, ports or pins.

<i>through_list</i>	A list of instances, pins or nets.
<i>to_list</i>	A list of clocks, instances, ports or pins.

The unset_path_exceptions command removes any matching set_false_path, set_multicycle_path, set_max_delay, and set_min_delay exceptions.

unset_propagated_clock	<i>objects</i>
<i>objects</i>	A list of clocks, ports or pins.

Remove a previous set_propagated_clock command.

unset_timing_derate

Remove all derating factors set with the set_timing_derate command.

user_run_time

Returns the total user cpu run time in seconds as a float.

with_output_to_variable	<i>var { commands }</i>
<i>var</i>	The name of a variable to save the output of <i>commands</i> to.
<i>commands</i>	TCL commands that the output will be redirected from.

The with_output_to_variable command redirects the output of TCL commands to a variable.

write_path_spice	<i>-path_args path_args</i> <i>-spice_directory spice_directory</i> <i>-lib_subckt_file lib_subckts_file</i> <i>-model_file model_file</i> <i>-power power</i> <i>-ground ground</i> <i>-from -through -to</i> arguments as in report_checks.
<i>path_args</i>	

spice_directory Spice output directory.

<i>lib_subckts_file</i>	Cell transistor level subckts.
<i>model_file</i>	Transistor model definitions .included by <i>spice_file</i> .
<i>power</i>	Voltage supply name in <i>voltage_map</i> of the default liberty library.
<i>ground</i>	Ground supply name in <i>voltage_map</i> of the default liberty library.

The `write_path_spice` command writes a spice netlist for timing paths. Use `path_args` to specify `-from/-through/-to` as arguments to the `find_timing_paths` command. For each path, a spice netlist and the subckts referenced by the path are written in *spice_directory*. The spice netlist is written in `path_<id>.sp` and subckt file is `path_<id>.subckt`.

The spice netlists used by the path are written to *subckt_file*, which *spice_file* .includes. The device models used by the spice subckt netlists in *model_file* are also .included in *spice_file*. Power and ground names are specified with the `-power` and `-ground` arguments. The spice netlist includes a piecewise linear voltage source at the input and `.measure` statement for each gate delay and pin slew.

Example command:

```
write_path_spice -path_args {-from "in0" -to "out1" -unconstrained} \
-spice_directory $result_dir \
-lib_subckt_file "write_spice1.subckt" \
-model_file "write_spice1.models" \
-power VDD -ground VSS
```

write_sdc	[-digits <i>digits</i>] [-gzip] [-no_timestamp] <i>filename</i>
<i>digits</i>	The number of digits after the decimal point to report. The default is 4.
<code>-gzip</code>	Write a gzip compressed file.
<code>-no_timestamp</code>	Do not include a time and date in the SDC file.
<i>filename</i>	The name of the file to write the constraints to.

Write the constraints for the design in SDC format to *filename*.

write_sdf	<code>[-corner <i>corner_name</i>] [-divider / .] [-include_typ] [-digits <i>digits</i>] [-gzip] [-no_timestamp] [-no_version] <i>filename</i></code>
<i>corner_name</i>	Process corner delays to write.
-divider	Divider to use between hierarchy levels in pin and instance names. Must be '/' or '.'.
-include_typ	Include a 'typ' value in the SDF triple that is the average of min and max delays to satisfy some Verilog simulators that require three values in the delay triples.
<i>digits</i>	The number of digits after the decimal point to report. The default is 4.
-gzip	Write a gzip compressed file.
-no_timestamp	Do not write a DATE statement.
-no_version	Do not write a VERSION statement.
<i>filename</i>	The name of the file to write the constraints to.
<i>digits</i>	The number of digits after the decimal point to report. The default is 4.

Write the delay calculation delays for the design in SDF format to *filename*. The SDF TIMESCALE is same as the time_unit in the first liberty file read.

write_timing_model	<code>[-library_name <i>lib_name</i>] [-cell_name <i>cell_name</i>] [-corner <i>corner</i>] <i>filename</i></code>
<i>lib_name</i>	The name to use for the liberty library. Defaults to <i>cell_name</i> .
<i>cell_name</i>	The name to use for the liberty cell. Defaults to the top level module name.

<i>corner</i>	The process corner to use for extracting the model.
<i>filename</i>	Filename to write.

The `write_timing_model` command constructs a liberty timing model for the current design and writes it to *filename*. *cell_name* defaults to the cell name of the top level block in the design.

The SDC used to extract the block should include the `clock` definitions, but not `set_input_delay` or `set_output_delay` commands. Using `set_input_transition` with the slew from the block context will be used will improve the match between the timing model and the block netlist. Use the `set_propagated_clock` command to include clock latencies in the timing model.

The resulting timing model can be used in a hierarchical timing flow as a replacement for the block to speed up timing analysis. This hierarchical timing methodology does not handle timing exceptions that originate or terminate inside the block. The timing model includes:

- combinational paths between inputs and outputs
- setup and hold timing constraints on inputs
- clock to output timing paths

Resistance of long wires on inputs and outputs of the block cannot be modeled in Liberty. To reduce inaccuracies from wire resistance in technologies with resistive wires place buffers on inputs and outputs.

The extracted timing model setup/hold checks are scalar (no input slew dependence). Delay timing arcs are load dependent but do not include input slew dependency.

<code>write_verilog</code>	<code>[-sort]</code> <code>[-include_pwr_gnd]</code> <code>[-remove_cells lib_cells]</code> <i>filename</i>
<code>-sort</code>	Sort the instances in the netlist.
<code>-include_pwr_gnd</code>	Include power and ground pins on instances.
<code>lib_cells</code>	Liberty cells to remove. Use <code>get_lib_cells</code> , a list of cells names, or a cell name with wildcards.
<i>filename</i>	Filename to write.

The `write_verilog` command writes a verilog netlist to *filename*. Use `-sort` to sort the instances so the results are reproducible across operating systems. Use `-remove_cells` to remove instances of *lib_cells* from the netlist.

Variables

<code>hierarchy_separator</code>	Any character.
---	----------------

The hierarchy_separator separates instance names in a hierarchical instance, net, or pin name. The default value is '/'.

link_make_black_boxes	0 1
------------------------------	-----

When `link_make_black_boxes` is 1 the `link_design` command will make empty “black box” cells for instances that reference undefined cells. The default value is 1.

sta_bidirect_net_paths_enabled	0 1
---------------------------------------	-----

When set to 0, paths from bidirectional (inout) ports back through nets are disabled. When set to 1, paths from bidirectional paths from the net back into the instance are enabled. The default value is 0.

sta_continue_on_error	0 1
------------------------------	-----

The `source` and `read_sdc` commands stop and report any errors encountered while reading a file unless `sta_continue_on_error` is 1. The default value is 0.

sta_crpr_mode	same_pin same_transition
----------------------	--------------------------

When the data and clock paths of a timing check overlap (see `sta_crpr_enabled`), pessimism is removed independent of whether of the path rise/fall transitions. When `sta_crpr_mode` is `same_transition`, the pessimism is only removed if the path rise/fall transitions are the same. The default value is `same_pin`.

sta_cond_default_arcs_enabled	0 1
--------------------------------------	-----

When set to 0, default timing arcs with no condition (Liberty timing arcs with no “when” expression) are disabled if there are other conditional timing arcs between the same pins. The default value is 1.

sta_crpr_enabled	0 1
-------------------------	-----

During min/max timing analysis for `on_chip_variation` the data and clock paths may overlap. For a setup check the maximum path delays are used for the data and the minimum path delays are used for the clock. Because the gates cannot simultaneously have minimum and maximum delays the timing check slack is pessimistic. This pessimism is known as Common Reconvergent Pessimism Removal, or “CRPR”. Enabling CRPR slows down the analysis. The default value is 1.

sta_dynamic_loop_breaking	0 1
----------------------------------	-----

When `sta_dynamic_loop_breaking` is 0, combinational logic loops are disabled by disabling a timing arc that closes the loop. When `sta_dynamic_loop_breaking` is 1, all paths around the loop are reported. The default value is 0.

sta_gated_clock_checks_enabled	0 1
When <code>sta_gated_clock_checks_enabled</code> is 1, clock gating setup and hold timing checks are checked. The default value is 1.	
sta_input_port_default_clock	0 1
When <code>sta_input_port_default_clock</code> is 1 a default input arrival is added for input ports that do not have an arrival time specified with the <code>set_input_delay</code> command. The default value is 0.	
sta_internal_bidirect_instance_paths_enabled	0 1
When set to 0, paths from bidirectional (inout) ports back into the instance are disabled. When set to 1, paths from bidirectional ports back into the instance are enabled. The default value is 0.	
sta_pocv_enabled	0 1
Enable parametric on chip variation using statistical timing analysis. The default value is 0.	
sta_propagate_all_clocks	0 1
All clocks defined after <code>sta_propagate_all_clocks</code> is set to 1 are propagated. If it is set before any clocks are defined it has the same effect as <code>set_propagated_clock [all_clocks]</code>	
after all clocks have been defined. The default value is 0.	
sta_propagate_gated_clock_enable	0 1
When set to 1, paths of gated clock enables are propagated through the clock gating instances. If the gated clock controls sequential elements setting <code>sta_propagate_gated_clock_enable</code> to 0 prevents spurious paths from the clock enable. The default value is 1.	
sta_recovery_removal_checks_enabled	0 1
When <code>sta_recovery_removal_checks_enabled</code> is 0, recovery and removal timing checks are disabled. The default value is 1.	
sta_report_default_digits	integer

The number of digits to print after a decimal point. The default value is 2.

sta_preset_clear_arcs_enabled	0 1
--------------------------------------	-------

When set to 1, paths through asynchronous preset and clear timing arcs are searched. The default value is 0.

Version 2.0.15, Apr 29, 2019

Copyright (c) 2019, Parallax Software, Inc.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.